

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова
праця на правах рукопису

СЕМЕНОВ Сергій Юрійович

УДК 004.8:004.67:631.1

ДИСЕРТАЦІЯ

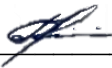
**ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ
ІНТЕЛЕКТУАЛІЗОВАНОЇ ТРАНСФОРМАЦІЇ ВИМІРЮВАЛЬНИХ
ДАНИХ АГРОМОНІТОРИНГУ НА ОСНОВІ ТУМАННОЇ АРХІТЕКТУРИ**

Спеціальність: 121 Інженерія програмного забезпечення

Галузь знань: 12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

 С.Ю. Семенов
(підпис, ініціали та прізвище здобувача)

Науковий керівник: ЛАКТИОНОВ Іван Сергійович, д-р техн. наук, професор

Дніпро – 2026

АНОТАЦІЯ

Семенов С.Ю. Програмно-алгоритмічне забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури. – Кваліфікаційна наукова робота на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 – Інженерія програмного забезпечення. – Національний технічний університет «Дніпровська політехніка», МОН України, Дніпро, 2026.

Дисертація складається зі вступу, чотирьох основних розділів, висновків, списку використаних джерел літератури зі 177 найменувань та чотирьох додатків загальним обсягом 38 сторінок. Обсяг дисертації складає 222 сторінки, з яких основний текст займає 150 сторінок та містить 44 рисунки і 27 таблиць.

Цю дисертаційну роботу присвячено розв’язанню актуальної науково-практичної задачі подальшого розвитку інформаційних і комп’ютерних технологій агротехнічного призначення шляхом розробки та використання програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури під час прогнозування врожайності сільськогосподарських культур із урахуванням значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту в конкретних агротехнологічних умовах.

У вступі обґрунтовано важливість і актуальність теми досліджень, сформульовано основну мету й завдання роботи, визначено об’єкт і предмет дослідження, наведено застосовані методи досліджень, сформульовано наукову новизну та практичне значення одержаних результатів, вказано інформацію щодо апробації, публікацій та особистого внеску здобувача, а також зазначено інформацію щодо використання й упровадження результатів досліджень.

Перший розділ присвячено аналізу й узагальненню характеристик сучасного стану світових і вітчизняних результатів науково-практичних досліджень і розробок у галузі цифровізації та інтелектуалізації рослинництва відкритого ґрунту, зокрема підходів, методів, моделей і програмно-алгоритмічних

засобів багатоетапного опрацювання даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур, що дозволило обґрунтувати мету й локалізувати напрямки і завдання досліджень цієї дисертації.

У другому розділі викладено результати розв’язання дослідницьких завдань, що пов’язані з формалізацією процесу комплексної інтелектуалізованої трансформації часово-просторових даних агромоніторингу під час прогнозування врожайності агрокультур у мінливих агрокліматичних умовах. Обґрунтовано доцільність використання туманної архітектури, розроблено структурно-функціональну організацію, інформаційну модель та базові модельні діаграми програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур.

У третьому розділі дисертації викладено результати розв’язання завдань щодо розробки та дослідження ефективності програмних компонент повноциклової трансформації даних агромоніторингу, що реалізують етапи збору, попередньої обробки, розвідувального аналізу, предикативної аналітики та інтерпретації моделей прогнозування врожайності із застосуванням методів статистичного й кореляційного аналізу, машинного навчання та пояснювального штучного інтелекту, що дозволило синтезувати деталізовану потокову модель функціонування досліджуваного програмно-алгоритмічного забезпечення.

У четвертому розділі висвітлено результати зі створення, тестування та валідації комп’ютерної технології повноциклової інтелектуалізованої трансформації даних агромоніторингу в межах туманної архітектури у вигляді вебзастосунку з інтерактивним користувацьким інтерфейсом, засобами візуалізації результатів, інтерпретації прогнозних моделей, інтеграції із зовнішніми агроінформаційними системами та механізмами сповіщення користувачів. Також у цьому розділі локалізовано основні обмеження реалізованого програмно-алгоритмічного забезпечення та обґрунтовано напрями подальшого розвитку й практичного використання запропонованих рішень.

У висновках узагальнено основні результати дисертаційного дослідження та сформульовано підсумкові положення щодо досягнення поставленої мети й виконання завдань роботи.

Додатки містять перелік наукових публікацій здобувача за темою дисертації, скан-копії документів, що підтверджують використання та впровадження результатів дослідження, а також програмні компоненти повноциклової інтелектуалізованої трансформації даних агромоніторингу та вебзастосунку прогнозування врожайності сільськогосподарських культур.

Наукова новизна одержаних результатів полягає в створенні нових та подальшому розвитку й удосконаленні існуючих методів і моделей повноциклової інтелектуалізованої трансформації часово-просторових даних агромоніторингу шляхом розробки та валідації туманної архітектури програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур, підходів до формалізації процесів багаторівневої агрегації й інтелектуальної аналітики даних агромоніторингу та методологічних засад проєктування програмно-алгоритмічних рішень агротехнічного призначення із залученням методів машинного навчання й пояснювального штучного інтелекту.

Практичне значення одержаних результатів полягає в реалізації та застосуванні програмно-алгоритмічних компонентів, структурно-функціональних і комп'ютерних моделей, що забезпечують комплексну обробку даних агрокліматичного моніторингу із підтримкою прийняття рішень під час прогнозування врожайності агрокультур, зокрема у вигляді вебзастосунку з інтерактивним інтерфейсом, засобами візуалізації, інтерпретації прогнозних моделей і можливістю інтеграції із зовнішніми агроінформаційними системами.

Пріоритетними напрямками подальших досліджень, які сформульовано на підставі критичного аналізу одержаних результатів, є розвиток функціональних можливостей і підвищення рівня технологічної готовності розробленого програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу.

Ключові слова: машинне навчання, штучний інтелект, цифровізація, програмне забезпечення, прогнозування, врожайність, агромоніторинг, просторово-часові дані, модель, вебзастосунок, туманна архітектура.

ABSTRACT

Semenov S.Yu. Software and algorithmic provision for the intellectualised transformation of agricultural monitoring data based on fog architecture. – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 121 – Software Engineering. – Dnipro University of Technology, Ministry of Education and Science of Ukraine, Dnipro, 2026.

The dissertation consists of an introduction, four chapters, conclusions, a list of 177 references, and four appendices totalling 38 pages. The total volume is 222 pages. The main text comprises 150 pages and includes 44 figures and 27 tables.

This dissertation addresses an important scientific and applied problem in the further development of information and computer technologies for agriculture. It focuses on the development and application of software and algorithmic provision for the comprehensive intellectualised transformation of agroclimatic monitoring data based on fog architecture for crop yield prediction. The proposed approach accounts for the significant variability of agroclimatic conditions, crop types, and crop protection practices under specific agricultural conditions.

The introduction substantiates the significance and relevance of the research topic. It defines the aim and tasks of the study, identifies the object and subject of the research, and describes the methods used. It also presents the scientific novelty and practical significance of the results, information on their approbation, publications, the author's personal contribution, and evidence of implementation and practical application.

The first chapter analyses and generalises the current state of international and national scientific and applied research in the field of digitalisation and intelligent transformation of open-field crop production. It focuses on approaches, methods, models, and software tools for multistage processing of agricultural monitoring data for crop yield prediction. This analysis substantiated the main aim and defined the research directions and tasks.

The second chapter presents the results of solving research tasks related to the formalisation of the comprehensive intellectualised transformation of spatio-temporal agroclimatic monitoring data for crop yield prediction under changing agroclimatic conditions. It substantiates the feasibility of applying fog architecture. It also develops the structural and functional organisation, the information model, and the basic model diagrams of the software and algorithmic provision for crop yield prediction.

The third chapter presents the results of developing and evaluating the effectiveness of software components for the end-to-end transformation of agroclimatic monitoring data. These components implement the stages of data collection, pre-processing, exploratory analysis, predictive analytics, and interpretation of crop yield prediction models. The study applies statistical and correlation analysis, machine learning, and explainable artificial intelligence. These results enabled the synthesis of a detailed operational pipeline for the proposed software and algorithmic provision.

The fourth chapter presents the development, testing, and validation of computer technology for the full-cycle intellectualised transformation of agricultural monitoring data within a fog architecture. The implementation is presented as a web application with an interactive user interface, visualisation tools, predictive model interpretation tools, integration with external agricultural information systems, and user notification mechanisms. This chapter also identifies the main limitations of the implemented software and algorithmic provision and substantiates directions for further development and practical application of the proposed solutions.

The conclusions summarise the main research results and present final statements regarding the achievement of the study aim and the completion of the research tasks.

The appendices include a list of the author's scientific publications related to the dissertation topic, scanned copies of documents confirming the implementation and practical application of the research results, software components for the end-to-end intellectualised transformation of agricultural monitoring data, and the web application for crop yield prediction.

The scientific novelty of the obtained results lies in the development of new methods and the further advancement of existing methods and models for the end-to-

end intellectualised transformation of spatio-temporal agroclimatic monitoring data. These results were achieved through the development and validation of a fog-based architecture for software and algorithmic provision for crop yield prediction, approaches to formalising multilevel aggregation and intelligent analytics of agroclimatic monitoring data, and methodological foundations for designing agricultural software solutions using machine learning and explainable artificial intelligence.

The practical significance of the obtained results lies in the implementation of software and algorithmic components, structural, functional and computer-based models that provide comprehensive processing of agroclimatic monitoring data for decision support in crop yield prediction. In particular, these results were implemented as a web application with an interactive interface, visualisation tools, predictive model interpretation, and the possibility of integration with external agricultural information systems.

The priorities for further research, identified through a critical analysis of the obtained results, include expanding the functional capabilities and improving the technological readiness of the developed software and algorithmic provision for the intellectualised transformation of agroclimatic monitoring data.

Keywords: *machine learning, artificial intelligence, digitalisation, software, prediction, crop yield, agricultural monitoring, spatio-temporal data, model, web application, fog architecture.*

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати:

1. Семенов С., Лактіонов І. Структурно-алгоритмічна формалізація програмного забезпечення комплексної трансформації просторово-часових даних агромоніторингу. *Вісник Хмельницького національного університету. Серія: Технічні науки*, 2025, Вип. 353 (3.2). С. 469–474. doi.org/10.31891/2307-5732-2025-353-64. *(Стаття в науковому фаховому періодичному виданні України. Особистий внесок здобувача Семенова С.: обґрунтування та реалізація підходів до формалізації програмного забезпечення, синтез та опис структурно-алгоритмічного забезпечення, написання тексту статті, аналіз одержаних результатів і формулювання висновків. Внесок Лактіонова І.: постановка дослідного завдання та редагування тексту статті).*

2. Лактіонов І., Семенов С. Компаративний аналіз програмно-алгоритмічних рішень інтелектуалізованого моніторингу в рослинництві відкритого ґрунту: сучасний стан і перспективи розвитку. *Вимірювальна та обчислювальна техніка в технологічних процесах*, 2025, Вип. 82 (2). С. 372–379. doi.org/10.31891/2219-9365-2025-82-54. *(Стаття в науковому фаховому періодичному виданні України. Особистий внесок здобувача Семенова С.: критичний категоріальний аналіз і логічна систематизація характеристик програмно-алгоритмічних рішень інтелектуалізованого агромоніторингу, написання тексту статті, узагальнення одержаних результатів і формулювання висновків. Внесок Лактіонова І.: постановка дослідного завдання, пошук інформаційно-аналітичних джерел та редагування тексту статті).*

3. Лактіонов І., Семенов С. Результати досліджень механізмів системної інтеграції та кросрівневої взаємодії програмно-алгоритмічних компонент прогнозування врожайності агрокультур. *Наукові праці ДонНТУ. Серія: «Інформатика, кібернетика та обчислювальна техніка»*, 2026, № 1 (42) ' 2026. С. 82–91. doi.org/10.31474/1996-1588-2026-1-42-82-91. *(Стаття в науковому фаховому періодичному виданні України. Особистий внесок здобувача*

Семенова С.: розробка і валідація підходів до системної інтеграції та механізмів кросрівневої взаємодії реалізованих програмно-алгоритмічних компонент, написання тексту статті, візуалізація і аналіз одержаних результатів, формулювання висновків. Внесок Лактіонова І.: постановка дослідного завдання, пошук інформаційно-аналітичних джерел та редагування тексту статті).

4. Laktionov I., Semenov S., Diachenko G., Rutkowski L. An end-to-end ML and XAI-based framework for crop yield prediction under agroclimatic variability. *Applied Soft Computing Journal*, 2026, Vol. 200. P. 1–25. doi.org/10.1016/j.asoc.2026.115394. **(Стаття в науковому періодичному закордонному виданні, що індексується в наукометричних базах Scopus та WoS – кuartиль Q1. Особистий внесок здобувача Семенова С.: методологія, програмне забезпечення, формальний аналіз, дослідження, написання первинного варіанту статті, візуалізація. Внесок Лактіонова І.: розробка концепції, наукове керівництво, забезпечення ресурсами, редагування статті. Внесок Дяченка Г.: валідація, обробка даних, дослідження, редагування статті. Внесок Рутковського Л.: концептуалізація, управління проектом, редагування статті).**

Опубліковані праці апробаційного характеру:

5. Diachenko G., Laktionov I., Moroz D., Derzhevetska M., Semenov S. A computerized method for predicting the risk of powdery mildew in wheat based on software analysis of soil and climatic monitoring data. In: *1st International Workshop on Advanced Applied Information Technologies (AdvAIT-2024)*, Khmelnytskyi, Ukraine – Zilina, Slovakia, December 05, 2024, in CEUR workshop proceedings, 2025, Vol. 3899. P. 172–185. URL: ceur-ws.org/Vol-3899/paper16.pdf. **(Публікація в матеріалах конференції, що індексується в наукометричній базі Scopus. Особистий внесок здобувача Семенова С.: розробка структурних і функціональних схем щодо реалізації комп'ютеризованого методу аналізу ґрунтокліматичних даних, розробка та валідація програмних компонент комп'ютерної моделі, аналіз одержаних результатів досліджень. Внесок Дяченка Г.: реалізація та тестування комп'ютерної моделі, візуалізація**

результатів досліджень, написання тексту статті, формулювання висновків. Внесок Лактіонова І.: постановка задачі досліджень, коригування підходів до досліджень. Внесок Мороза Д.: аналіз відомих наукових результатів та локалізація дослідних завдань. Внесок Держевецької М.: пошук і попередня обробка даних ґрунтокліматичного моніторингу, редагування тексту статті).

6. Лактіонов І., Дяченко Г., Семенов С. Компаративний синтез туманної архітектури програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу // *Програмне та апаратне забезпечення в інформаційних технологіях: матеріали Міжнар. наук.-практ. конф. молодих вчених та студентів* (6 травня 2025 р, Луцьк). Луцьк: ЛНТУ, 2025. С. 106–109. URL: https://drive.google.com/drive/folders/1BaojxLgSvEj-o0o_90OCXukJx5NBdzbo. (**Тези доповіді.** Особистий внесок здобувача Семенова С.: розробка та опис структурно-функціонального забезпечення туманної архітектури програмно-алгоритмічних засобів інтелектуалізованої трансформації даних агромоніторингу, написання тексту тез. Внесок Лактіонова І.: постановка дослідного завдання, коригування тексту тез. Внесок Дяченка Г.: уточнення напрямків досліджень, коригування тексту тез).

7. Лактіонов І., Семенов С., Дяченко Г., Євстрат'єв М. Результати критичного аналізу та систематизації програмно-алгоритмічних рішень інформаційно-орієнтованого прогнозування врожайності сільськогосподарських культур // *«Наукова весна» 2025: матеріали XV Міжнародної науково-технічної конференції аспірантів та молодих вчених* (26–28 березня 2025 р., Дніпро). Дніпро: НТУ «ДП», 2025. С. 196–197. URL: https://rmv.nmu.org.ua/ua/arkhiv-zbirok-konferentsiy/naukova-vesna-2025/Scientific_Spring_2025.pdf. (**Тези доповіді.** Особистий внесок здобувача Семенова С.: написання тексту тез, графічна інтерпретація результатів досліджень, формулювання перспектив подальших досліджень. Внесок Лактіонова І.: постановка дослідного завдання. Внесок Дяченка Г.: коригування тексту тез, аналіз одержаних результатів. Внесок Євстрат'єва М.: підбір та апіорний аналіз літературних джерел).

8. Семенов С.Ю., Лактіонов І.С. Архітектура програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур на основі технології туманних обчислень // *Молодь: наука та інновації: матеріали XIII Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених: Том 2* (Дніпро, 12–14 листопада 2025 р.). Дніпро: НТУ «Дніпровська політехніка», 2025. С. 124–125. URL: <https://rmv.nmu.org.ua/ua/arkhiv-zbirok-konferentsiy/molod-nauka-ta-innovatsii-2025/molod-2025-vol2.pdf>. (**Тези доповіді.** Особистий внесок здобувача Семенова С.: написання тексту тез, розробка та аналіз архітектури досліджуваного програмно-алгоритмічного забезпечення. Внесок Лактіонова І.: постановка дослідного завдання, коригування тексту тез).

9. Семенов С.Ю., Лактіонов І.С., Дяченко Г.Г. Результати розробки вебзастосунку прогнозування врожайності агрокультур із інтеграцією алгоритмів пояснювального штучного інтелекту // *Проблеми використання інформаційних технологій в освіті, науці та промисловості: XX міжнар. конф.* (5 грудня 2025 р., м. Дніпро): зб. наук. пр. Дніпро: НТУ «ДП», 2025. №10. С. 56–59. URL: https://pzks.nmu.org.ua/ua/science/2025_fin.pdf. (**Тези доповіді.** Особистий внесок здобувача Семенова С.: розробка і опис діаграм функціонування вебзастосунку прогнозування врожайності, аналіз одержаних результатів, формулювання наукової новизни і висновків. Внесок Лактіонова І.: постановка дослідного завдання, коригування тексту тез. Внесок Дяченка Г.: уточнення мети досліджень, коригування тексту тез).

10. Лактіонов І., Семенов С. Вебзастосунок інтелектуалізованого аналізу врожайності сільськогосподарських культур у мінливих агрокліматичних умовах // *IX Міжнародна науково-практична конференція «Кліматичні зміни та сільське господарство. Виклики для аграрної науки та освіти»* (Київ, 24 березня 2026 р.). Київ: НМЦ ВФПО, 2026. С. 9–11. URL: <https://cloud.nmc-vfpo.gov.ua/s/ggaS6Fe72sbs2A9?dir=/&editing=false&openfile=true>. (**Тези доповіді.** Особистий внесок здобувача Семенова С.: розробка і опис вебзастосунку прогнозування врожайності, аналіз одержаних результатів, написання тез. Внесок Лактіонова І.: постановка задачі досліджень, коригування тез).

ЗМІСТ

	Стор.
ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	15
ВСТУП.....	17
РОЗДІЛ 1. СУЧАСНИЙ СТАН ПРОГРАМНО-АЛГОРИТМІЧНИХ РІШЕНЬ ІЗ ЦИФРОВІЗАЦІЇ ТА ІНТЕЛЕКТУАЛІЗАЦІЇ РОСЛИННИЦТВА ВІДКРИТОГО ҐРУНТУ.....	25
1.1 Аналіз та узагальнення світових та вітчизняних трендів провадження рослинництва відкритого ґрунту.....	25
1.2 Аналіз відомих підходів і моделей опрацювання даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур.....	35
1.3 Аналіз відомих методів і програмно-алгоритмічних засобів інтелектуалізованої трансформації даних агромоніторингу.....	44
1.4 Обґрунтування мети та завдань досліджень дисертаційної роботи.....	56
1.5 Висновки за першим розділом.....	58
РОЗДІЛ 2. ФОРМАЛІЗАЦІЯ ПРОЦЕСУ КОМПЛЕКСНОЇ ІНТЕЛЕКТУАЛІЗОВАНОЇ ТРАНСФОРМАЦІЇ ЧАСОВО- ПРОСТОРОВИХ ДАНИХ АГРОМОНІТОРИНГУ.....	61
2.1 Компаративне обґрунтування туманної архітектури досліджуваного програмно-алгоритмічного забезпечення.....	61
2.2 Структурно-функціональна модель досліджуваного програмно- алгоритмічного забезпечення.....	66
2.3 Базові модельні діаграми досліджуваного програмно- алгоритмічного забезпечення.....	71

2.4 Інформаційна модель досліджуваного програмно-алгоритмічного забезпечення.....	77
2.5 Висновки за другим розділом.....	88

РОЗДІЛ 3. ПРОГРАМНІ КОМПОНЕНТИ ПОВНОЦИКЛОВОЇ ТРАНСФОРМАЦІЇ ДАНИХ АГРОМОНІТОРИНГУ ПІД ЧАС ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР.....	90
3.1 Підходи до розробки програмних компонент повноциклової трансформації даних агромоніторингу.....	90
3.2 Програмні компоненти збору та попередньої обробки даних.....	93
3.3 Програмні компоненти розвідувального аналізу даних.....	103
3.4 Програмні компоненти предикативного аналізу даних.....	117
3.5 Програмні компоненти інтерпретації моделей прогнозування врожайності.....	128
3.6 Потокова модель програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур.....	139
3.7 Висновки за третім розділом.....	141

РОЗДІЛ 4. КОМП'ЮТЕРНА ТЕХНОЛОГІЯ КРОСРІВНЕВОЇ ВЗАЄМОДІЇ ПРОГРАМНО-АЛГОРИТМІЧНИХ КОМПОНЕНТ ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ АГРОКУЛЬТУР ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ.....	144
4.1 Опис застосованих компонент та ролей імпортованих модулів.....	144
4.2 Архітектура програмного забезпечення та її основні функціональні сценарії.....	147
4.3 Результати розробки інтерфейсу користувача.....	154
4.4 Аналіз обмежень реалізації та перспективи подальших досліджень.....	159
4.5 Висновки за четвертим розділом.....	162

ВИСНОВКИ.....	164
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	167
ДОДАТОК А. Перелік публікацій за темою дисертації.....	185
ДОДАТОК Б. Документи, які підтверджують використання та впровадження одержаних результатів досліджень.....	188
ДОДАТОК В. Програмне забезпечення прогнозування врожайності сільськогосподарських культур (.irupb скрипт).....	191
ДОДАТОК Г. Програмне забезпечення серверної та клієнтської частин вебзастосунку прогнозування врожайності сільськогосподарських культур.....	208

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

AI – artificial intelligence (штучний інтелект)

API – application programming interface (прикладний програмний інтерфейс)

APSIM – agricultural production systems simulator (система моделювання сільськогосподарських виробничих систем)

ARIMA – autoregressive integrated moving average (авторегресійна інтегрована модель ковзного середнього)

CNN – convolutional neural network (згортова нейронна мережа)

CSRF – cross-site request forgery (міжсайтова підробка запиту)

CSS – cascading style sheets (каскадні таблиці стилів)

DSSAT – decision support system for agrotechnology transfer (система підтримки прийняття рішень для трансферу агротехнологій)

EDA – exploratory data analysis (розвідувальний аналіз даних)

ELI5 – explain like I'm five (поясни мені так, ніби мені п'ять років)

FAO – Food and agriculture organization (Продовольча та сільськогосподарська організація)

GDD – growing degree day (сума активних температур)

HMAC – hash-based message authentication code (код автентифікації повідомлення на основі хешування)

IoT – Internet of things (Інтернет речей)

JSON – JavaScript object notation (об'єктна нотація JavaScript)

JWT – JSON web token (вебтокен JSON)

KNN – k-nearest neighbors (метод k-найближчих сусідів)

LAI – leaf area index (індекс листкової поверхні)

LIME – local interpretable model-agnostic explanations (локальні інтерпретовані модельно-незалежні пояснення)

LLM – large language model (велика мовна модель)

LSTM – long short-term memory (довга короткострокова пам'ять)

MAE – mean absolute error (середня абсолютна помилка)

MAPE – mean absolute percentage error (середня абсолютна відсоткова помилка)

MDI – mean decrease in impurity (середнє зменшення неоднорідності)

MI – mutual information (взаємна інформація)
 ML – machine learning (машинне навчання)
 MLP – multilayer perceptron (багатошаровий перцептрон)
 MVC – model-view-controller (модель-вигляд-контролер)
 NDVI – normalized difference vegetation index (нормалізований відносний вегетаційний індекс)
 NOAA – National oceanic and atmospheric administration (Національне управління океанічних та атмосферних досліджень)
 ORM – object-relational mapping (об'єктно-реляційне відображення)
 PSO – particle swarm optimization (метод оптимізації на основі рою часток)
 R^2 – коефіцієнт детермінації
 RAG – retrieval-augmented generation (генерація з доповненням пошуком)
 RBF – radial basis function (радіально-базисна функція)
 RMSE – root mean square error (середньоквадратична помилка)
 SARIMA – seasonal autoregressive integrated moving average (сезонна авторегресійна інтегрована модель ковзного середнього)
 SVR – support vector regression (регресія опорних векторів)
 SVM – support vector machine (машина опорних векторів)
 TLS – transport layer security (протокол безпеки транспортного рівня)
 WHO – World health organization (Всесвітня організація охорони здоров'я)
 WOFOST – World food studies simulation model (імітаційна модель світових продовольчих досліджень)
 XAI – explainable artificial intelligence (пояснювальний штучний інтелект)
 XGBoost – eXtreme gradient boosting (екстремальний градієнтний бустинг)
 БД – база даних
 БПЛА – безпілотний літальний апарат
 ЄС – Європейський союз
 НВ – навчальна вибірка
 с/г – сільське господарство
 ТВ – тестова вибірка
 ФАР – фотосинтетично активна радіація

ВСТУП

Актуальність теми дослідження цієї дисертації зумовлена зростанням ролі комп'ютерних і інформаційно-аналітичних сервісних технологій у забезпеченні сталого розвитку сільськогосподарського виробництва в умовах мінливості агрокліматичних умов і техніко-технологічних факторів, обмеженості ресурсів та підвищених вимог до ефективності агротехнологічних рішень. Ефективність та продуктивність процесів вирощування агрокультур в умовах відкритого ґрунту визначається сукупним впливом динамічних ґрунтокліматичних процесів, біологічних особливостей агрокультур і технологічних прийомів їх вирощування, тоді як сучасні інформаційні технології з інтегрованими програмними засобами інтелектуальної прогнозної трансформації даних відіграють важливу роль під час планування й реалізації агротехнічних заходів зі збереження врожаю протягом всього вегетаційного циклу.

Враховуючи розвиненість сучасних підходів до агромоніторингу, зокрема прогнозування врожайності сільськогосподарських культур, варто акцентувати увагу на тому, що існуючі програмно-алгоритмічні рішення попри їх значну функціональність здебільшого не забезпечують комплексної інтелектуалізованої трансформації часово-просторових даних із одночасним урахуванням динамічності агрокліматичних умов, типів агрокультур і практик їх захисту, в межах децентралізованої архітектури з автоматизованою підтримкою прийняття агротехнічних рішень. Одночасно з цим світовий науково-практичний досвід підтверджує високу ефективність застосування цифровізованих і інтелектуалізованих рішень агромоніторингу, зокрема під час прогнозування врожайності агрокультур на основі збору, багаторівневої трансформації та аналітики часово-просторових ґрунтокліматичних даних із забезпеченням автоматизованої підтримки прийняття рішень щодо підвищення продуктивності вирощування сільськогосподарських культур.

Науково-прикладні дослідження, які проведено Снитюк В.Є, Вовна О.В., Гнатушенко В.В., Дяченко Г.Г., Каштан В.Ю., Morales A., Rokade A. та іншими

науковцями, присвячено розв'язанню задач інтелектуалізації та комп'ютеризації агротехнічних процесів, зокрема із використанням сучасних технологій на основі штучного інтелекту та машинного навчання. Під час розробки та дослідження інформаційних і комп'ютерних технологій агротехнічного призначення автори акцентують увагу на перспективності та доцільності врахування передового світового досвіду з системної реалізації повного циклу багатоетапної обробки даних агромоніторингу в межах повноциклової децентралізованої програмно-алгоритмічної архітектури із залученням інтелектуальних алгоритмів.

У цьому контексті важливою й актуальною є **науково-практична задача**, яка полягає в подальшому розвитку інформаційних і комп'ютерних технологій агротехнічного призначення завдяки розробці та використанню програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури під час прогнозування врожайності сільськогосподарських культур із урахуванням значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту в конкретних агротехнологічних умовах, що визначає наукову та практичну значущість цього дисертаційного дослідження.

Зв'язок роботи з науковими програмами, планами, темами. Тема дисертаційної роботи відповідає вітчизняним напрямкам стратегічного розвитку, які визначено в документах «Стратегія цифрового розвитку інноваційної діяльності України (WINWIN) до 2030 року», «Закон України «Про пріоритетні напрями розвитку науки і техніки», «Постанова Кабінету Міністрів України «Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 31 грудня року, наступного після припинення або скасування воєнного стану в Україні», а також секторальній стратегії розвитку ЄС «Common agricultural policy». Дослідження цієї дисертації виконано в межах науково-дослідної роботи Національного технічного університету «Дніпровська політехніка» «Методи, моделі та технології обробки і аналізу даних в комп'ютеризованих системах» (№ державної реєстрації 0124U004583).

Основною **метою** цієї дисертаційної роботи є вдосконалення інформаційно-орієнтованого підходу до прогнозування врожайності сільськогосподарських культур завдяки розробці та використанню програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу. Для досягнення поставленої мети були поставлені та вирішені такі **завдання**:

- проведення комплексного аналізу й логічного узагальнення відомих науково-прикладних розробок, які характеризують сучасний стан програмно-алгоритмічних рішень із цифровізації та інтелектуалізації рослинництва відкритого ґрунту, із формулюванням пріоритетів досліджень цієї дисертації;

- обґрунтування структурно-функціональної організації та інформаційної моделі досліджуваного програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур на основі туманної архітектури;

- розробка та дослідження ефективності програмно-алгоритмічних засобів комплексної трансформації даних агромоніторингу, які реалізують етапи попередньої обробки, інтелектуальної аналітики, а також інтерпретації та пояснення предикативних моделей із можливістю автоматизованої підтримки прийняття агротехнічних рішень;

- створення та валідація комп'ютерної технології, яка в межах єдиної архітектури реалізує комплексну трансформацію результатів агромоніторингу на основі розробленого програмно-алгоритмічного забезпечення прогнозування врожайності сільськогосподарських культур;

- проведення критичного аналізу отриманих результатів щодо розробки та дослідження програмно-алгоритмічного забезпечення агротехнічного призначення з прогнозування врожайності сільськогосподарських культур, а також обґрунтування подальших напрямків перспективних досліджень із розвитку та практичного використання розробки.

Об'єкт дослідження – інформаційні процеси програмної трансформації даних агромоніторингу, що протікають під час прогнозування врожайності

сільськогосподарських культур.

Предмет дослідження – методи, засоби й моделі створення та валідації програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур.

Методи дослідження. Під час виконання дисертаційного дослідження використано як загальнонаукові, так і спеціальні методи, які спрямовано на досягнення основної мети та розв'язання поставлених завдань, зокрема: порівняльний аналіз та логічне узагальнення; декомпозиція та формалізація інформаційних процесів; комп'ютерне, структурно-функціональне та інформаційне моделювання; методи статистичного і розвідувального аналізу даних; методи пояснювального штучного інтелекту та машинного навчання; об'єктно-орієнтоване програмування; комп'ютерний експеримент та архітектурний синтез децентралізованих вебзастосунків.

Наукова новизна одержаних результатів. Під час виконання дисертаційного дослідження створено нові, вдосконалено та розвинуто відомі методи, моделі та програмно-алгоритмічні компоненти повноциклової інтелектуалізованої трансформації часово-просторових даних агромоніторингу задля прогнозування врожайності сільськогосподарських культур, а саме:

1. Уперше синтезовано повноциклову туманну архітектуру програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур, що, на відміну від відомих рішень, комплексно враховує вплив агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту, а також реалізує наскрізну логіку виконання процедур агрегування, інтелектуалізованого опрацювання даних та підтримки прийняття рішень, які об'єктивно розподілені за функціональними рівнями, що дозволяє підвищити ефективність і масштабованість процесу трансформації результатів агромоніторингу.

2. Дістали подальшого розвитку концептуально-методологічні засади формалізації багатоетапної інтелектуалізованої трансформації результатів агромоніторингу, які відрізняються від відомих тим, що комплексно враховують

просторово-часову гетерогенність агрокліматичних даних, а також механізми людино-машинної та кросрівневої взаємодії компонент розробленого програмно-алгоритмічного забезпечення, що дозволило обґрунтувати підходи до побудови та валідації комп'ютерної технології прогнозування врожайності агрокультур.

3. Дістав подальшого розвитку інформаційно-орієнтований підхід до прогнозування врожайності агрокультур завдяки комплексній програмній реалізації процедур попередньої обробки, розвідувального аналізу і прогнозної аналітики даних та інтерпретації предикативних моделей у межах децентралізованої архітектури, що дозволило підвищити точність та адекватність прогнозування врожайності агрокультур шляхом досягнення відносної похибки прогнозування $\pm 7,8 \%$ і коефіцієнта детермінації 0,9904 у широкому діапазоні варіювання типів агрокультур і просторово-часових умов їх вирощування.

4. Дістали подальшого розвитку методи автоматизованої інтерпретації результатів прогнозної аналітики в комп'ютерних технологіях агромоніторингу завдяки програмній реалізації та інтеграції алгоритмів пояснювального штучного інтелекту до процесу багаторівневої трансформації даних, що, на відміну від відомих методів, забезпечує як глобальну оцінку впливу агрокліматичних факторів на врожайність агрокультур, так і локальне пояснення результатів прогнозів, підвищуючи прозорість і об'єктивність підтримки прийняття рішень.

5. Удосконалено теоретико-прикладні підходи до створення й застосування програмно-алгоритмічних рішень прогнозного агромоніторингу шляхом розробки та валідації вебзастосунку, який реалізує логіку на основі запитів під час трансформації часово-просторових даних у туманній архітектурі, що, на відміну від відомих рішень, одночасно поєднує децентралізовану предикативну аналітику й інтерпретацію, забезпечує сумісність із агроінформаційними системами та реалізує механізми оперативного сповіщення користувачів, що дозволяє підвищити практичну ефективність використання результатів прогнозування врожайності агрокультур під час управління агротехнологічними процесами.

Практичне значення одержаних результатів. У цій дисертації запропоновано, реалізовано та валідовано програмно-алгоритмічні компоненти,

комп'ютерні моделі, архітектурні рішення та способи функціонування інтелектуалізованих комп'ютерних технологій повноциклової трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур, які сукупно забезпечують такі положення практичного ефекту:

1. Розроблено структурно-компонентне забезпечення та базові модельні діаграми програмно-алгоритмічного рішення прогнозування врожайності агрокультур з урахуванням функціональних особливостей туманної архітектури.

2. Реалізовано та валідовано програмні компоненти багаторівневої агрегації, обробки й предикативної аналітики кліматичних і агротехнологічних даних із використанням методів машинного навчання та пояснювального штучного інтелекту під час прогнозування врожайності агрокультур.

3. Створено та протестовано вебзастосунок із користувацьким інтерактивним інтерфейсом, який забезпечує візуалізацію результатів агромоніторингу, інтерпретацію прогнозних моделей, сумісність із зовнішніми агроінформаційними системами через програмні інтерфейси API та механізми оперативного сповіщення користувачів у фоновому режимі.

4. Запропоновано децентралізовану архітектуру програмно-алгоритмічного забезпечення предикативного агромоніторингу, що підвищує автономність і масштабованість використання результатів прогнозування врожайності в умовах мінливих агрокліматичних факторів.

Практичну цінність результатів досліджень, які одержано в цій дисертації, підтверджено використанням у науково-дослідній і виробничій діяльності Донецької державної сільськогосподарської дослідної станції НААН України. Ці результати було використано як засоби підтримки прийняття рішень під час планування і виконання агропроцедур, пов'язаних із вирощуванням і селекцією сільськогосподарських культур. Також результати досліджень упроваджено в освітній процес НТУ «Дніпровська політехніка» під час підготовки здобувачів за третім (освітньо-науковим) рівнем вищої освіти за спеціальностями галузі F Інформаційні технології.

Особистий внесок здобувача. Всі науково-практичні результати досліджень, що відображають наукову новизну й практичну значущість цієї дисертації, отримано автором особисто. Результати дисертаційної роботи висвітлено в наукових публікаціях, які зазначено в додатку А. Ідеї співавторів наукових публікацій в цій дисертації не були використані. В наукових працях, які написано й опубліковано в співавторстві, дисертанту належить: [1] – обґрунтування та реалізація підходів до формалізації програмного забезпечення, синтез та опис структурно-алгоритмічного забезпечення, написання тексту статті, аналіз одержаних результатів і формулювання висновків; [2] – критичний категоріальний аналіз і логічна систематизація характеристик програмно-алгоритмічних рішень інтелектуалізованого агромоніторингу, написання тексту статті, узагальнення одержаних результатів і формулювання висновків; [3] – розробка і валідація підходів до системної інтеграції та механізмів кросрівневої взаємодії реалізованих програмно-алгоритмічних компонент, написання тексту статті, візуалізація і аналіз одержаних результатів, формулювання висновків; [4] – методологія, програмне забезпечення, формальний аналіз, дослідження, написання первинного варіанту статті, візуалізація; [5] – розробка структурних і функціональних схем щодо реалізації комп'ютеризованого методу аналізу ґрунтокліматичних даних, розробка та валідація програмних компонент комп'ютерної моделі, аналіз одержаних результатів досліджень; [6] – розробка та опис структурно-функціонального забезпечення туманної архітектури програмно-алгоритмічних засобів інтелектуалізованої трансформації даних агромоніторингу, написання тексту тез; [7] – написання тексту тез, графічна інтерпретація результатів досліджень, формулювання перспектив подальших досліджень; [8] – написання тексту тез, розробка та аналіз архітектури досліджуваного програмно-алгоритмічного забезпечення; [9] – розробка і опис діаграм функціонування вебзастосунку прогнозування врожайності, аналіз одержаних результатів, формулювання наукової новизни і висновків; [10] – розробка і опис вебзастосунку прогнозування врожайності, аналіз одержаних результатів, написання тез.

Апробація результатів досліджень дисертації. Науково-практичні результати досліджень цієї дисертації висвітлено та обговорено під час 1^й Міжнародний семінар з передових прикладних інформаційних технологій (Хмельницький, Україна; Жиліна, Словаччина, 2024), Міжнародна науково-практична конференція молодих вчених та студентів «Програмне та апаратне забезпечення в інформаційних технологіях» (Луцьк, 2025), XV Міжнародна науково-технічна конференція аспірантів та молодих вчених «Наукова весна 2025» (Дніпро, 2025), XIII Міжнародна науково-технічна конференція студентів, аспірантів та молодих вчених «Молодь: наука та інновації» (Дніпро, 2025), XX Міжнародна конференція «Проблеми використання інформаційних технологій в освіті, науці та промисловості» (Дніпро, 2025), IX Міжнародна науково-практична конференція «Кліматичні зміни та сільське господарство. Виклики для аграрної науки та освіти» (Київ, 2026), а також наукових заходів і семінарів, що відбувались на кафедрі програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка».

Публікації. За темою дисертації опубліковано 10 наукових праць, із яких: 1 стаття в журналі, що індексується в наукометричних базах Scopus і Web of Science (квартиль Q1); 3 статті в періодичних наукових виданнях України категорії Б; 6 тез доповідей у матеріалах міжнародних наукових конференцій, із яких 1 індексується в наукометричній базі Scopus.

Структура та обсяг дисертації. Дисертація викладена за наступною структурою: вступ, чотири основні розділи, загальні висновки, список використаних літературних джерел зі 177 найменувань та 4 додатки. Загальний обсяг дисертаційної роботи становить 222 сторінки, з яких основний текст – 150 сторінок, включаючи 44 рисунки та 27 таблиць. Додатки на 38 сторінках містять перелік наукових публікацій здобувача за темою дисертаційної роботи, документи, що підтверджують використання й упровадження результатів дисертації, а також програмне забезпечення повноциклової трансформації результатів агромоніторингу та вебзастосунку прогнозування врожайності сільськогосподарських культур.

РОЗДІЛ 1

СУЧАСНИЙ СТАН ПРОГРАМНО-АЛГОРИТМІЧНИХ РІШЕНЬ ІЗ ЦИФРОВІЗАЦІЇ ТА ІНТЕЛЕКТУАЛІЗАЦІЇ РОСЛИННИЦТВА ВІДКРИТОГО ҐРУНТУ

1.1 Аналіз та узагальнення світових та вітчизняних трендів провадження рослинництва відкритого ґрунту

В останні десятиріччя людство стикається з низкою глобальних проблем і викликів, що значно впливають на ефективність і продуктивність провадження сільськогосподарської (с/г) діяльності у світі, серед яких особливої уваги потребують такі: зміна клімату, динамічне зменшення природних ресурсів, деградація ґрунтів і зростання населення. З урахуванням цього на підставі аналізу релевантних статистичних даних, які оприлюднено профільними всесвітньо визнаними організаціями, встановлено наступне. Швидкість потепління за актуальними даними Національного управління океанічних та атмосферних досліджень (NOAA) за останні сорок років становить 0,2 °C на десятиріччя [1]. За даними, які висвітлено під час Світового економічного форуму в 2024 році, за останні двадцять п'ять років відносна частка споживання природних ресурсів (у тому числі біомаси) збільшувалась і продовжує збільшуватись у середньому на 2,3 % у рік [2]. У доповіді, що була опублікована за результатами досліджень Організацією об'єднаних націй (ООН) у 2022 році, зазначено, що до 40 % усіх ґрунтів у світі знаходяться в стані помірної або серйозної деградації, а також за прогнозами цей показник може зрости до 90 % до 2050 року, якщо буде продовжуватись вирубування лісів, урбанізація, надмірне використання хімікатів під час с/г діяльності та інші негативні практики [3]. За актуальними даними, в останні десятиріччя спостерігається стала динаміка збільшення населення планети, і за останні тридцять років відносна частка його збільшення становить від 0,84 % до 1,44 % на рік і на 01 січня 2026 р. становила 8,27 млрд та за прогнозами у 2050 р. становитиме 9,66 млрд [4]. У свою чергу, на підставі даних

Всесвітньої організації охорони здоров'я (WHO), встановлено, що в 2023 році близько 733 млн. людей у світі страждали від голоду [5]. Таким чином, все вищезазначене в сукупності створює значні ризики для глобальної продовольчої безпеки та обумовлює необхідність обґрунтування та реалізації дієвих підходів щодо їх зменшення та/або усунення.

У таких умовах, що склалися на сьогоднішній день через глобальні виклики, значно актуалізується роль сільського господарства під час забезпечення продовольчої безпеки на мікро- та макрорівнях, сталого економічного розвитку та раціонального використання природних ресурсів. Своєю чергою, сучасне сільське господарство потребує постійної модернізації та адаптації до вищеописаних дестабілізуючих чинників. Отже, однією з найбільш актуальних задач є наукове обґрунтування підходів до ефективного прогнозування врожайності с/г культур із урахуванням сукупного впливу екологічних, агрокліматичних, економічних і соціальних факторів. Вирішення цієї задачі набуває стратегічного значення, адже дозволить об'єктивно планувати аграрну діяльність у середньо- та довгостроковій перспективі, зменшити втрати врожаю і відповідні продовольчі ризики та оптимізувати використання ресурсів.

На глобальному рівні, зокрема, країн Європейського союзу (ЄС), аграрна галузь стикається з низкою факторів, що обумовлюють темпи її розвитку, які в тому числі пов'язані з динамічним попитом на органічну й екологічно чисту с/г продукцію та масштабністю впровадження цифрових і інформаційних технологій, що регламентуються положеннями Спільної сільськогосподарської політики ЄС [6]. Орієнтовний прибуток від виробництва с/г продукції тваринництва і рослинництва країнами ЄС у 2023 році становив 537,1 млрд євро, з яких 51 % припадає на продукцію галузі рослинництва [7], як у деталізованому вигляді показано на рисунку 1.1. Також варто зазначити, що сільське господарство є ключовою галуззю формування національних економік значної кількості країн світу, особливо країн Східної та Південно-Східної Європи, як показано на рисунку 1.2 [8]. Отже, пошук і реалізація ефективних підходів до модернізації аграрної галузі є потенційно перспективним і дієвим аспектом під час вирішення

значної кількості проблем економічного, соціального та екологічного характеру.

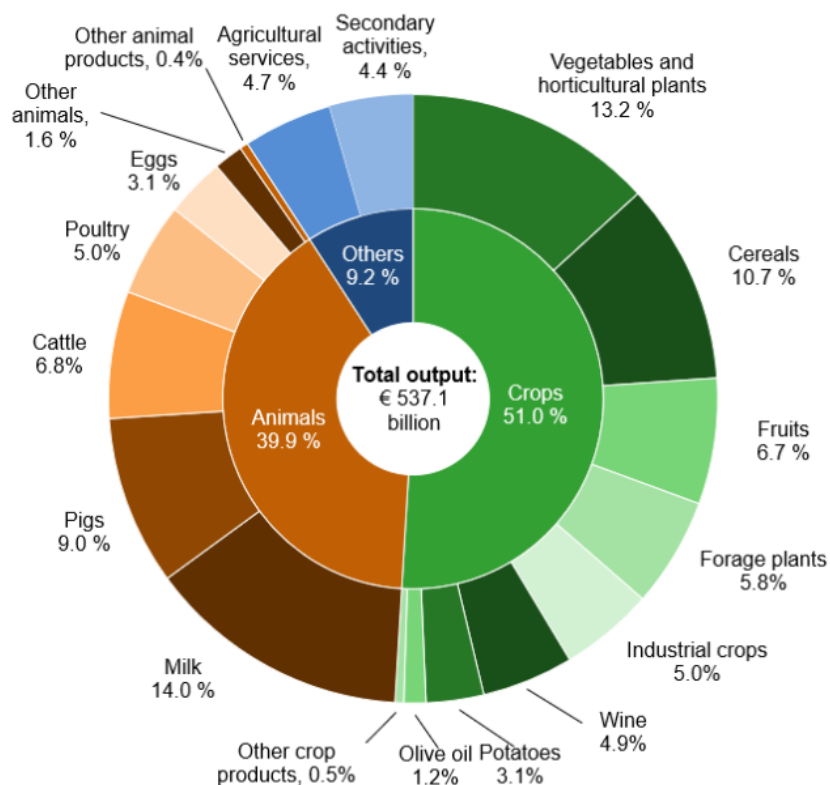


Рисунок 1.1 – Статистичні показники прибутку від провадження сільськогосподарської діяльності країнами ЄС у 2023 році [7]

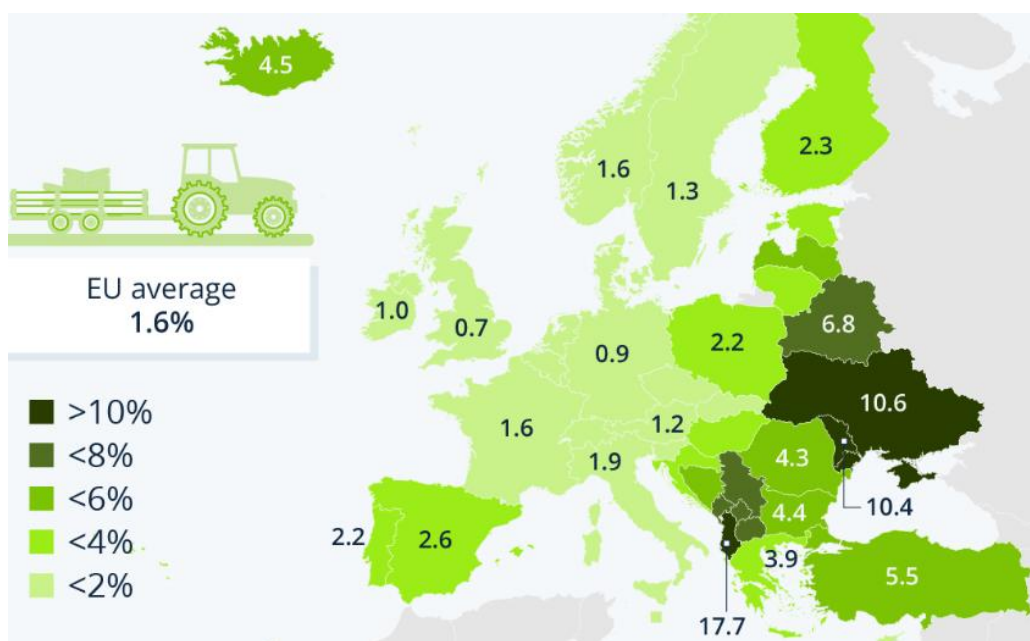


Рисунок 1.2 – Статистичні показники внеску сільськогосподарської галузі до національних економік країн у 2021 році [8]

Додатково важливість розробки та практичної реалізації підходів до модернізації агротехнічних практик рослинництва в світовому масштабі підтверджується статистичним аналізом показників обсягів вирощування та задіяних оброблюваних площ за даними Всесвітньої продовольчої та сільськогосподарської організації (FAO) за останнє десятиріччя до дати останньої актуалізації [9], як показано на рисунку 1.3.

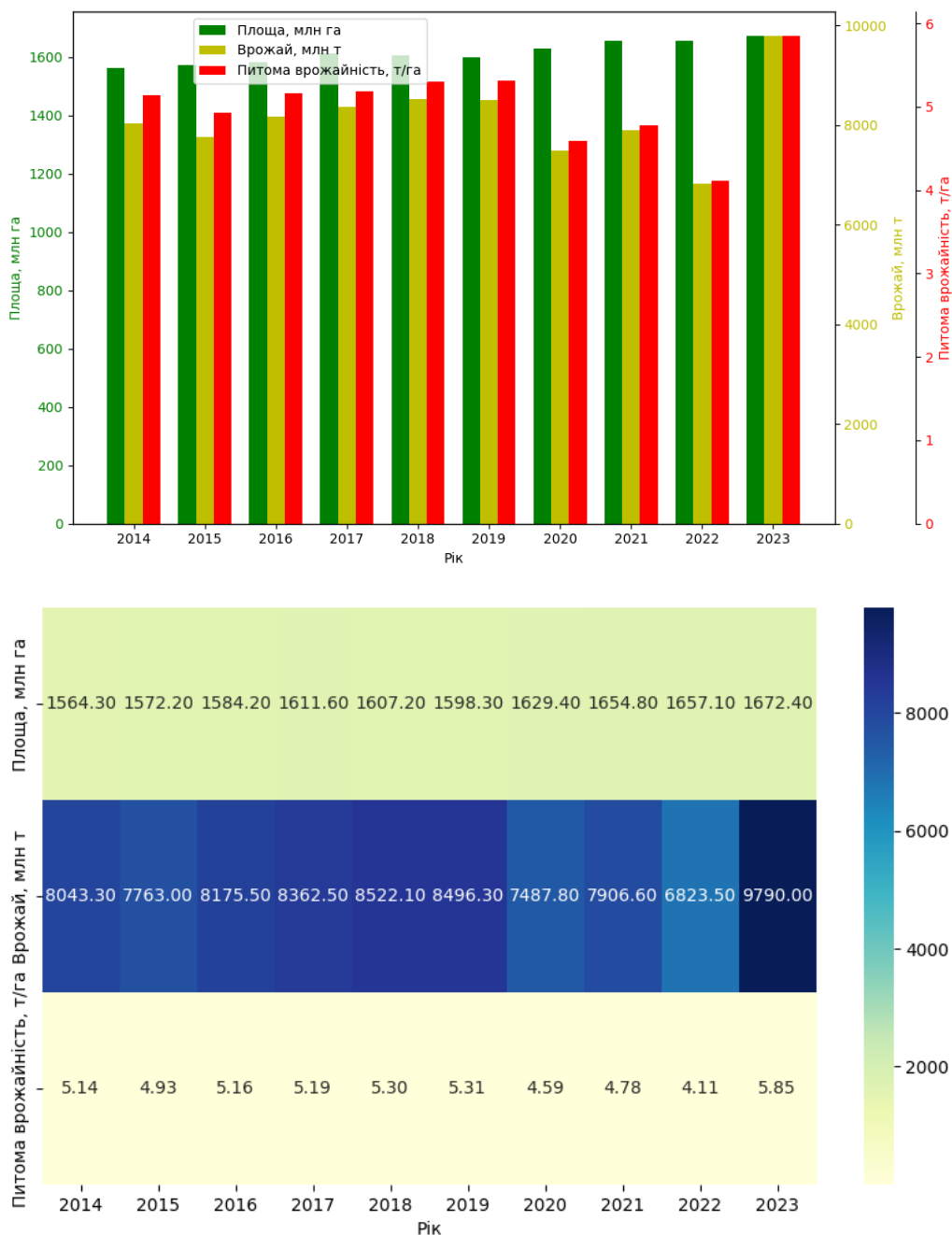


Рисунок 1.3 – Статистичні показники провадження сільськогосподарської діяльності у світі

У розрізі аналізу статистичних показників провадження с/г діяльності рослинництва особливу увагу необхідно сконцентрувати на аналогічних характеристиках вітчизняного аграрного сектору, адже окрім зазначених вище глобальних проблем, в Україні спостерігаються значні проблеми і виклики, які пов'язані зі зменшенням і забрудненням земельних ресурсів через повномасштабні воєнні дії. Результати статистичного аналізу даних FAO за останнє десятиріччя до дати останньої актуалізації [9] щодо провадження с/г діяльності в Україні наведено на рисунку 1.4.

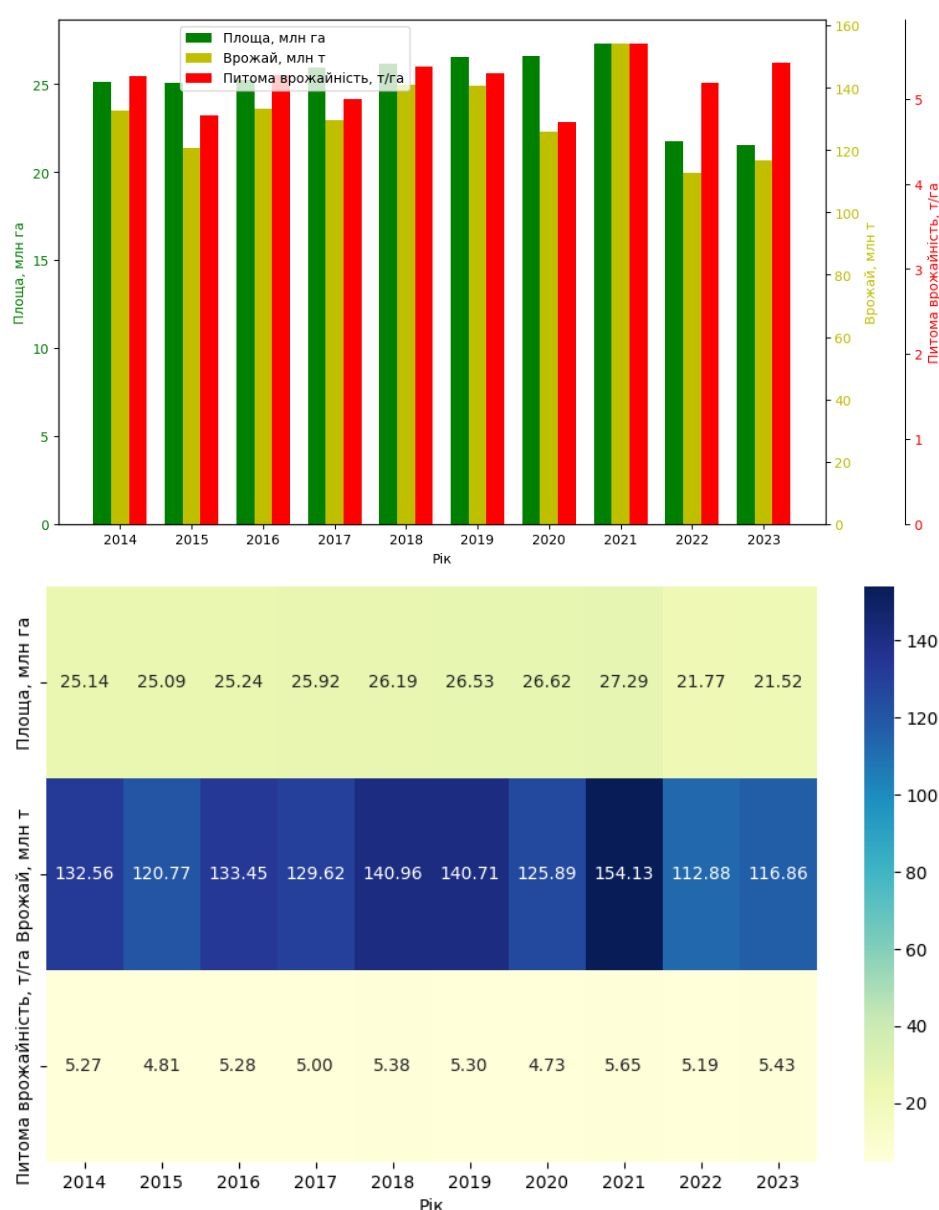


Рисунок 1.4 – Статистичні показники провадження сільськогосподарської діяльності в Україні

Також варто зазначити, що в Україні розвиненими є всі напрямки с/г діяльності із вирощування зернових, олійних, бобових, овочевих, плодових і ягідних культур. Це підтверджено статистичними даними, що викладені в інфографічному довіднику «Агробізнес України», який було створено контент-маркетинговою агенцією Top Lead і агромедіа-холдингом Latifundist Media за підтримки EFSE Development Facility і Credit Agricole Bank та фінансового сприяння ЄС у межах EU4Business [10], як показано на рисунках 1.5–1.7.

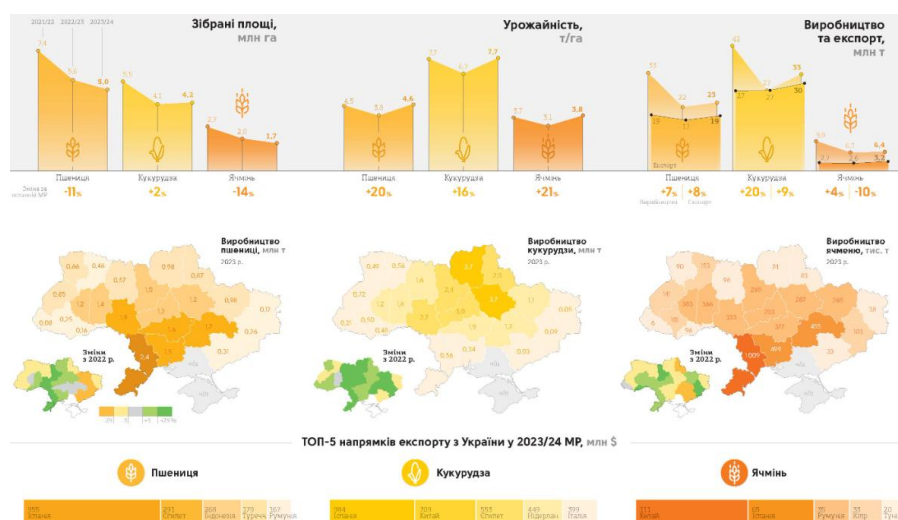


Рисунок 1.5 – Графічна інтерпретація статистичних показників виробництва основних зернових культур [10]

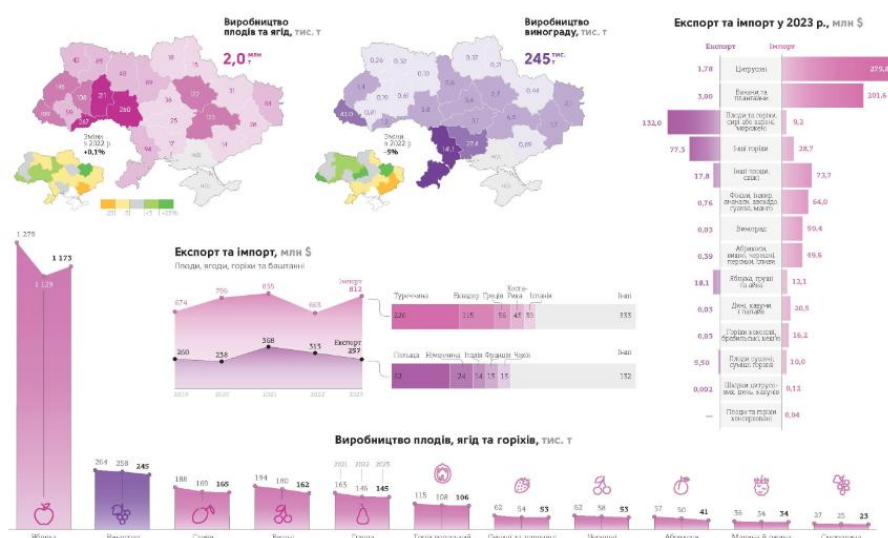


Рисунок 1.6 – Графічна інтерпретація статистичних показників виробництва основних плодових і ягідних культур [10]

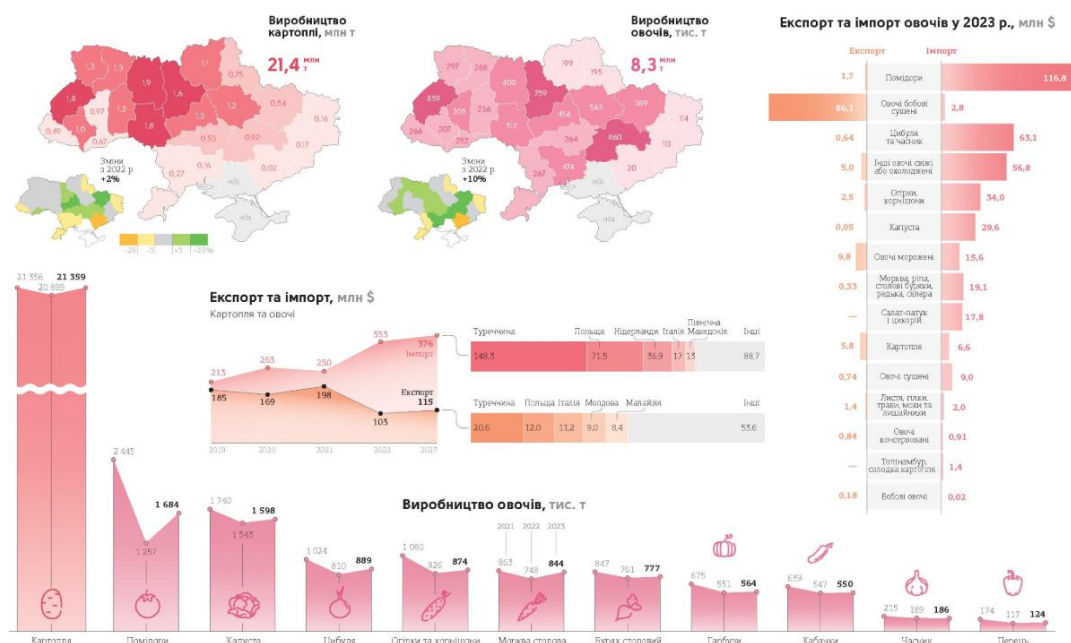


Рисунок 1.7 – Графічна інтерпретація статистичних показників виробництва основних овочевих культур [10]

Порівняльний аналіз статистичних даних щодо індикативних показників провадження с/г діяльності рослинництва на глобальному (див. рис. 1.3) та національному (див. рис. 1.4) рівнях дозволив встановити таке:

1. Оброблювані с/г площі у світі постійно зростають: за період із 2014 р. по 2023 р. відносний приріст становить 6,9 %. В Україні за період з 2014 р. по 2021 р. також спостерігалася позитивна динаміка оброблюваних с/г площ із граничним показником відносного приросту на рівні 8,6 %, проте в 2022 і 2023 рр. порівняно з 2021 р. спостерігається негативна динаміка – від –21,1 % до –20,2 %.

2. У показниках абсолютної та питомої врожайності як у світовому, так і національному масштабах не спостерігається сталої позитивної динаміки, як показано на рисунках 1.3 і 1.4. Це свідчить про значний сукупний вплив дестабілізуючих факторів (кліматичні умови, доступність водних ресурсів, наявність шкідників, своєчасність та ефективність планування й реалізації агротехнічних заходів та інше) на обсяги виробництва с/г продукції рослинництва, а також про певну кореляцію проблематики, що склалась в глобальному та вітчизняному сільському господарстві, та нагальну необхідність пошуку та реалізації науково обґрунтованих шляхів її вирішення.

Наведені вище результати узагальнення статистичних даних підтверджують актуальність розробки концептуально-методологічних основ та практичної реалізації програмно-алгоритмічних рішень, спрямованих на оптимізацію агротехнічних процесів під час вирощування с/г культур через об'єктивне планування та оперативне реагування на дестабілізуючі фактори щодо продуктивності вирощування с/г культур у динамічних агрокліматичних умовах та варіативних агротехнічних практиках, зокрема шляхом комплексного інтелектуального аналізу даних агромоніторингу програмними засобами.

Варто акцентувати увагу на тому, що на теперішній час розроблено та впроваджено до використання нормативно-аналітичні документи, які регламентують цілі сталого розвитку щодо підвищення ефективності с/г процесів вирощування, зокрема за рахунок використання цифрових і інформаційних технологій, як у деталізованому вигляді представлено в таблиці 1.1.

Таблиця 1.1 – Результати аналізу й систематизації нормативно-аналітичних документів зі сталого розвитку сільського господарства

Назва документу і масштаб розповсюдження	Цілі, що корелюють із розвитком і впровадженням інформаційних технологій	Посилання
G20 Agriculture Ministers Declaration (світовий)	Розвиток точного землеробства, оптимізація використання ресурсів, поширення знань у галузі використання інноваційних технологій та розвиток кліматично-орієнтованого сільського господарства	[11]
United Nations Sustainable Goals (світовий)	Підвищення продовольчої безпеки, розвиток кліматично-орієнтованого сільського господарства і модернізація с/г інфраструктури	[12]

Продовження табл. 1.1

E-agriculture strategy (світовий)	Підвищення продовольчої безпеки, боротьба зі шкідниками та хворобами с/г культур, розвиток кліматично-орієнтованого сільського господарства, модернізація с/г інфраструктури та впровадження стратегії інформаційно-орієнтованої трансформації аграрної галузі	[13]
World Bank's Digital Agriculture Initiative (світовий)	Розвиток точного землеробства, впровадження стратегії інформаційно-орієнтованої трансформації аграрної галузі, оптимізація використання ресурсів, покращення логістики постачання с/г продукції та поширення знань у галузі використання інноваційних технологій	[14]
EU Common Agricultural Policy (ЄС)	Розвиток цифрового сільського господарства, модернізація засобів автоматизації та комп'ютеризації агротехнічних виробництв, оптимізація використання ресурсів та розвиток кліматично-орієнтованого сільського господарства	[15]
EU Digital Strategy (ЄС)	Впровадження інноваційних інформаційних і комп'ютерних технологій до с/г підприємств різних масштабів і форм власності	[16]
Стратегія розвитку с/г та сільських територій в Україні до 2030 (Україна)	Розвиток цифрового сільського господарства, модернізація засобів автоматизації та комп'ютеризації агротехнічних виробництв, оптимізація використання ресурсів та поширення знань у галузі використання інноваційних технологій	[17]

Продовження табл. 1.1

Стратегія сталого розвитку України до 2030 року (Україна)	Забезпечення національної продовольчої безпеки та сталого розвитку с/г галузі	[18]
Стратегія розвитку сфери інноваційної діяльності на період до 2030 року (Україна)	Масштабне впровадження прикладних цифрових, інформаційних і комп'ютерних технологій, зокрема до аграрної галузі	[19]
Національна економічна стратегія 2030 (Україна)	Розвиток цифрового сільського господарства, модернізація засобів автоматизації та комп'ютеризації агротехнічних виробництв, покращення логістики постачання с/г продукції та поширення знань у галузі використання інноваційних технологій	[20]

Таким чином, на підставі аналізу та логічного узагальнення інформації, яку наведено в таблиці 1.1, можна стверджувати, що одним із основних трендів розвитку сучасного сільського господарства як у глобальному, так і в національному масштабах, є інформаційно-орієнтована трансформація завдяки розробці й упровадженню цифрових і комп'ютеризованих технологій агротехнічного призначення, що, в свою чергу, дозволяє оптимізувати ряд агротехнічних процедур, як наприклад, використання ресурсів, зменшення негативного впливу шкідників і хвороб с/г культур, покращення логістики під час виробництва та постачання с/г продукції, а також підвищення рівня продовольчої безпеки в цілому.

1.2 Аналіз відомих підходів і моделей опрацювання даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур

Як зазначалось вище, одним із основних напрямків інформаційно-орієнтованої трансформації сільського господарства є розробка та впровадження методів і засобів комп'ютеризованого планування агротехнічних процедур під час повного циклу вирощування. Це є можливим, у тому числі за рахунок реалізації програмно-алгоритмічних засобів трансформації даних агромоніторингу на основі інтелектуалізованих і комп'ютеризованих моделей прогнозування врожайності с/г культур. Адже, прогнозування врожайності с/г культур є критично важливим процесом, який допомагає планувати технологічні процедури і виробничу діяльність у цілому, оперативно й достовірно оцінювати індикативні показники, що впливають на продовольчу безпеку, а також приймати економічно та технологічно доцільні рішення.

За останні роки підходи до прогнозування врожайності с/г культур значно трансформувалися завдяки розвитку інформаційних і цифрових технологій, що націлені на підвищення точності, релевантності та оперативності прогнозів. Використання адаптивних математичних моделей, алгоритмів машинного навчання (ML) та штучного інтелекту (AI), мікрокомп'ютерних і сенсорних пристроїв, технологій дистанційного зондування для збору й обробки даних у реальному часі довело на практиці свою ефективність під час вирішення задач із планування, реалізації та адаптації прогресивних агротехнічних практик щодо підвищення врожайності с/г культур у мінливих агрокліматичних умовах [21–23]. Інтегральний соціально-економічний ефект від цифровізації підходів до прогнозування врожайності с/г культур полягає в покращенні механізмів управління сільським господарством як на рівні окремих господарств, так і на державному рівні, що позитивно впливає на забезпечення продовольчої безпеки [24, 25], як у деталізованому вигляді показано на рисунку 1.8.

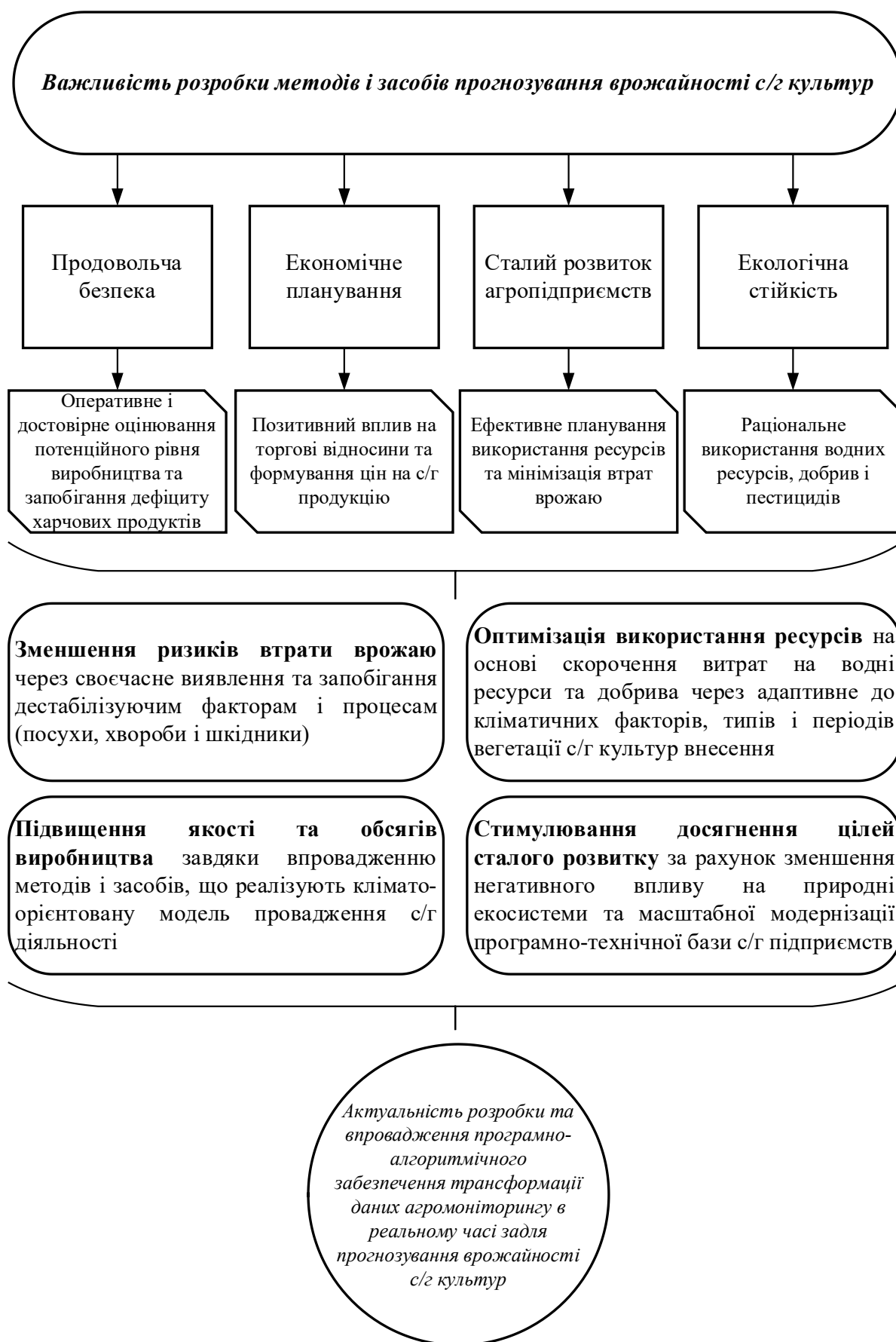


Рисунок 1.8 – Графічна інтерпретація соціально-економічного ефекту від цифровізації підходів до прогнозування врожайності с/г культур

На теперішній час підходи до прогнозування врожайності с/г культур є доволі розвиненими та базуються на різних фізико-хімічних і біологічних аспектах агротехнічних процедур, а також на значному спектрі використовуваних засобів збору даних та обчислювальних алгоритмах їх обробки. Результати систематизації відомих результатів досліджень [26–28] щодо критичного аналізу найбільш популярних підходів до прогнозування врожайності агрокультур наведено в таблиці 1.2.

Таблиця 1.2 – Відомі підходи до прогнозування врожайності сільськогосподарських культур

Підхід	Засади функціонування	Прикладний ефект
Статистичний	Базується на довгострокових масивах даних за попередні роки щодо врожайності, кліматичних параметрів, використання добрив, виникнення хвороб і шкідників	Дозволяє виявляти закономірності в масивах даних, які використовуються під час довгострокового планування та виконання агротехнічних процедур
Агрокліматичний	Базується на основі багатофакторного обліку погодних умов, фізико-хімічних характеристик ґрунту, сівозміни та виконуваних агротехнічних процедур	Дозволяє ефективно передбачати вплив ґрунтокліматичних змін на врожайність с/г культур і використовується під час планування посівів, зрошення та внесення добрив
Біофізичний	Ґрунтується на фізіологічних моделях росту с/г культур та їхній взаємодії з навколишнім середовищем	Дозволяє врахувати генетичні особливості с/г культур під час оцінки продуктивності їх вирощування

Продовження табл. 1.2

Інформаційно-орієнтований	Базується на основі використання інформаційних технологій збору даних (ІоТ-мережі, супутники, БПЛА) та методів штучного інтелекту під час аналізу вимірювальних даних	Дає змогу аналізувати значні масиви вимірювальних даних, які розподілені в часі й просторі, підвищує чутливість і точність прогнозів завдяки встановленню неявних закономірностей, дозволяє прогнозувати врожайність с/г культур із обліком значної кількості дестабілізуючих факторів
---------------------------	---	--

У свою чергу, підходи до прогнозування врожайності с/г культур, які наведено в таблиці 1.2, диференціюються на відповідні моделі, що враховують специфіку впливових факторів та обчислювальних алгоритмів, на яких вони базуються. Результати аналізу найбільш популярних моделей прогнозування врожайності с/г культур наведено в таблиці 1.3.

Таблиця 1.3 – Відомі моделі прогнозування врожайності сільськогосподарських культур

Моделі	Використовувані технології, методи та обчислювальні алгоритми	Посилання
Емпіричні	Функціональну основу становлять алгоритми регресії (лінійна, багатофакторна, нелінійна) та аналізу часових рядів результатів спостережень (ARIMA, SARIMA) щодо врожайності с/г культур та факторів, що на неї впливають, в основному без обліку біологічних показників зростання культур	[29–31]

Механістичні (агроекологічні)	Функціональну основу становлять закономірності щодо показників продуктивності вирощування с/г культур із урахуванням агрокліматичних і ґрунтових даних (DSSAT), агротехнологічних особливостей і кліматичних умов провадження с/г діяльності (APSIM), біологічного потенціалу вирощуваних культур (WOFOST), які використано в якості алгоритмічного забезпечення обчислювальних процедур	[32–34]
Геоінформаційні	В якості функціональної основи використовують супутникові знімки та дані картографування земної поверхні з урахуванням динаміки кліматичних параметрів під час прогнозування врожайності с/г культур на значних територіях. Наприклад, моделі на основі NDVI та LAI індексів, MODIS-GPP, Yield Prophet, CropScape та інші	[35–40]
Моделі на основі штучного інтелекту та машинного навчання	Функціональний базис являють алгоритми штучного інтелекту (LSTM, CNN, Transformers та ін.) і машинного навчання (XGBoost, Random Forest, Linear Regression та ін.), що використовуються під час встановлення закономірностей впливу значних масивів даних (агрокліматичні дані від сенсорів та БПЛА, супутникові знімки) на показники ефективності вирощування с/г культур з урахуванням фізико-хімічних властивостей ґрунту, кліматичних умов та біологічних і генетичних характеристик рослин	[41–46]

Комбіновані	Основою їх розгортання та функціонування є поєднання декількох видів моделей (наприклад, емпіричні та машинне навчання, агроекологічні та штучний інтелект, геоінформаційні та штучний інтелект або машинне навчання) задля врахування більшої кількості факторів з метою підвищення достовірності прогнозів і їх адаптації до динаміки агрокліматичних показників, біологічних особливостей і стадій вегетації рослин та виконуваних агротехнічних процедур	[47–50]
-------------	--	---------

На підставі проведеного аналізу та логічного узагальнення відомих підходів та моделей прогнозування врожайності с/г культур, які наведено в таблицях 1.2 і 1.3, встановлено, що на сьогоднішній день найбільш ефективними та динамічно впроваджуваними до практичного використання у різних агрокліматичних умовах є технології на основі інформаційно-орієнтованого підходу із залученням алгоритмів штучного інтелекту та машинного навчання, що відносяться до категорії інтелектуалізованих методів. Функціонально-алгоритмічною основою таких моделей у реальній практиці є засоби дистанційного зондування земної поверхні та наземні сенсорні технології збору даних із їх подальшою обробкою на основі залежностей врожайності від характеристик навколишнього середовища та видів вирощуваних с/г культур із залученням інтелектуалізованих методів під час вирішення задач предикативної аналітики даних. Це дозволяє підвищити точність прогнозів завдяки обліку значної кількості інформативних і дестабілізуючих ґрунтокліматичних і біологічних факторів, а також адаптувати прогнози до динаміки впливових показників.

Також варто зазначити, що під час побудови моделей прогнозування врожайності с/г культур, у тому числі й інтелектуалізованих, використовуються відомі емпіричні залежності з урахуванням фізико-хімічних і біологічних

параметрів, які впливають на індикативні показники продуктивності вирощування с/г культур. До основних таких емпіричних залежностей, що набули найбільшої популярності під час побудови відповідних моделей у агротехнічній практиці, відносяться наступні.

Модель FAO AquaCrop на основі рівняння (1.1) [51], яка призначена для оцінки впливу факторів навколишнього середовища на продуктивність вирощування с/г культур, завдяки імітації їх реакції на режими зволоження:

$$Yield = WP \cdot \sum_{i=1}^n \frac{ET_{C_i}}{ET_0}, \quad (1.1)$$

де $Yield$ – врожайність с/г культур; WP – коефіцієнт продуктивності вологи; ET_{C_i} – споживання вологи с/г культурами в i^{th} день; ET_0 – потенційне споживання вологи с/г культурами; n – кількість днів у циклі вирощування.

Модель Growing degree day (GDD), що призначена для оцінки ефективності росту с/г культур на основі акумульованих теплових одиниць. Ця модель може бути використана як у ретроспективному форматі для оцінки поточної стадії росту с/г культур, так і в предикативному форматі задля прогнозування періоду досягнення визначеної стадії росту. Оцінка врожайності виконується за емпіричними формулами (1.2) і (1.3) [52]:

$$Yield = k_1 + k_2 \cdot GDD, \quad (1.2)$$

де $Yield$ – врожайність с/г культур; k_1, k_2 – емпіричні коефіцієнти моделі лінійної регресії врожайності, що залежать від агрокліматичних умов вирощування та видів с/г культур; GDD – акумульовані теплові одиниці.

$$GDD = \sum_{i=1}^n \frac{T_{\max_i} - T_{\min_i}}{2} - T_{base}, \quad (1.3)$$

де $T_{\max_i}$, $T_{\min_i}$ – максимальна та мінімальна температури в $i^{\text{й}}$ день; T_{base} – гранична температура, нижче якої с/г культура не зростає (визначається видом вирощуваних с/г культур); n – кількість днів у циклі вирощування.

Фототермічні моделі, що враховують сукупний вплив температури та сонячної радіації на якість вирощування с/г культур протягом вегетаційного циклу [53, 54]. В таких моделях врожайність оцінюється за емпіричною залежністю (1.4) на основі фототермічного індексу (1.5).

$$Yield = k_1 + k_2 \cdot PTI, \quad (1.4)$$

де $Yield$ – врожайність с/г культур; k_1 , k_2 – емпіричні коефіцієнти моделі лінійної регресії врожайності, що залежать від агрокліматичних умов вирощування та видів с/г культур; PTI – акумульовані теплові одиниці.

$$PTI = \sum_{i=1}^n (T_i - T_{\text{base}}) \cdot SR_i, \quad (1.5)$$

де T_i – середня температура повітря в $i^{\text{й}}$ день; T_{base} – гранична температура, нижче якої с/г культура не зростає (визначається видом вирощуваних с/г культур); SR_i – інтенсивність сонячної радіації в $i^{\text{й}}$ день; n – кількість днів у циклі вирощування.

Математичні моделі оцінки врожайності с/г культур на основі рівнянь водного балансу, що дозволяють визначити продуктивність зростання с/г культур на основі доступної та спожитої вологи з ґрунту за рівнянням виду (1.6) [55, 56].

$$\Delta WBS = I + P - ET - R - D, \quad (1.6)$$

де ΔWBS – зміна показника водного балансу ґрунту; I – кількість вологи, що надходить через полив; P – кількість вологи, що надходить через опади; ET – показник евапотранспірації; R – показник поверхневого стоку ґрунтової вологи;

D – показник інфільтрації води в ґрунті.

У такому випадку прогнозування врожайності може бути реалізовано на підставі емпіричних залежностей між продуктивністю зростання с/г культур та водним балансом із урахуванням типів і стадій їх росту.

Також значного поширення в сучасній сільськогосподарській практиці набули:

- емпіричні моделі на основі показника накопичення фотосинтетично активної радіації (ФАР), що призначені для оцінки продуктивності вирощування с/г культур на основі впливу доступної ФАР [57, 58].

- моделі, що враховують динаміку температури, водного балансу та вмісту азоту в ґрунті під час оцінки продуктивності фотосинтезу с/г культур на різних стадіях вегетації [59–61].

На підставі проведеного аналізу відомих емпіричних залежностей (1.1)–(1.6) щодо впливу ґрунтокліматичних параметрів на продуктивність вирощування с/г культур [51–61] можна зробити висновок, що більшість із них ґрунтується на обліку показників динаміки температури, ґрунтової води та інтенсивності сонячної радіації. Ці моделі є як однофакторними, так і багатфакторними, а отже дозволяють оцінювати врожайність із різним ступенем достовірності прогнозів, що обумовлює необхідність обліку специфічних агрокліматичних умов і видів вирощуваних с/г культур під час вибору конкретної моделі в процесі подальших досліджень.

Таким чином, проведений аналіз і систематизація відомих результатів досліджень щодо розробки і використання підходів і моделей опрацювання даних агромоніторингу доводять необхідність і перспективність проведення подальших досліджень із розробки програмно-алгоритмічного забезпечення трансформації даних агромоніторингу задля прогнозування врожайності с/г культур на основі комплексного підходу із залученням комп'ютеризованих і інтелектуалізованих моделей предикативної аналітики врожайності з урахуванням агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту.

1.3 Аналіз відомих методів і програмно-алгоритмічних засобів інтелектуалізованої трансформації даних агромоніторингу

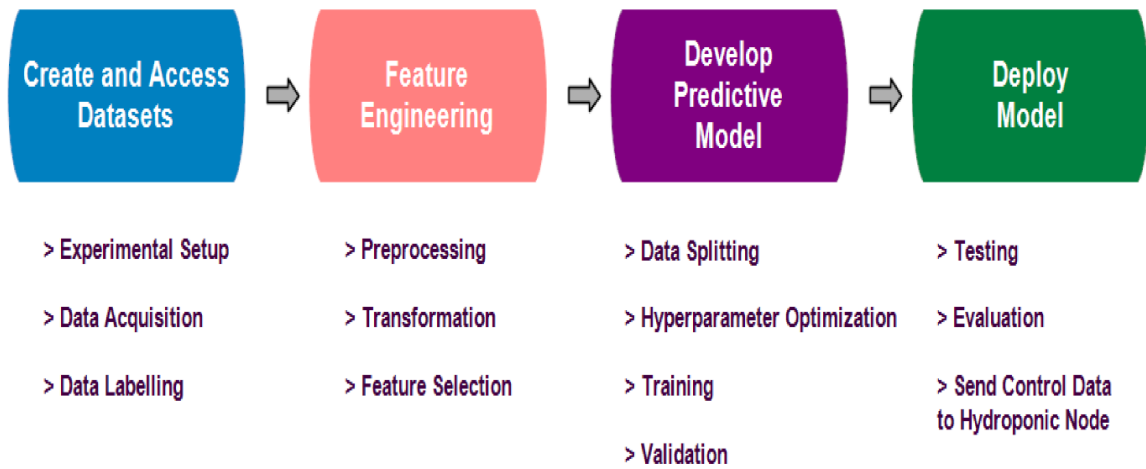
У сьогоденній світовій практиці науково-прикладні засади щодо розробки методів і програмно-алгоритмічних засобів обробки розподілених у часі й просторі масивів даних, у тому числі агромоніторингу, базуються на концептуальних положеннях Інтернету речей, теорії великих даних, точного землеробства, цифрових двійників тощо. В агротехнічній практиці ці методи і засоби мають на меті реалізацію інтеграційного підходу щодо збору, багатоетапної обробки, аналізу й інтерпретації значних обсягів даних і інформації задля підвищення продуктивності с/г процесів.

З урахуванням мінливих агрокліматичних умов та варіативних агротехнічних практик у сьогоденних реаліях аграрних виробництв предикативний моніторинг ґрунтокліматичних характеристик та стану посівів під час повного циклу вирощування відіграє важливу роль у зменшенні ризиків та забезпеченні високих показників обсягів, якості й темпів вирощування с/г культур. На підставі аналізу релевантних наукових джерел [44, 62–66] встановлено, що задля досягнення високих показників адекватності та об'єктивності обробки даних агромоніторингу має бути реалізований комплексний підхід. Він має ґрунтуватися на застосуванні методів і програмно-алгоритмічних засобів інтелектуалізованої трансформації, що охоплюють етапи масштабного збору, попередньої обробки, інтелектуального аналізу та автоматизованої підтримки прийняття рішень, як показано на рисунку 1.9.

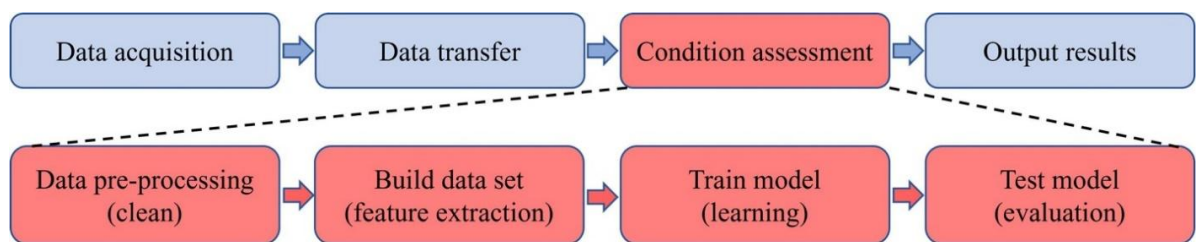
Першим етапом такого комплексного підходу є збір розподілених у часі й просторі вимірювальних даних від різних джерел, як наприклад: наземні сенсорні мережі, комп'ютеризовані метеостанції, аерокосмічні технології, БПЛА, лабораторний аналіз, зовнішні вебресурси та інше.

На наступному етапі має бути обов'язково реалізована попередня обробка масивів даних, що зазвичай включає етапи фільтрації, форматування, усереднення, агрегування за відповідними ознаками, нормалізації,

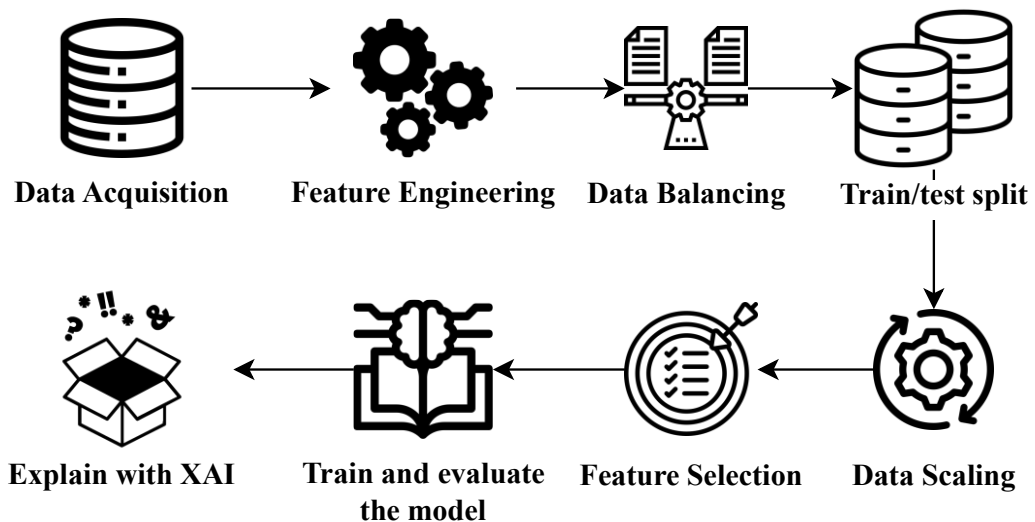
масштабування, заповнення пропусків та інше.



а) блок-схема впровадження засобів інтелектуальної аналітики даних [62]



б) блок-схема підходу до інтелектуальної трансформації даних [63]



в) блок-схема узагальненої послідовності інтелектуальної трансформації даних із застосуванням алгоритмів машинного навчання [44]

Рисунок 1.9 – Графічна інтерпретація прикладів розробки та застосування комплексного підходу до інтелектуальної аналітики даних [44, 62, 63]

Після попередньої обробки виконується етап інтелектуальної аналітики даних на основі методів штучного інтелекту та машинного навчання задля вирішення задач прогнозування, класифікації та / або оптимізації виходячи із цілей аналізу. Завершальним етапом такого комплексного підходу під час трансформації вимірювальних даних є автоматизована інтерпретація результатів та/або моделей, а також підтримка прийняття рішень, що дозволяє підвищити ефективність планування й реалізації агротехнічних заходів.

Таким чином, на підставі проведеного аналізу та систематизації відомих результатів досліджень щодо розробки та практичного використання методів і засобів інтелектуального аналізу вимірювальних даних на основі комплексного підходу було сформульовано узагальнену послідовність основних етапів трансформації результатів агромоніторингу під час прогнозування врожайності с/г культур (див. табл. 1.4), яка може бути використана як базова під час подальших досліджень цієї дисертаційної роботи [24, 25].

Таблиця 1.4 – Результати систематизації основних етапів комплексного підходу до інтелектуальної трансформації даних агромоніторингу

Етап трансформації даних	Використовувані технології та/або базові операції
Збір даних	Наземні сенсорні мережі, супутники, БПЛА, комп'ютеризовані метеостанції, зовнішні вебсервіси, результати лабораторного аналізу
Попередня обробка	Фільтрація, агрегування, заповнення пропусків, форматування, усереднення, нормалізація, стандартизація, балансування
Інтелектуальна аналітика	Факторний і кореляційний аналіз, виявлення інформативних ознак, прогнозування на основі алгоритмів штучного інтелекту та/або машинного навчання

Інтерпретація та пояснення результатів	Ергономічна візуалізація, застосування алгоритмів пояснювального штучного інтелекту (ХАІ)
Автоматизована підтримка прийняття рішень	Автоматизація звітності та генерування сповіщень, інтеграція із зовнішніми інформаційними сервісами, зокрема генеративного штучного інтелекту

З урахуванням сучасного стану розвитку інформаційних та комп'ютерних технологій інтелектуальної трансформації даних найбільшу зацікавленість з точки зору дослідницького характеру являють етапи інтелектуальної аналітики, інтерпретації й пояснення та автоматизованої підтримки прийняття рішень. Це дозволяє програмними засобами реалізувати системний підхід до комплексної аналітики вимірювальних даних, що масштабю характеризують технологічні процеси с/г виробництв та, як наслідок, оптимізувати агротехнічні процедури в умовах значної динаміки та обсягів інформації.

Як зазначено в таблиці 1.4, одним із основних етапів інтелектуальної аналітики даних є відбір інформативних ознак задля покращення загальної продуктивності моделей машинного навчання під час обробки просторово-часових результатів агромоніторингу. Відомі методи відбору інформативних ознак поділяються на три основні групи: вбудовані, фільтраційні та обгорткові. Ці групи методів, своєю чергою, можуть бути реалізовані з використанням значної кількості алгоритмів [67, 68], як у деталізованому вигляді показано на рисунку 1.10.

Кожна із зазначених груп методів і відповідних алгоритмів відбору інформативних ознак характеризується своїми перевагами та обмеженнями, що обумовлені складністю та ступенем автоматизації обчислювальних процедур, обсягами вхідних масивів даних, адаптацією та масштабуванням до динаміки вхідних параметрів та інше. Таким чином, виникає необхідність у врахуванні цього аспекту під час реалізації програмно-алгоритмічних засобів комплексної трансформації даних агромоніторингу.

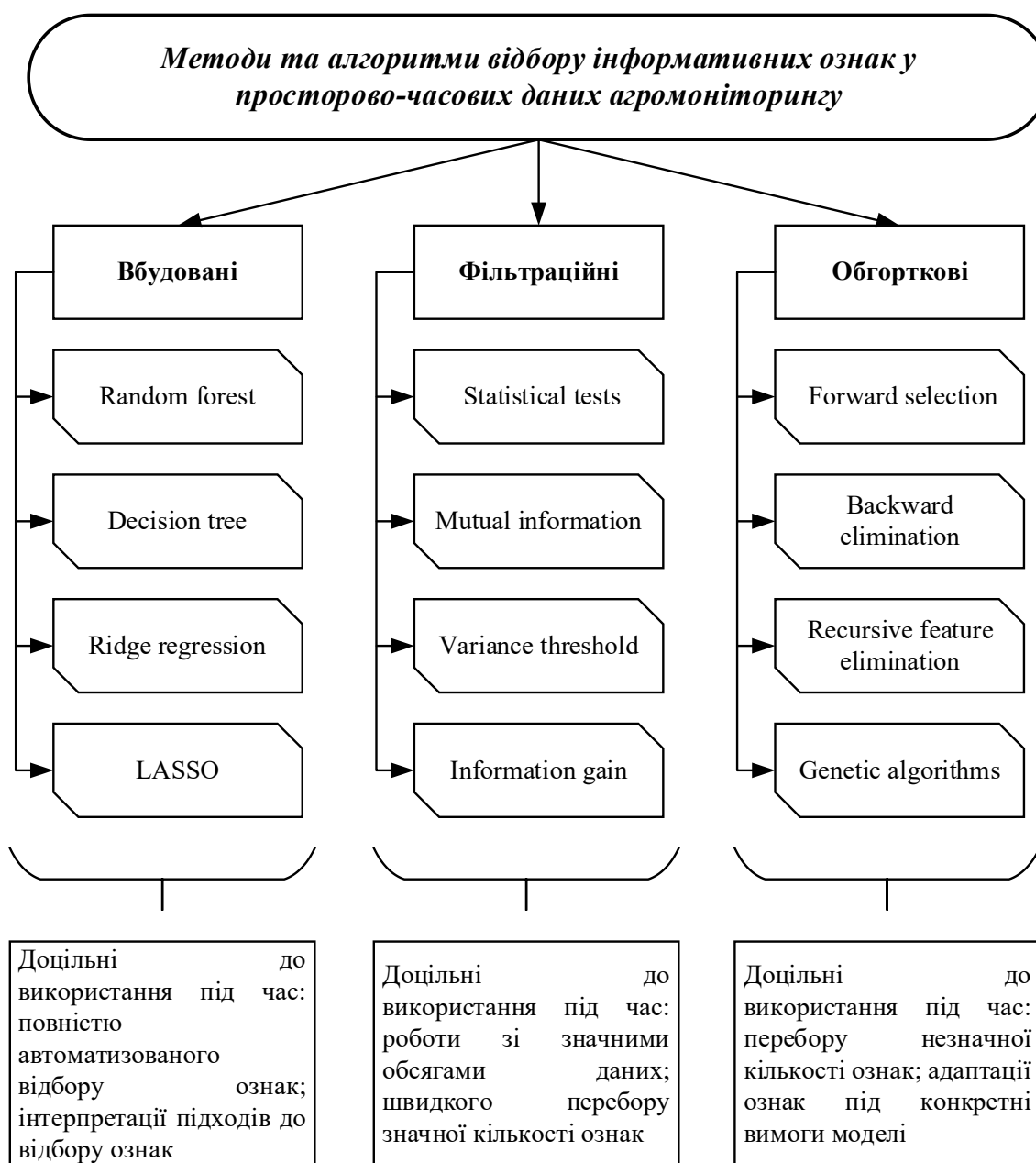


Рисунок 1.10 – Класифікація методів і алгоритмів відбору інформативних ознак, що потенційно можуть бути застосовані під час обробки даних агромоніторингу

Наступним основним етапом інтелектуальної аналітики даних є застосування алгоритмів машинного навчання під час прогнозування врожайності с/г культур на основі розподілених у часі й просторі масивів значень щодо динаміки ґрунтокліматичних і агротехнологічних параметрів. Відомі на сьогоднішній день алгоритми машинного навчання є доволі розвиненими та можуть бути класифіковані за критерієм наявності міток у навчальних даних [44, 69, 70], як показано на рисунку 1.11.

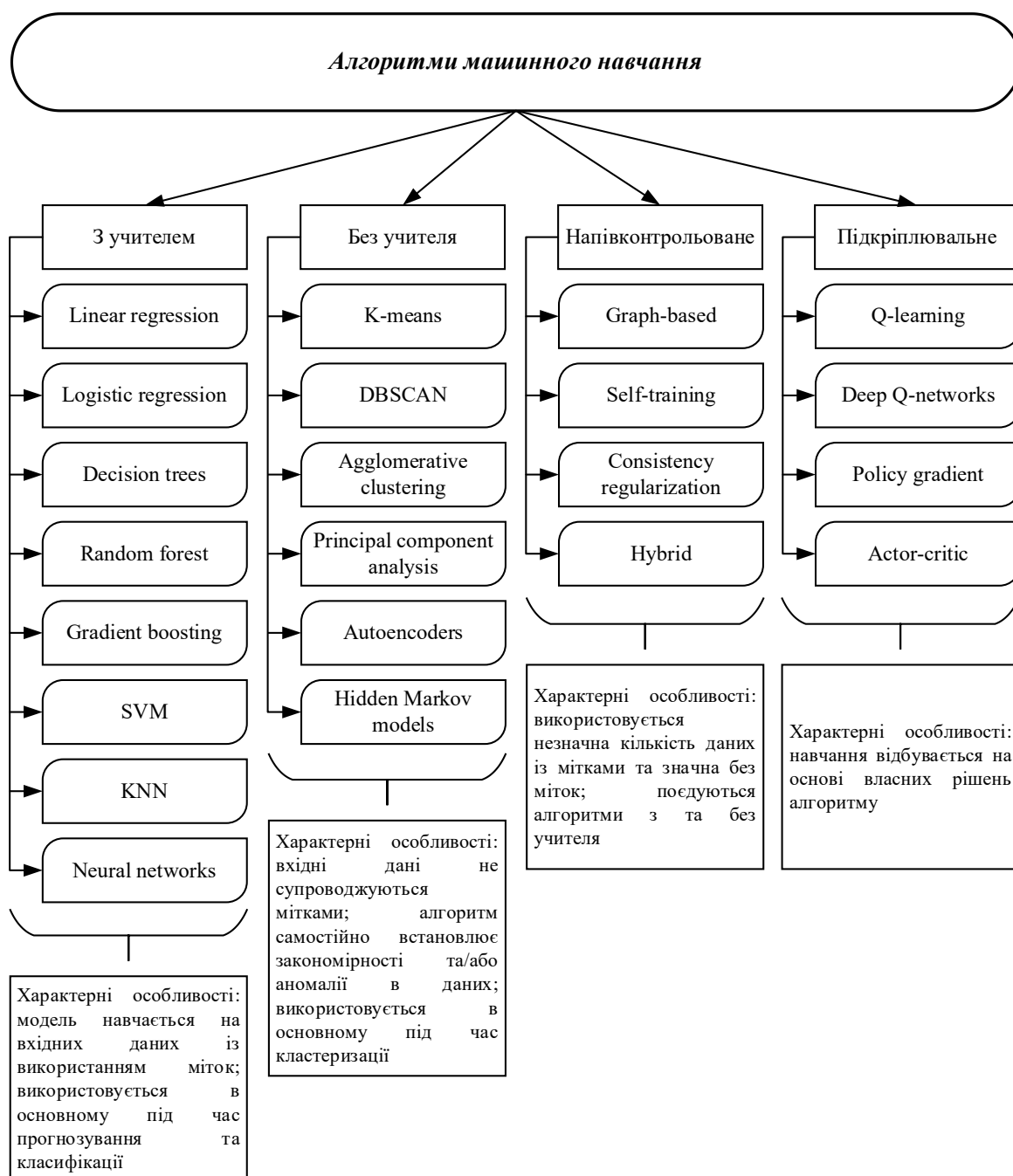


Рисунок 1.11 – Класифікація алгоритмів машинного навчання, що потенційно можуть бути застосовані під час аналітики даних агромоніторингу

Таким чином, на підставі аналізу результатів класифікації відомих алгоритмів машинного навчання, які у систематизованому вигляді наведено на рисунку 1.11, виникає необхідність у проведенні досліджень щодо вибору найбільш раціонального алгоритму машинного навчання під час прогнозування врожайності с/г культур з урахуванням ґрунтокліматичних умов, що спостерігають у процесі вирощування агрокультур на відкритому ґрунті.

Варто зазначити, що моделі прогнозування цільової функції (врожайність с/г культур) із використанням алгоритмів машинного навчання на основі трансформації ґрунтокліматичних даних потребують автоматизованої інтерпретації та пояснення задля підвищення об'єктивності підтримки прийняття рішень. Для цього можуть бути використані достатньо розвинуті на теперішній час алгоритми пояснювального штучного інтелекту (XAI) [71–73], що класифікуються за різними критеріями, зокрема, як в узагальненому вигляді показано на рисунку 1.12.

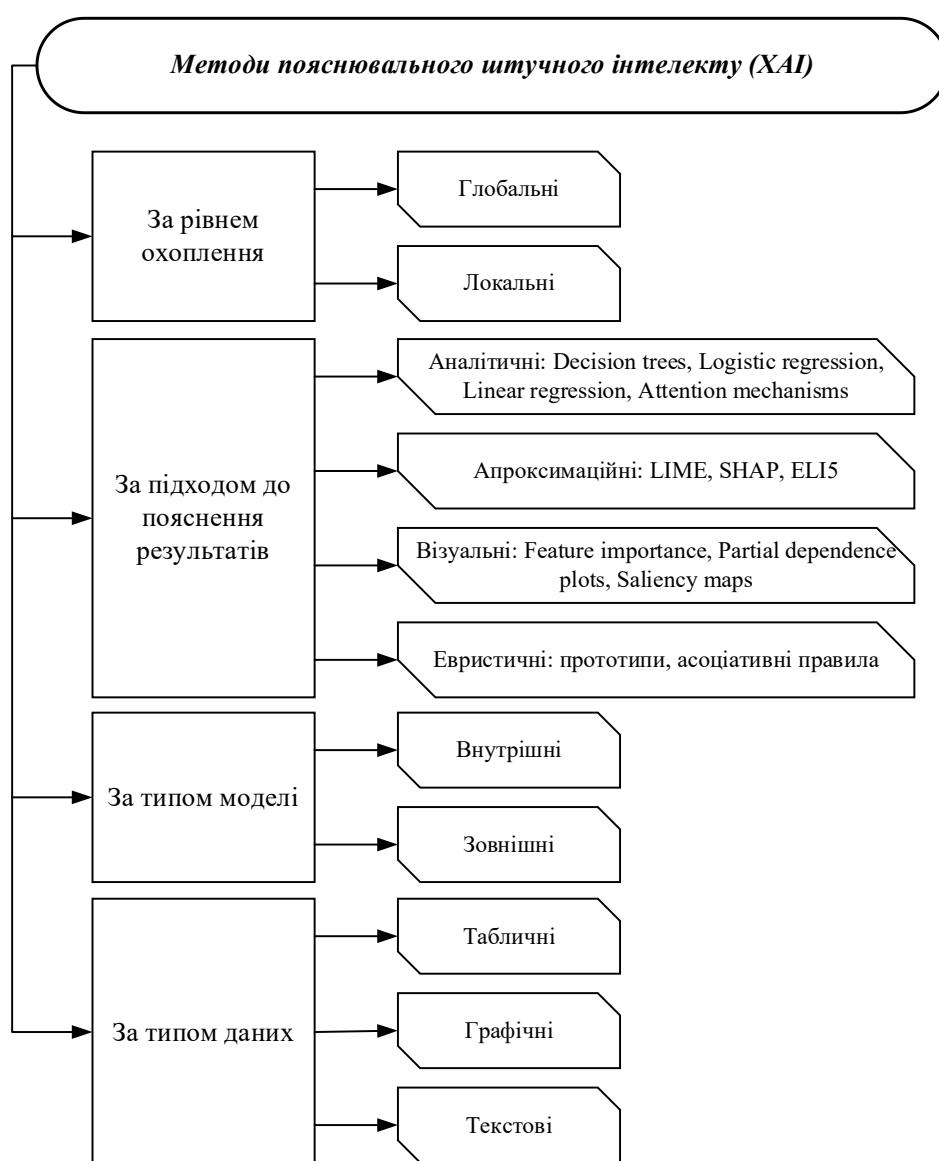


Рисунок 1.12 – Класифікація відомих методів пояснювального штучного інтелекту (XAI)

З урахуванням специфіки провадження рослинництва відкритого ґрунту під час пояснення моделей прогнозування врожайності с/г культур на основі інтелектуального аналізу результатів моніторингу ґрунтокліматичних параметрів найбільш доцільними до використання є методи, що враховують темпоральні залежності між вхідними даними (результати агромоніторингу) та цільовою функцією (врожайність с/г культур) [44]. До таких методів апріорно відносяться апроксимаційні методи SHAP, ELI5, LIME та евристичний Anchor, які за рівнем охоплення можуть бути використані як на локальному рівні, так і на глобальному.

Завершальним етапом реалізації комплексного підходу до інтелектуальної трансформації даних агромоніторингу виходячи з актуального стану та трендів розвитку інформаційних технологій агротехнічного призначення є автоматизована підтримка прийняття рішень, яка дозволяє оперативно спланувати та ефективно реалізувати агротехнічні процедури із забезпечення бажаних показників якості й обсягів виробництва с/г продукції рослинництва. Підходи до забезпечення автоматизованої підтримки прийняття рішень є доволі розвиненими та можуть бути класифіковані за способом обробки інформації (на основі правил, на основі аналізу даних, оптимізаційні, предикативні та інтерактивні), за типом задіяних алгоритмів (експертні, на основі алгоритмів машинного навчання, гібридні), за ступенем автоматизації (автоматизовані, автоматичні) та інше [74–76]. Таким чином, наявність широкого спектру методів і засобів щодо реалізації засад автоматизованої підтримки прийняття рішень обумовлює необхідність проведення подальших досліджень із розробки програмно-алгоритмічних засобів їх інтеграції до складу інформаційних технологій, що реалізують комплексний підхід трансформації агрокліматичних даних під час прогнозування врожайності с/г культур.

Варто зазначити, що на сьогоднішній день відома значна кількість результатів досліджень і розробок в галузі інформаційно-орієнтованої трансформації сільського господарства на основі створення й впровадження інтелектуалізованих методів і програмно-алгоритмічних засобів комплексної інтелектуальної аналітики даних агромоніторингу. Отже, актуальним завданням є

аналіз і логічне узагальнення відомих результатів досліджень зі створення й використання таких методів і засобів з урахуванням отриманого прикладного ефекту, як наведено в таблиці 1.5.

Таблиця 1.5 – Результати аналізу і логічного узагальнення відомих результатів досліджень і розробок щодо інтелектуальної трансформації даних агромоніторингу під час прогнозування врожайності с/г культур

Предмет дослідження та/або розробки	Наукова новизна та/або практична цінність	Посилання
Наукові дослідження		
Методи та моделі класифікації с/г культур та прогнозування їх врожайності за допомогою алгоритмів машинного навчання	Розроблено та досліджено рекомендаційні комп'ютерні моделі щодо оптимізації посівів на основі алгоритмів машинного навчання (Extra Tree Classifier, Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbour, Gaussian Naive Bayes, Support Vector Machine). Запропонований підхід є адаптивним і масштабованим до нових наборів кліматичних даних і географічних локацій. До моделей інтегровано ХАІ алгоритми, які пояснюють вплив температури, вологості й рівня поживних речовин у ґрунті та кількості опадів на ефективність прогнозування	[77]
Систематизація відомих результатів досліджень із застосування алгоритмів машинного навчання, засобів Інтернету речей і дистанційного зондування під час прогнозування врожайності с/г культур	Надано обґрунтування підходів до розробки веб-та/або мобільних застосунків прогнозування врожайності шляхом інтеграції технологій збору (IoT, супутники) та обробки (машинне навчання) даних. На підставі результатів критичного аналізу сучасного стану інтеграції цифрових і інформаційних технологій обґрунтовано узагальнену архітектуру та сформульовано перспективи подальшого розвитку комп'ютерних засобів і технологій прогнозування врожайності с/г культур	[78]
Критичний аналіз та узагальнення методів та моделей прогнозування врожайності картоплі	Представлено результати огляду засад розвитку та застосування методів і моделей на основі машинного та глибокого навчання для прогнозування врожайності картоплі. Доведено важливість інтеграції різних джерел даних, включаючи метеорологічні й фенотипічні та дані дистанційного зондування під час створення комбінованих, геостатистичних та байєсівських інформаційних моделей	[79]
Підходи до застосування алгоритмів машинного навчання під час довготривалого прогнозування врожайності с/г культур	Доведено ефективність застосування різних видів алгоритмів машинного навчання (Linear Regression, Random Forest, Neural Networks) під час прогнозування врожайності соняшника в агрокліматичних умовах Іспанії на основі статистичних даних із 2001 по 2020 рр. Досліджено вплив гіперпараметрів застосовуваних моделей на ефективність прогнозування врожайності	[80]

Продовження табл. 1.5

Підходи до підвищення точності прогнозування врожайності с/г культур шляхом застосування моделей машинного та глибинного навчання	Наведено результати комплексних досліджень щодо ефективності застосування різних моделей на основі алгоритмів машинного та глибинного навчання (Logistic Regression, Multilayer Perceptron, Sequential Minimal Optimization (SMO), J48, Random Forest, REP Tree та авторська модель глибинного навчання) під час прогнозування врожайності с/г культур шляхом аналітики інформативних ґрунтокліматичних параметрів (вміст азоту, фосфору й калію в ґрунті, вологості, кислотності й температури ґрунту та кількості опадів) за критерієм точності класифікації с/г об'єктів рослинництва	[81]
Оцінка продуктивності моделей на основі алгоритмів машинного навчання під час прогнозування врожайності с/г культур	Доведено ефективність застосування інформаційних технологій прогнозування врожайності широкого спектру с/г культур (рис, сорго, бавовна, цукрова тростина) на основі алгоритмів машинного навчання, які навчені з урахуванням даних про погодні умови, характеристики ґрунту та передісторію врожайності	[82]
Критичний аналіз і узагальнення технологій прогнозування врожайності зернових культур на основі даних дистанційного моніторингу з БПЛА та алгоритмів машинного навчання	Представлено результати критичного аналізу та узагальнення відомих результатів досліджень, що охоплюють характеристики створення інформаційних технологій прогнозування врожайності с/г культур із урахуванням аспектів впливу типів культур, збору даних, вибору інформативних ознак, адаптації моделей до періодів вегетації. Встановлено, що етап вибору інформативних ознак є одним із найбільш впливових на підсумкову точність моделі, а також те, що в більшості випадків найкраще себе зарекомендували алгоритми типу випадковий ліс та згорток нейронна мережа під час обробки даних з БПЛА щодо прогнозування врожайності с/г культур	[83]
Метод прогнозування врожайності с/г культур на основі алгоритмів мультифакторного аналізу та нейронних мереж	Запропоновано підхід до оптимізації процесу переджнивного підсушування с/г культур задля прискорення досягання й полегшення збирання врожаю соняшнику на основі моделі оцінки обсягів потенційних втрат врожаю. Такий підхід дозволив виконати аналіз впливу локальних факторів і параметрів на показники розвитку й врожайності с/г культур. Розроблений метод робить можливою оптимізацію щодо диференційованого внесення добрив під час вирощування с/г культур	[84]
Комплексний аналіз технологій підтримки прийняття рішень с/г призначення на основі великих мовних моделей (LLM), доповнених генерацією з розширеним пошуком (RAG)	Проаналізовано світовий досвід використання LLM під час автоматизованої підтримки прийняття рішень у агротехнічній практиці. Доведено необхідність інтеграції засобів RAG задля вирішення проблеми обмеженості відомих баз знань. Сформульовано пріоритетні напрямки досліджень, які полягають у підвищенні достовірності та пояснювальності під час інтеграції LLM та RAG моделей до систем підтримки прийняття рішень	[76]

Продовження табл. 1.5

Методика попередньої обробки даних агромоніторингу задля підвищення точності детектування хвороб с/г культур в агрокліматичних умовах України	Розроблено та доведено ефективність комп'ютеризованої методики підвищення точності прогнозування хвороб зернових культур на основі комплексних досліджень, які включали: попередню апроксимацію вхідних масивів даних, ідентифікацію параметрів моделей предикативної аналітики ймовірності виникнення конкретних видів хвороб зернових культур у агрокліматичних умовах України; формалізований облік кліматичних параметрів та експертного досвіду в галузі стресостійкості с/г культур	[46]
Інформаційний аналіз і комплексна систематизація методів прогнозування врожайності за допомогою алгоритмів машинного навчання	Представлено результати аналізу та систематизації досліджень, які націлено на встановлення атрибутів моделей на основі алгоритмів машинного навчання під час прогнозування врожайності с/г культур за період із 2017 по 2024 рр. Встановлено, що найбільшого поширення серед алгоритмів машинного навчання набули Linear Regression, Random Forest та Gradient Boosting Trees, а серед алгоритмів глибокого навчання – Convolutional Neural Network та Long Short-Term Memory. Ефективність моделей у більшості випадків оцінюється на основі метрик: RMSE, R^2 та MAE	[85]
Прикладні розробки		
Agricultural Production Systems Simulator (APSIM)	Призначено для довгострокового прогнозування врожайності с/г культур. Реалізовано засоби моделювання режимів зростання с/г культур із урахуванням стану ґрунту та клімату. Наявні версії з різним функціоналом у відповідності до ліцензії (комерційне і некомерційне використання)	[86]
CropSyst	Призначено для моделювання потреб у забезпеченні вологою та азотом під час вирощування с/г культур із урахуванням ґрунтокліматичних факторів. Варіативними параметрами моделі є: тип с/г культур, сівозмінна, режими зрошення й внесення азотних добрив та інше. Є розробкою, яка поширюється всесвітньо визнаною організацією FAO	[87]
Climate FieldView	Моделі прогнозування врожайності с/г культур базуються на даних від супутників і наземних IoT-технологій. Основним функціональним призначенням є аналіз ефективності агрозаходів та локалізація зон продуктивності с/г полів. Розробка доступна у форматах мобільного та вебзастосунку	[88]
EOSDA Crop Monitoring	Професійна онлайн платформа, яка зокрема призначена для прогнозування врожайності с/г культур на основі супутникових знімків та метеоданих із залученням алгоритмів машинного навчання	[89]
Granular	Являє собою комп'ютеризовану аналітичну технологію прогнозування врожайності с/г культур і керування полями на основі даних про погоду, ґрунт, типи агрокультур, сівозміну тощо	[90]

Доцільно відзначити, що в таблиці 1.5 наведено невичерпний перелік сучасних науково-прикладних досягнень у галузі інформаційно-орієнтованого прогнозування врожайності с/г культур. У цій таблиці наведено типові результати, які доводять ефективність і перспективність застосування комплексного підходу на основі багатоетапної предикативної аналітики агрокліматичних даних із залученням алгоритмів машинного навчання та штучного інтелекту. Таким чином, можна констатувати факт того, що під час розробки програмно-алгоритмічного забезпечення засобів повноциклової трансформації даних агромоніторингу доцільним є врахування наступного: функціональний рівень збору агрокліматичних даних має реалізовуватись із залученням технологій Інтернету речей, комп'ютеризованих метеостанцій, дистанційного зондування та зовнішніх вебресурсів; функціональний рівень обробки даних має враховувати засади багатоетапності з точки зору попередньої підготовки даних, виявлення інформативних ознак та предикативної аналітики на основі алгоритмів машинного навчання; етап інтерпретації та пояснення предикативних моделей має базуватись на алгоритмах пояснювального штучного інтелекту; обов'язковим етапом має бути автоматизована підтримка прийняття рішень із залученням засобів оперативного сповіщення користувачів на основі обробки запитів та/або подій у режимі реального часу.

З огляду на вищезазначене, встановлено, що інформаційно-орієнтована модернізація сільського господарства є актуальним глобальним трендом сталого розвитку цієї галузі. Саме тому важливим завданням є вдосконалення та подальший розвиток методів і програмно-алгоритмічних засобів інтелектуальної трансформації даних агромоніторингу, які реалізують комплексний підхід. Адже, це дозволить підвищити адаптивність програмних рішень прогнозування врожайності с/г культур та інформаційних технологій на їх основі до значної динаміки агрокліматичних умов завдяки покращенню показників точності прогнозів та інтерпретованості предикативних моделей із можливістю автоматизованої підтримки прийняття рішень щодо планування та реалізації агротехнічних процедур під час всіх етапів вирощування [24].

1.4 Обґрунтування мети та завдань досліджень дисертаційної роботи

Результати критичного аналізу та логічної систематизації відомих наукових і аналітичних досліджень та практичних розробок щодо сучасного стану програмно-алгоритмічних рішень із цифровізації та інтелектуалізації рослинництва відкритого ґрунту дозволили довести необхідність подальшого розвитку й удосконалення методів і засобів побудови програмно-алгоритмічного забезпечення інформаційних і комп'ютерних технологій прогнозування врожайності с/г культур. Це обумовлено такими обмеженнями з точки зору науково-прикладної пропрацьованості наступних питань:

- недостатність науково обґрунтованих результатів щодо синтезу програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації гетерогенних даних агромоніторингу, що агрегуються від різних джерел, із урахуванням повноциклової реалізації етапів попередньої обробки, інтелектуальної аналітики, інтерпретації й пояснення прогнозних моделей, а також автоматизованої підтримки прийняття рішень;

- обмеженість результатів науково-прикладних досліджень щодо ефективності розподілу та програмної інтеграції обчислювальних і аналітичних процедур між ієрархічними рівнями інформаційних технологій на основі децентралізованої архітектури із використанням туманних обчислень;

- обмежена адаптація відомих моделей, які побудовані на історичних даних та потребують уточнення з обліком значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту в конкретних агротехнологічних умовах.

Отже, важливою й актуальною є науково-практична задача, яка потребує проведення теоретико-прикладних досліджень та полягає в подальшому розвитку інформаційних і комп'ютерних технологій агротехнічного призначення завдяки розробці та використанню програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури під час прогнозування врожайності сільськогосподарських культур із

урахуванням значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту в конкретних агротехнологічних умовах. Під час розробки та дослідження зазначеного програмно-алгоритмічного забезпечення агротехнічного призначення мають бути враховані наступні сучасні світові засади й тренди розвитку прикладних інформаційних і комп'ютерних технологій, а саме: реалізація повного циклу багатоетапної обробки вимірювальних даних в межах децентралізованої програмно-алгоритмічної архітектури із залученням алгоритмів штучного інтелекту та машинного навчання; об'єктивний розподіл агрегаційних, обчислювальних і аналітичних процедур між функціональними рівнями технології трансформації даних агромоніторингу; достовірна інтерпретація моделей прогнозування із залученням методів пояснювального штучного інтелекту; автоматизована підтримка прийняття рішень та сповіщення користувачів.

Основна мета дисертаційної роботи полягає в удосконаленні інформаційно-орієнтованого підходу до прогнозування врожайності сільськогосподарських культур завдяки розробці та використанню програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу.

Для досягнення поставленої мети необхідно вирішити такі дослідницькі завдання:

- проведення комплексного аналізу й логічного узагальнення відомих науково-прикладних розробок, які характеризують сучасний стан програмно-алгоритмічних рішень із цифровізації та інтелектуалізації рослинництва відкритого ґрунту, із формулюванням пріоритетів досліджень цієї дисертації;

- обґрунтування структурно-функціональної організації та інформаційної моделі досліджуваного програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур на основі туманної архітектури;

- розробка та дослідження ефективності програмно-алгоритмічних засобів

комплексної трансформації даних агромоніторингу, які реалізують етапи попередньої обробки, інтелектуальної аналітики, а також інтерпретації та пояснення предикативних моделей із можливістю автоматизованої підтримки прийняття агротехнічних рішень;

– створення та валідація комп'ютерної технології, яка в межах єдиної архітектури реалізує комплексну трансформацію результатів агромоніторингу на основі розробленого програмно-алгоритмічного забезпечення прогнозування врожайності сільськогосподарських культур;

– проведення критичного аналізу отриманих результатів щодо розробки та дослідження програмно-алгоритмічного забезпечення агротехнічного призначення з прогнозування врожайності сільськогосподарських культур, а також обґрунтування подальших напрямків перспективних досліджень із розвитку та практичного використання розробки.

На підставі зазначеного вище сформульовано об'єкт і предмет дослідження дисертації.

Об'єкт дослідження – інформаційні процеси програмної трансформації даних агромоніторингу, що протікають під час прогнозування врожайності сільськогосподарських культур.

Предмет дослідження – методи, засоби й моделі створення та валідації програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур.

1.5 Висновки за першим розділом

У цьому розділі дисертації вирішено одне з основних дослідних завдань, що полягало в комплексному аналізі й логічному узагальненні відомих науково-прикладних розробок, які характеризують сучасний стан програмно-алгоритмічних рішень із цифровізації та інтелектуалізації рослинництва відкритого ґрунту. В результаті проведених досліджень встановлено наступне:

1. Інформаційно-орієнтована трансформація сільськогосподарської галузі, що може бути реалізована завдяки розробці й упровадженню цифрових і інформаційних технологій агротехнічного призначення, як на світовому рівні, так і на національному, є актуальним напрямком сталого розвитку, що потребує додаткових досліджень, адже під час збільшення оброблюваних площ у показниках абсолютної та питомої врожайності в глобальних масштабах не спостерігається сталої позитивної динаміки, що свідчить про значний вплив негативної динаміки кліматичних умов, доступності водних ресурсів, наявності шкідників і прогресування хвороб, недостатньої ефективності планування й реалізації агротехнічних заходів під час забезпечення високих показників обсягів і якості виробництва с/г продукції.

2. На теперішній час підходи й моделі прогнозування врожайності с/г культур є значно розвиненими та враховують різні фізико-хімічні, кліматичні й біологічні аспекти агротехнічних процесів, а також ґрунтуються на широкому спектрі задіяних засобів збору даних і обчислювальних алгоритмах їх трансформації. Обґрунтовано доцільність і перспективність досліджень щодо розробки програмно-алгоритмічного забезпечення повноциклової трансформації даних агромоніторингу під час прогнозування врожайності с/г культур із залученням алгоритмів штучного інтелекту щодо оцінки впливу ґрунтокліматичних параметрів на продуктивність вирощування с/г культур та системної організації інформаційних технологій під час предикативної аналітики врожайності з урахуванням агрокліматичних умов вирощування, а також типів вирощуваних агрокультур та застосовуваних практик їх захисту.

3. На підставі класифікаційного аналізу релевантних комп'ютерно-орієнтованих методів і засобів інтелектуалізованої трансформації даних агромоніторингу доведено, що під час розробки програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур доцільним є врахування наступного: рівень збору має забезпечувати надходження просторово-часових даних, що характеризують різні агрокліматичні умови вирощування; рівень обробки даних має реалізовувати засади багатоетапності щодо попередньої

підготовки даних, визначення інформативних ознак і предикативної аналітики на основі алгоритмів машинного навчання; етап інтерпретації й пояснення предикативних моделей має ґрунтуватись на алгоритмах пояснювального штучного інтелекту; завершальним етапом трансформації даних має бути автоматизована підтримка прийняття рішень із залученням засобів оперативного сповіщення користувачів.

4. Локалізовано основні обмеження відомих результатів наукових досліджень і практичних розробок у галузі програмно-алгоритмічного забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу, що дозволило обґрунтувати науково-практичну задачу дисертаційного дослідження, яка полягає в подальшому розвитку інформаційних і комп'ютерних технологій агротехнічного призначення завдяки розробці та використанню програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури під час прогнозування врожайності сільськогосподарських культур із урахуванням значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту в конкретних агротехнологічних умовах.

5. Сформульовано основну мету, дослідницькі завдання, об'єкт та предмет дослідження цієї дисертації, що дозволить провести комплексні дослідження зі створення й валідації програмно-алгоритмічного рішення інтелектуалізованої трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур із урахуванням специфіки ведення аграрного виробництва в умовах високої варіативності агрокліматичних чинників і різноманіття вирощуваних культур, а також особливостей техніко-технологічних підходів до захисту агрокультур у конкретних агротехнологічних середовищах, задля підвищення ефективності технологічних та виробничих процесів аграрних підприємств, що здійснюють вирощування сільськогосподарських культур на відкритому ґрунті.

РОЗДІЛ 2

ФОРМАЛІЗАЦІЯ ПРОЦЕСУ КОМПЛЕКСНОЇ ІНТЕЛЕКТУАЛІЗОВАНОЇ ТРАНСФОРМАЦІЇ ЧАСОВО-ПРОСТОРОВИХ ДАНИХ АГРОМОНІТОРИНГУ

2.1 Компаративне обґрунтування туманної архітектури досліджуваного програмно-алгоритмічного забезпечення

Науково-практична задача трансформації результатів інструментальних спостережень агромоніторингу, зокрема під час прогнозування врожайності с/г культур, передбачає необхідність агрегування та опрацювання значних обсягів даних щодо інформативних і дестабілізуючих параметрів, які надходять від різнорідних джерел і характеризуються значною часовою й просторовою варіативністю. Таким чином, у загальному випадку розв'язання цієї задачі являє собою побудову інформаційно-орієнтованої моделі, в якій вхідні дані є часо- й просторово розподіленими, неструктурованими та зашумленими [91, 92].

У теперішній науково-технічній практиці під час побудови інформаційних технологій опрацювання даних, що надходять від просторово розподілених пристроїв, найбільшого розвитку та застосування набули дві основні обчислювальні архітектури: хмароцентрична та децентралізована (туманна) [93, 94]. Варто акцентувати увагу на тому, що ці архітектури мають принципові відмінності, саме, в засадах обробки даних, що обумовлені розподілом обчислювально-аналітичних процедур за ієрархічними рівнями комплексних сервісно-орієнтованих систем, при цьому мережева організація апаратних компонент і пристроїв, що входять до їх складу залишається в більшості випадків незмінною, та відповідає еталонній моделі систем і мереж Інтернету речей, що містить такі рівні: фізичні пристрої, мережевий зв'язок, локальні обчислення, накопичення даних, абстракція даних, застосунки, засоби взаємодії з користувацькими пристроями та керування процесами [95, 96].

Під час обробки значних обсягів часово-просторових вимірювальних даних хмароцентричний підхід спроможний забезпечити високі показники масштабності й адаптивності обчислень. Проте цей підхід характеризується підвищеними часовими затримками через формування черг під час централізованої обробки та потребує наявності стабільного високошвидкісного Інтернет-з'єднання. На противагу цьому туманна (децентралізована) обчислювальна архітектура характеризується низьким рівнем затримок і підвищеним рівнем обчислювальної продуктивності, що дозволяє агрегувати та обробляти потоки даних на рівні локальних пристроїв, зменшуючи черги та обчислювальне навантаження на ядро мережі (сервер). Результати детальної порівняльної характеристики обчислювальних архітектур, що зазначені вище, які одержано шляхом аналізу наукових досліджень [93, 94, 97–100], наведено в таблиці 2.1.

Таблиця 2.1 – Результати критеріального порівняльного аналізу хмароцентричної та туманної обчислювальних архітектур

Порівняльний критерій	Хмароцентрична архітектура	Туманна архітектура
Навантаження на інфокомунікаційну мережу	Максимальне (всі неструктуровані й необроблені дані передаються на серверний рівень мережі)	Мінімальне (на ядро мережі передається оброблена інформація)
Надійність	Обумовлена стабільністю Інтернет-з'єднання та доступністю сервера	Мережа підтримує автономний режим роботи
Часова затримка під час обробки даних	Висока (залежить від обсягів даних, швидкості Інтернет-зв'язку, пропускної здатності мережі)	Низька (більша кількість обчислювальних процедур реалізується на локальному рівні)

Продовження табл. 2.1

Оперативність зворотного зв'язку та підтримки прийняття рішень	Наявні часові затримки	Можлива реакція на зміни й події в реальному часі
Масштабованість	Висока	Помірно висока через потребу в розгортанні додаткових пристроїв туманної інфраструктури
Кібербезпека даних	Обмежена (всі дані передаються через мережеву інфраструктуру)	Висока (через мережеву інфраструктуру передаються лише результати обробки даних)
Облік супровідного часово-просторового контенту	Існує потреба у реалізації функціоналу прив'язки обчислювальних процедур до геолокацій одержання вимірювальних даних	Підтримується завдяки обробці даних на локальному рівні
Апаратно-програмна архітектура управління процесами	Централізована	Децентралізована

Таким чином, результати компаративного критеріального аналізу, які наведено в таблиці 2.1, дозволяють встановити наступне: хмароцентрична архітектура в більшості практичних застосунків орієнтована на централізовану обробку значних обсягів даних із певними часовими затримками, а розподілена (туманна) архітектура є більш доцільною до застосування під час агрегування та обробки вимірювальних даних інформаційно-орієнтованого моніторингу з підтримкою прийняття рішень в умовах, що є чутливими до факторів

оперативності й надійності. Враховуючи характерні особливості прикладної сфери застосування розроблюваного програмно-алгоритмічного забезпечення, що націлене на обробку часово-просторових показників результатів агромоніторингу, де в якості джерел даних у більшості випадків виступають бездротові сенсорні мережі, що розташовані в місцевості з нестабільним Інтернет-покриттям, то найбільш доцільною є туманна архітектура [101, 102]. Узагальнена структурно-функціональна організація такої обчислювальної архітектури наведена на рисунку 2.1, а основні техніко-функціональні переваги, які вона спроможна забезпечити під час прикладного застосування в процесі агромоніторингу, у графічному вигляді зазначено на рисунку 2.2.

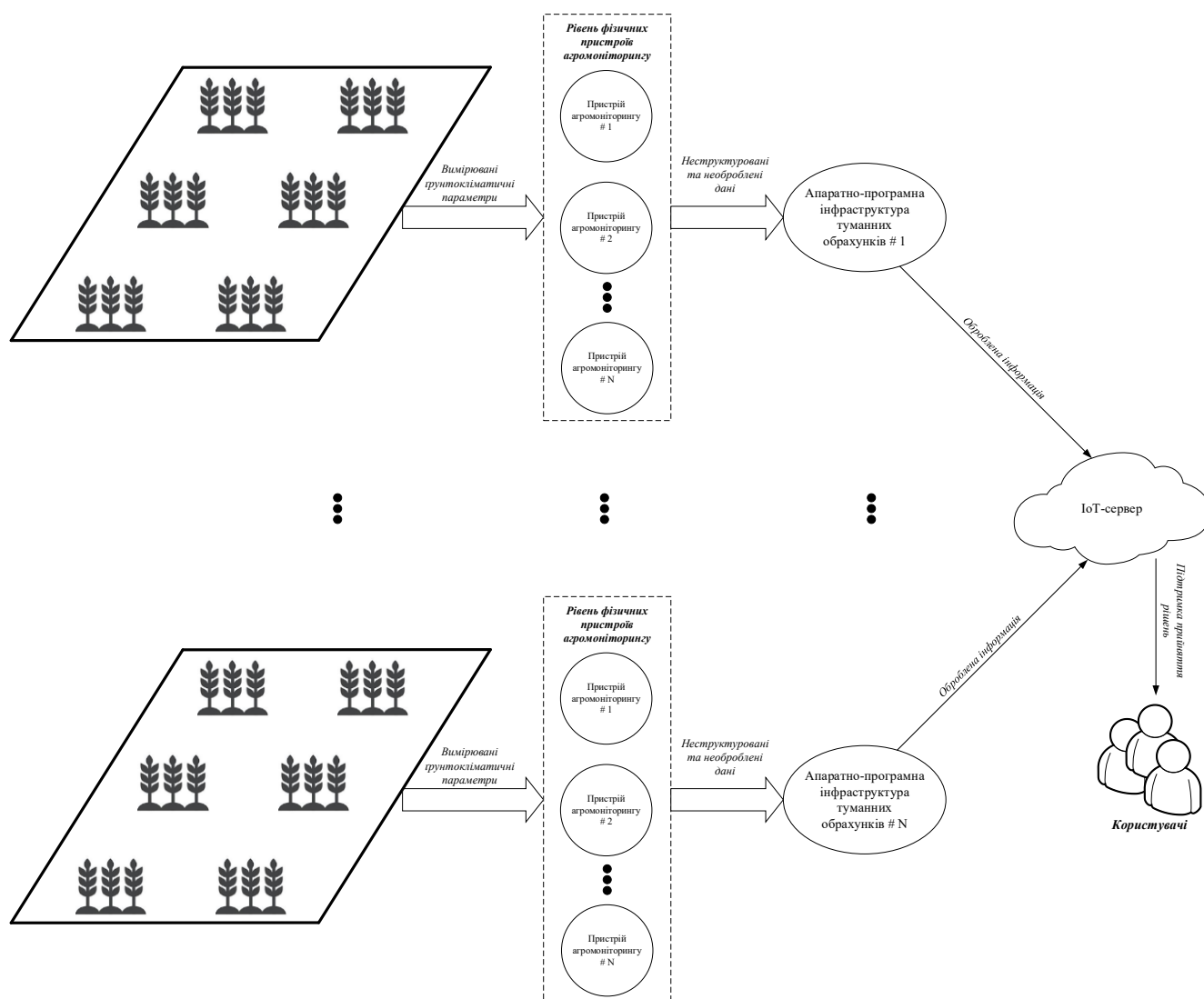


Рисунок 2.1 – Узагальнена структурно-функціональна організація туманної обчислювальної архітектури під час агромоніторингу

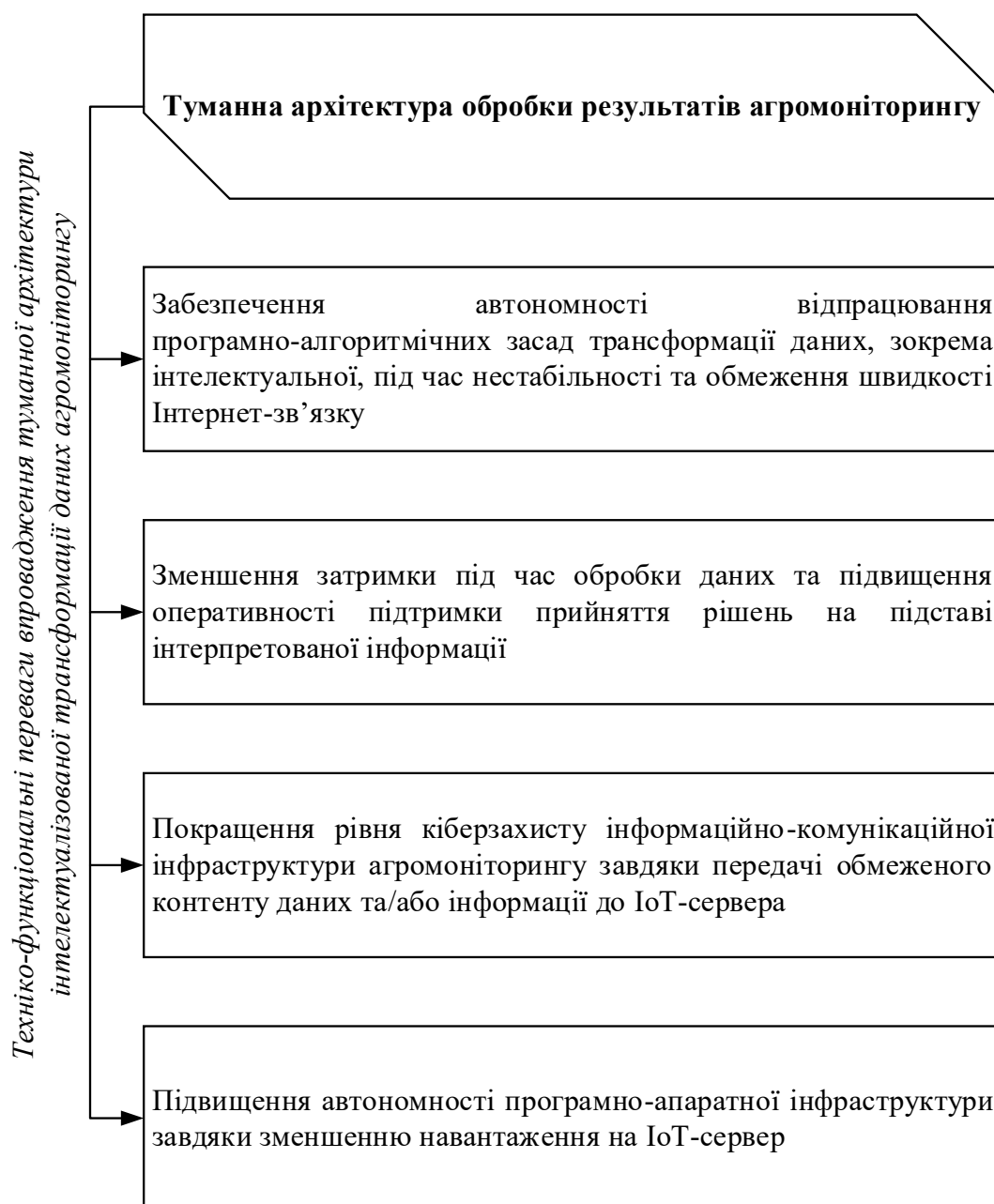


Рисунок 2.2 – Графічна інтерпретація основних техніко-функціональних переваг від розробки та впровадження туманної архітектури інтелектуалізованої трансформації даних агромоніторингу

Таким чином, приймаючи до уваги специфіку науково-практичної задачі та об'єкту моніторингу (польова інфраструктура агропідприємств) цієї дисертації, можна стверджувати, що техніко-функціональна організація програмно-алгоритмічного забезпечення трансформації даних агромоніторингу на основі туманної архітектури дозволяє забезпечити обчислювально-раціональне врахування агрокліматичних та техніко-технологічні факторів, що

характеризують варіативність умов вирощування різних видів с/г культур, завдяки об'єктивному розподілу процедур перетворення та аналітики даних між інфраструктурними рівнями інформаційних технологій с/г призначення, що зі свого боку позитивно впливає на мінімізацію часових затримок і залежності від нестабільності Інтернет-з'єднання. У результаті цього туманна обчислювальна архітектура є доцільною до використання в якості базової під час подальших досліджень із розробки програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу, зокрема для прогнозування врожайності с/г культур із обліком мінливих ґрунтокліматичних умов, типів агрокультур та застосовуваних агротехнічних практик їх захисту.

2.2 Структурно-функціональна модель досліджуваного програмно-алгоритмічного забезпечення

Процес розробки та дослідження програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури, зокрема під час прогнозування врожаю с/г культур, передбачає багатоетапну обробку різнорідних часово-просторових даних, які надходять від різноманітних пристроїв детектування показників і характеристик ґрунтокліматичних умов вирощування, а також облік факторів, що характеризують типи агрокультур і застосовувані практики їх захисту [101].

Таким чином, на першому етапі синтезу структурно-функціональної моделі досліджуваного програмно-алгоритмічного забезпечення виникає необхідність у побудові узагальненої контекстної структурної схеми [103, 104], яка комплексно враховує спектр інформативних параметрів і характеристик, а також факторів, що обумовлюють методологічні засади інтелектуалізованої трансформації даних агромоніторингу з точки зору забезпечення узгодженої взаємодії між ієрархічними рівнями апаратно-програмної архітектури сучасних інформаційних технологій сільськогосподарської спеціалізації. Запропоновану узагальнену

контекстну структурну схему досліджуваного процесу комплексної програмно-алгоритмічної трансформації даних агромоніторингу наведено на рисунку 2.3.

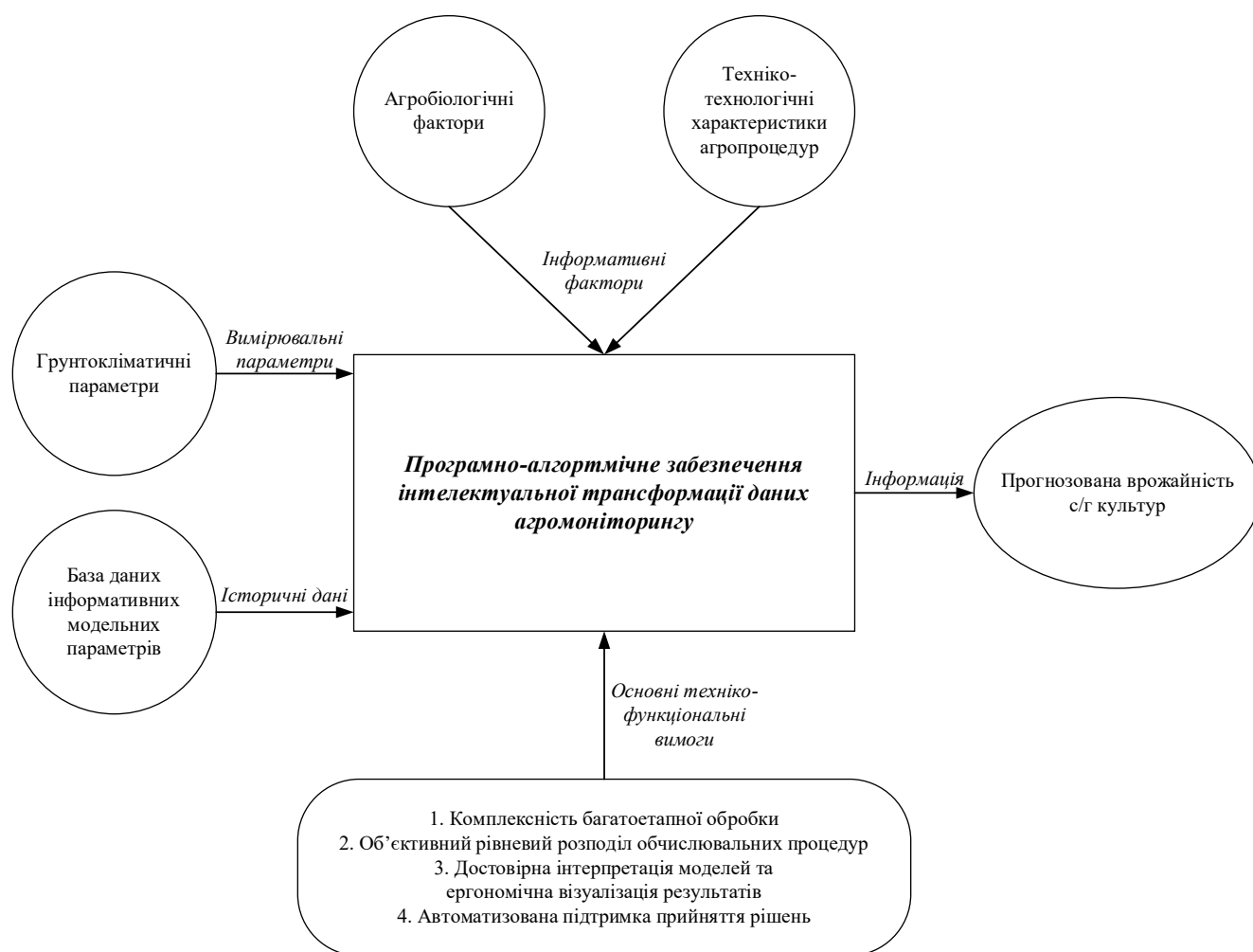


Рисунок 2.3 – Узагальнена контекстна структурна схема комплексної трансформації даних агромоніторингу

До основних ґрунтокліматичних параметрів, що детектуються польовими засобами в реальному часі, в більшості випадків відносяться показники, які характеризують температурно-вологісний режим вирощування (див. п. 1.2). Інформативними агробіологічними факторами є вид вирощуваних агрокультур та кількість днів від посівів (характеризує стадію зростання), а техніко-технологічними характеристиками агропроцедур, в більшості випадків, є фактори поливу, внесення добрив та застосування пестицидів. Своєю чергою, база інформативних модельних параметрів має містити дані щодо результатів

спостережень за динамікою ґрунтокліматичних параметрів, інформацію про інформативні агробіологічні фактори та техніко-технологічні характеристики виконаних агропроцедур, а також врожайності с/г культур за попередні цикли вирощування.

Отже, на підставі розроблених узагальненої контекстної структурної схеми комплексної трансформації даних агромоніторингу (див. рис. 2.3) та структурно-функціональної організації туманної обчислювальної архітектури агромоніторингу (див. рис. 2.1) з урахуванням попередньо одержаних результатів систематизації етапів комплексного підходу до інтелектуальної трансформації даних (див. табл. 1.4) запропоновано деталізовану структурну схему програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури [101], яку наведено на рисунку 2.4.

Запропонована деталізована структура дозволяє формалізувати розподіл комплексу обчислювальних і функціональних процедур між основними ієрархічними рівнями (польові пристрої агромоніторингу, туманні обчислення та IoT-сервер) досліджуваного програмно-апаратного забезпечення аграрного призначення з урахуванням засад їх кросрівневої мережевої взаємодії. Основним призначенням цієї структури під час подальших дисертаційних досліджень є узгодження логіки функціонування та взаємодії програмно-алгоритмічних компонент інформаційних технологій інтелектуалізованого агромоніторингу.

Це, в свою чергу, є базовою функціональною основою під час розробки блок-схеми послідовності дій у процесі проведення комплексних досліджень із розробки, розгортання та практичного застосування відповідного програмно-алгоритмічного забезпечення трансформації вимірювальних даних агромоніторингу на основі туманної архітектури задля прогнозування врожайності с/г культур, що дозволить формалізувати та візуалізувати логіку переходу від теоретичного обґрунтування до практичної реалізації основних етапів дослідницького процесу.

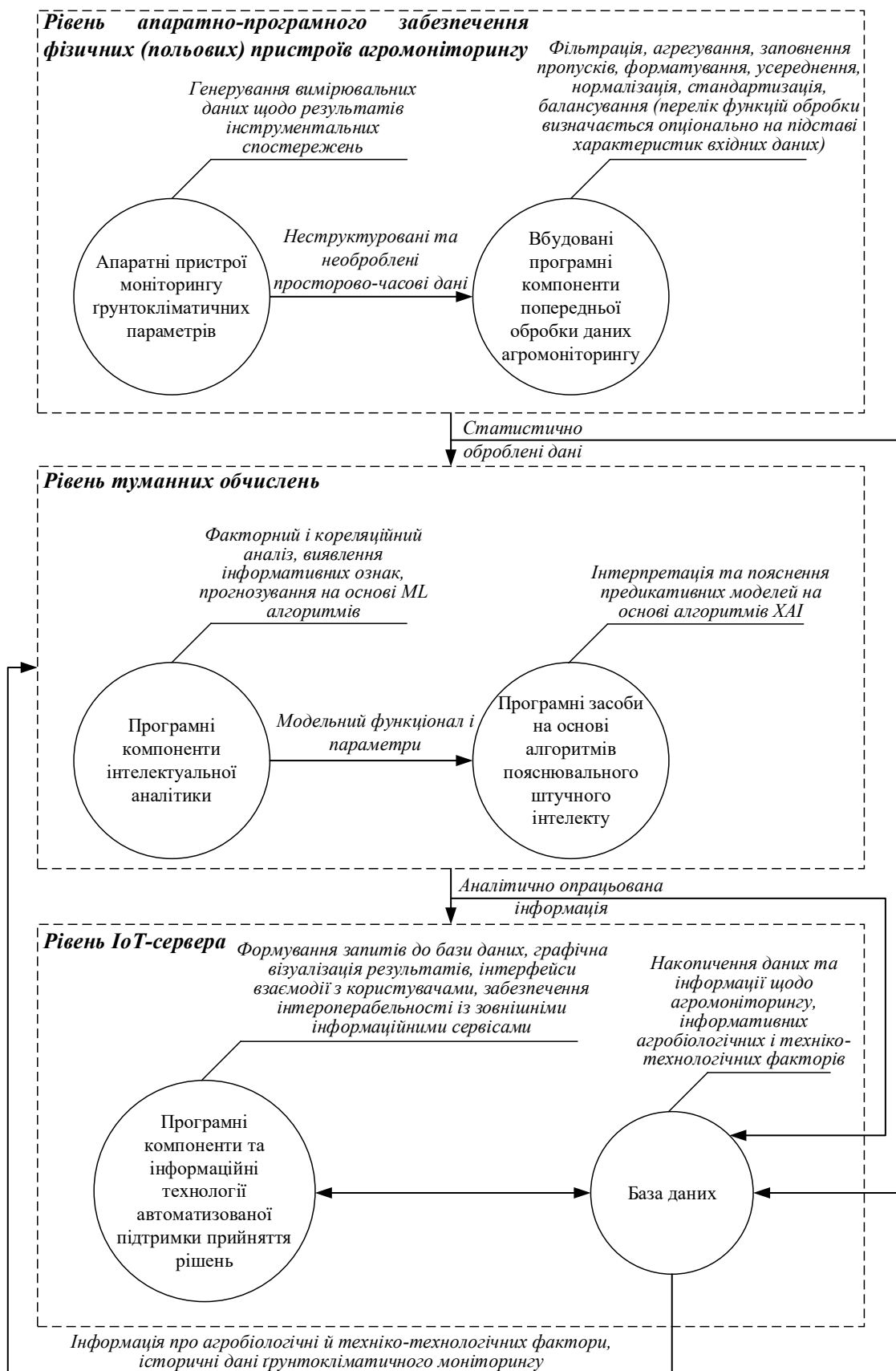


Рисунок 2.4 – Деталізована структурна схема програмно-алгоритмічного забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури

Відповідно до зазначеного вище розроблено блок-схему комплексних досліджень із обліком процедур розробки, розгортання та застосування програмно-алгоритмічного забезпечення трансформації даних агромоніторингу (див. рис. 2.5), що узгоджується з деталізованою структурою (див. рис. 2.4).

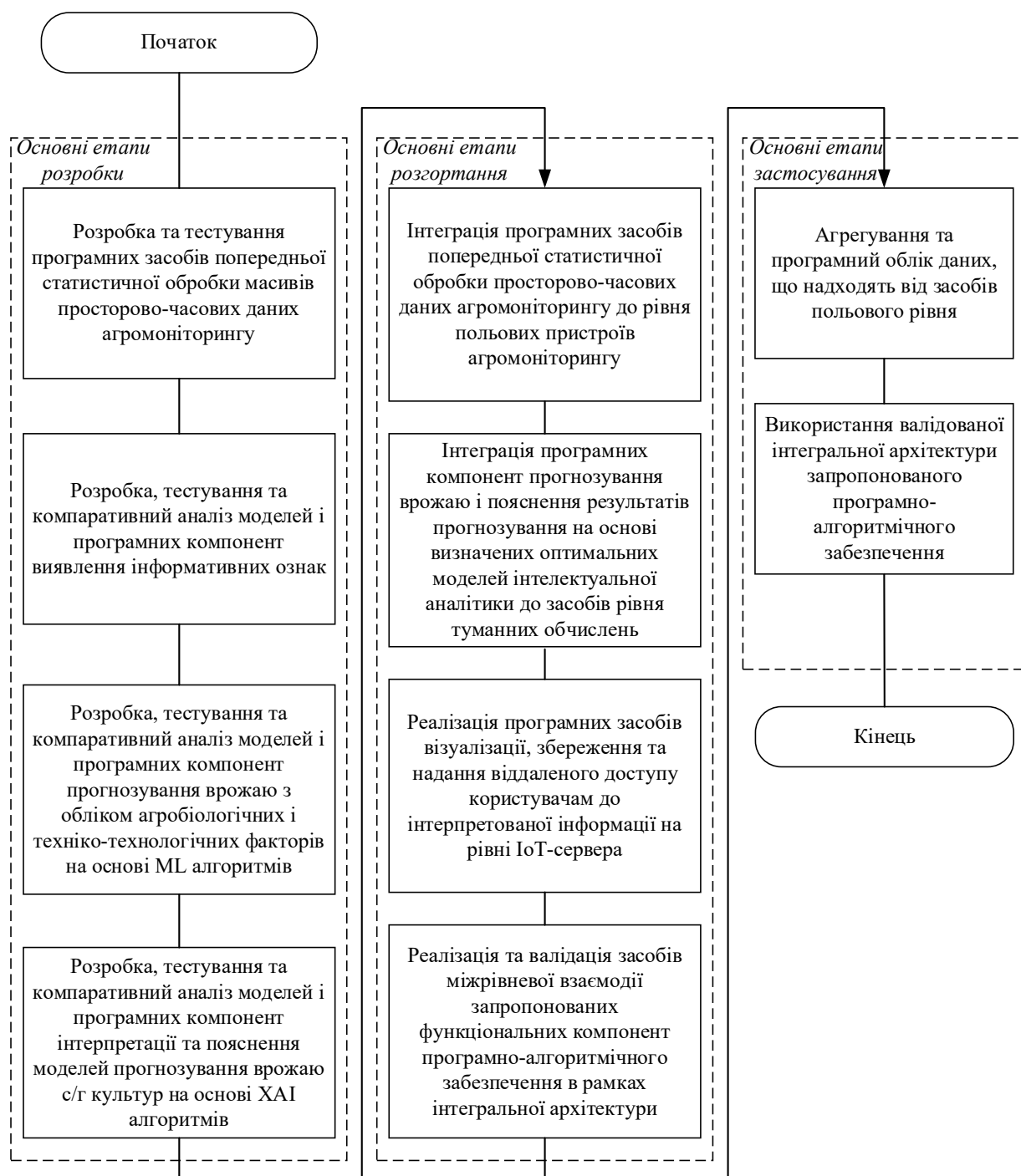


Рисунок 2.5 – Блок-схема послідовності комплексних досліджень програмно-алгоритмічного забезпечення трансформації даних агромоніторингу

Таким чином, запропоновані компоненти (див. рис. 2.3–2.5) структурно-функціональної моделі досліджуваного програмно-алгоритмічного забезпечення дозволяють реалізувати засади узгодження та контрольованості основних теоретичних і практичних етапів під час подальших досліджень із розробки, тестування та валідації програмних засобів багаторівневої обробки та аналітики даних агромоніторингу на основі туманної архітектури.

2.3 Базові модельні діаграми досліджуваного програмно-алгоритмічного забезпечення

У контексті обґрунтування базових засад функціонування досліджуваного програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу із залученням туманної обчислювальної архітектури виникає об'єктивна потреба в декомпозиції загального обчислювально-аналітичного процесу на окремі функціональні процедури [101]. Така необхідність обумовлена як просторовою розподіленістю апаратно-програмної інфраструктури, так і необхідністю надійного опрацювання даних на периферійних рівнях [105, 106] (див. рис. 2.4, польовий і туманний рівні).

Декомпозицій підхід до розв'язання задачі комплексної трансформації агромоніторингових даних під час прогнозування врожайності с/г культур дозволяє об'єктивно й ефективно розподілити обчислювально-аналітичне навантаження між функціональними компонентами відповідно до їх можливостей, зменшуючи при цьому обсяг мережевого трафіку та мінімізуючи часові затримки. У такому випадку методологічно доцільним підходом під час формалізації логіки взаємодії ієрархічних рівнів і обчислювально-аналітичних компонентів програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу є побудова та аналіз діаграми послідовності [107], що описує процеси протікання інформаційних потоків. Запропоновану діаграму послідовності обміну даними та інформаційними повідомленнями в розрізі досліджуваного програмно-алгоритмічного забезпечення наведено на рисунку 2.6.

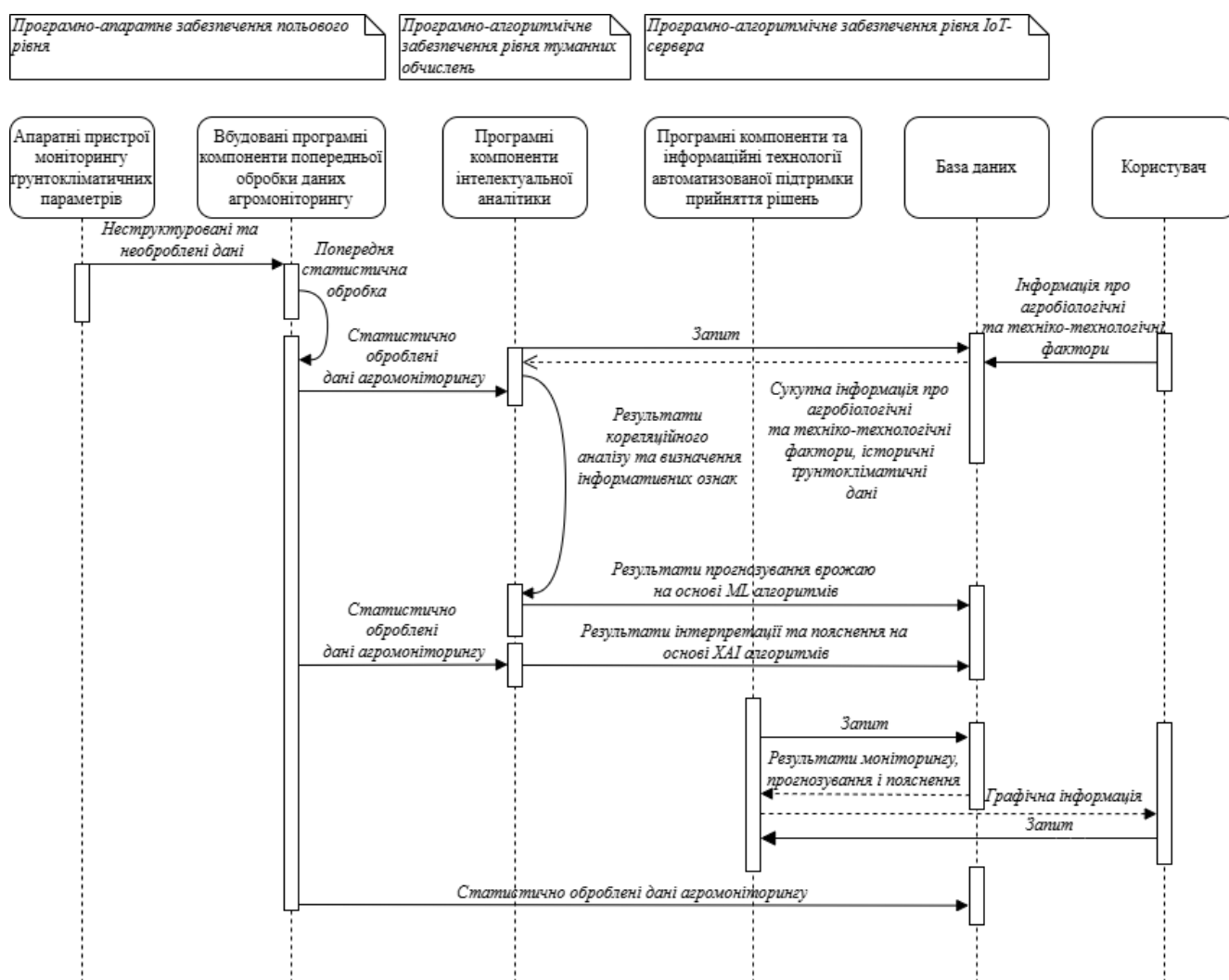


Рисунок 2.6 – Діаграма послідовності обміну даними та інформаційними повідомленнями в програмно-алгоритмічному забезпеченні комплексної трансформації даних агромоніторингу

У розробленій діаграмі (див. рис. 2.6) виділено три основних архітектурних рівні, що корелює з деталізованою структурною схемою відповідного програмно-алгоритмічного забезпечення (див. рис. 2.4), а також враховано взаємодію цієї багаторівневої архітектури з користувачами. Відповідні архітектурні рівні, своєю чергою, деталізовано на певні функціональні компоненти, що призначені для відпрацювання загальної логіки роботи досліджуваного програмно-алгоритмічного забезпечення під час прогнозування врожайності с/г культур. Зокрема, на польовому рівні у вигляді вбудованого програмного забезпечення можуть виконуватись процедури попередньої статистичної обробки

вимірювальних даних: фільтрація, агрегування, заповнення пропусків, форматування, усереднення, нормалізація, стандартизація або балансування. Перелік процедур попередньої обробки даних, що мають бути реалізовані, є опціональним і повинен визначатись із урахуванням характеристик джерел (апаратні пристрої агромоніторингу), які генерують ці дані. На рівні туманних обчислень реалізуються процедури, які пов'язані з інтелектуальним аналізом даних щодо прогнозування врожайності с/г культур та поясненням одержаних моделей предикативної аналітики, а саме: факторний і кореляційний аналіз, виявлення інформативних ознак, прогнозування на основі ML алгоритмів, а також інтерпретація та пояснення прогнозних моделей на основі алгоритмів ХАІ. У такому випадку, на рівень IoT-сервера покладаються задачі агрегування та зберігання даних, обробки системних і користувацьких запитів та графічна інтерпретація одержаних обчислювально-аналітичних результатів.

На етапах розробки та тестування відповідного програмно-алгоритмічного забезпечення до бази даних має бути внесена інформація про агробіологічні та техніко-технологічні фактори процесу вирощування. Під час застосування розробленого програмно-алгоритмічного рішення має бути забезпечена підтримка можливості користувацького корегування внесеної інформації виходячи із засад адаптації та масштабування прикладного застосування розробки.

Таким чином, запропонована діаграма послідовності (див. рис. 2.6) дозволила відобразити черговість обміну даними та інформаційними повідомленнями між архітектурними рівнями та функціональними модулями досліджуваного програмно-алгоритмічного забезпечення, формалізувати декомпоновану логіку комплексної обробки та ретрансляції даних, а також відобразити інтеграційні процеси генерування та обробки системних і користувацьких запитів у межах комплексної архітектури.

Окремої уваги в контексті опису та аналізу методологічних засад агрегаційних та обчислювально-аналітичних процедур багаторівневої інтелектуалізованої трансформації даних агромоніторингу потребує деталізація процесів системної й користувацької взаємодії щодо руху даних і протікання

інформаційних потоків. У такому випадку одним із найбільш доцільних є підхід на основі побудови діаграм руху даних [108]. Запропоновану діаграму інформаційних потоків під час багаторівневої трансформації даних агромоніторингу наведено на рисунку 2.7.

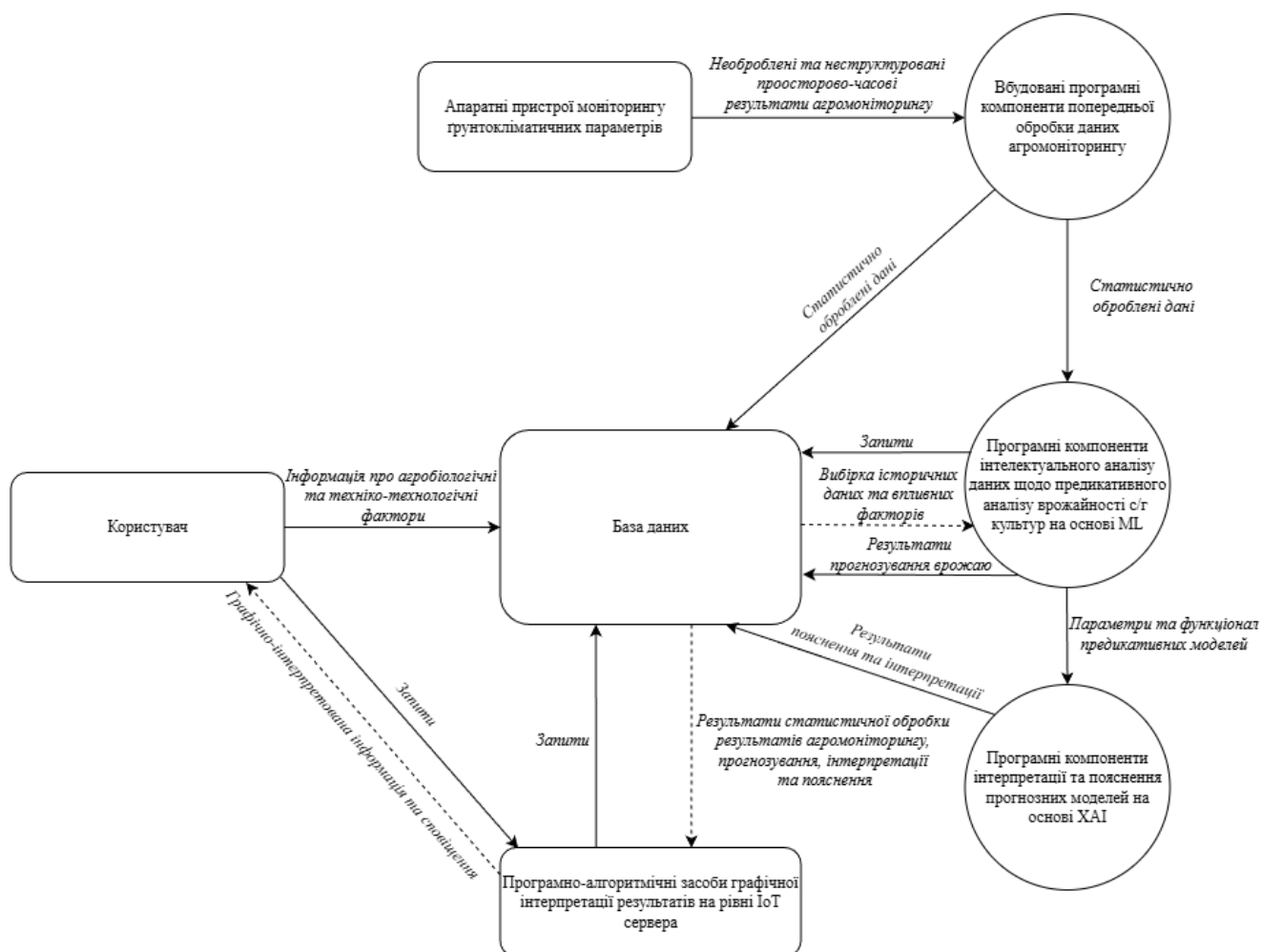


Рисунок 2.7 – Діаграма руху інформаційних потоків під час багаторівневої трансформації даних агромоніторингу

Розроблена та наведена на рисунку 2.7 діаграма руху інформаційних потоків дозволила реалізувати засади масштабування рівня деталізації теоретико-прикладних аспектів подальшої розробки, розгортання та застосування досліджуваного програмно-алгоритмічного забезпечення під час протікання агрегаційних і трансформаційних процесів системного інтелектуалізованого агротехнічного моніторингу з урахуванням характерних особливостей задачі

прогнозування врожайності с/г культур.

Крім того, одним із ключових аспектів побудови діаграми руху даних та інформації є те, що вона дала змогу формалізувати та графічно інтерпретувати шляхи й логіку проходження інформаційних потоків між рівнями та функціональними компонентами досліджуваного програмно-алгоритмічного забезпечення з урахуванням техніко-функціональних аспектів застосування туманної архітектури. Додатково варто акцентувати увагу на тому, що на підставі побудови та аналізу цієї діаграми (див. рис. 2.7) було ідентифіковано інтеграційні засади бази даних із обчислювально-аналітичними модулями програмно-алгоритмічного забезпечення, що розробляється й досліджується в цій дисертаційній роботі.

Наступним кроком дослідницького етапу побудови та аналізу базових модельних діаграм досліджуваного програмно-алгоритмічного забезпечення агропризначення є обґрунтування сценаріїв і маршрутів обробки даних та інформаційних повідомлень із урахуванням засад розгалудження обчислювально-аналітичних процедур та виконання умов, що мають місце в межах процесу комплексної інтелектуалізованої трансформації даних під час прогнозування врожайності с/г культур. У такому випадку практично доцільним підходом є створення та аналіз діаграм діяльності [109], що дозволяють структуровано візуалізувати послідовність відпрацювання операцій агрегування й обробки даних та інформації з урахуванням системно-користувацької взаємодії. Виходячи з цього, було розроблено діаграму діяльності програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу, як показано на рисунку 2.8.

Розроблена діаграма діяльності під час комплексної трансформації даних агромоніторингу, яку наведено на рисунку 2.8, дозволила формалізувати логіку системно-користувацької взаємодії з урахуванням засад агрегування, попередньої й інтелектуальної обробки даних агромоніторингу, автоматизованої інтерпретації, пояснення прогнозних моделей та графічної візуалізації результатів

прогнозування врожаю, а також розгортання й застосування досліджуваного програмно-алгоритмічного забезпечення.

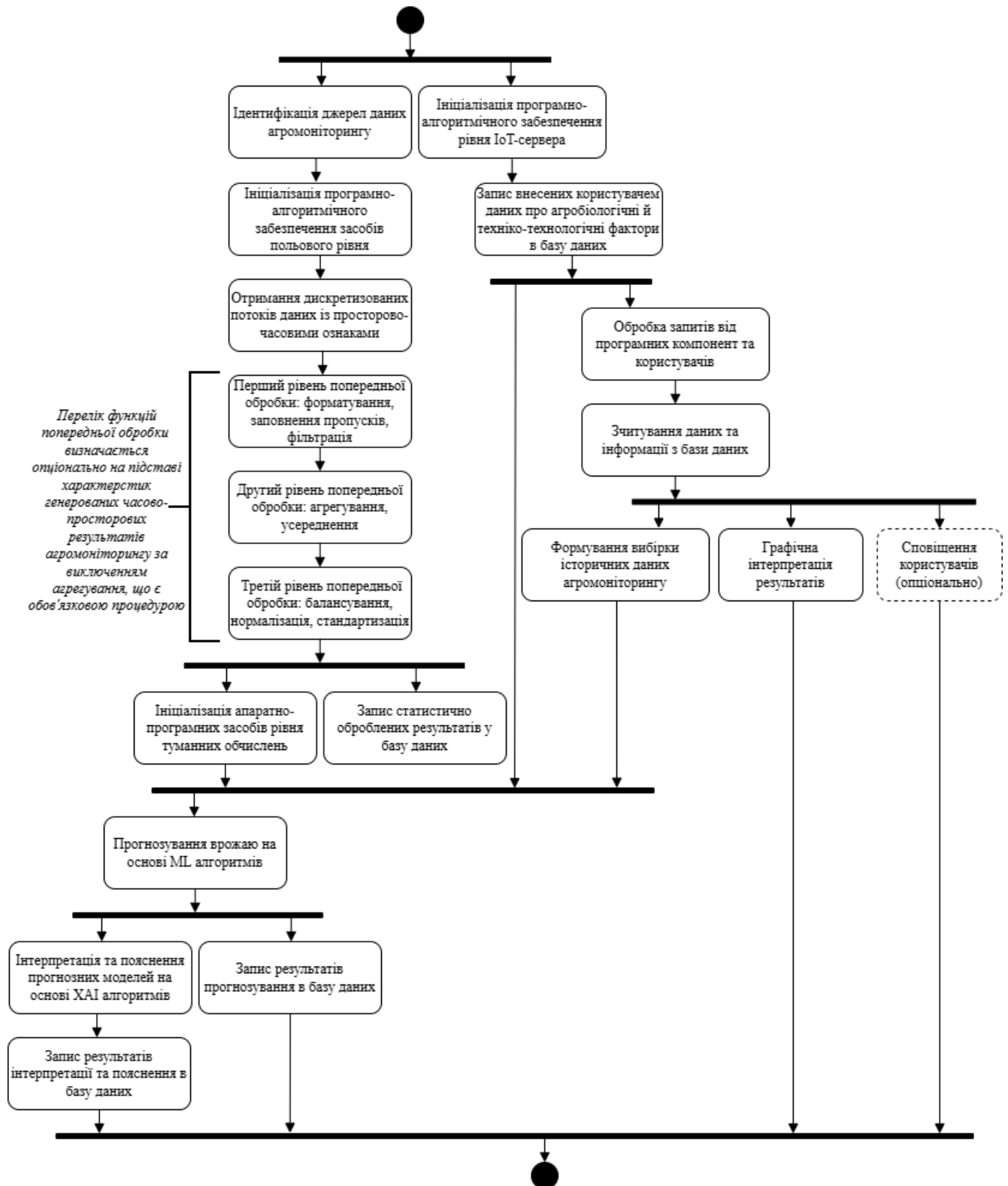


Рисунок 2.8 – Діаграма діяльності програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу

Таким чином, результати досліджень, які присвячено розробці та аналізу базових модельних діаграм, що характеризують теоретико-прикладні аспекти функціонування програмно-алгоритмічного забезпечення комплексної трансформації даних агромоніторингу дозволили структуровано й системно інтерпретувати та деталізувати в графічній формі основні сценарії його розробки, розгортання та функціонування під час прогнозування врожайності с/г культур. Одержані результати досліджень цього підрозділу дисертаційної роботи становлять собою методологічне підґрунтя під час подальших досліджень із розробки, тестування та валідації програмних компонент архітектурних рівнів, а також їх кросрівневої взаємодії в межах програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу.

2.4 Інформаційна модель досліджуваного програмно-алгоритмічного забезпечення

Як зазначалось вище, під час створення програмно-алгоритмічного забезпечення трансформації вимірювальних даних агромоніторингу виникає потреба в інтелектуалізованому, багатоетапному й комплексному опрацюванні динамічних і гетерогенних потоків просторово-часових результатів спостережень. У зв'язку з цим постає необхідність у обґрунтуванні інформаційної моделі, яка формалізує агрегаційні та обчислювально-аналітичні процедури у вигляді логічних і математичних операторів, що дозволяє відобразити черговість і функціональний зв'язок перетворень на різних архітектурних рівнях [110, 111].

У програмно-алгоритмічних рішеннях, що входять до складу інформаційних технологій агромоніторингу, синтезованих на базі туманної архітектури, важливу роль відіграють процеси наскрізної обробки та кросрівневого обміну даними й інформаційними повідомленнями. Таким чином, із метою виконання вимог щодо узгодженості агрегаційних і обчислювально-аналітичних етапів трансформації даних агромоніторингу та структурно-функціональної цілісності їх обробки виникає необхідність у синтезі інформаційної моделі досліджуваного програмно-

алгоритмічного забезпечення у вигляді формалізованого опису основних процедур і функцій, що реалізуються на відповідних архітектурних рівнях.

З урахуванням засад потокової обробки даних [112] та розроблених у цій роботі модельних діаграм (див. рис. 2.6–2.8), що визначають основні методологічні засади багатоетапної комплексної трансформації даних агромоніторингу, які являють собою серії результатів інструментальних спостережень за інформативними агрокліматичними параметрами, запропоновано узагальнений вираз, що в композиційній формі відображає логіку реалізації агрегаційних і обчислювально-аналітичних процедур [101]:

$$Results = (F_{interaction} \circ F_{analytics} \circ F_{preprocessing})(Data, Factors), \quad (2.1)$$

де *Results* – інтерпретовані результати багатоетапної обробки даних, що доступні для користувацького використання; $F_{interaction}$ – процедури і функції графічної візуалізації результатів, реалізації інтерфейсів взаємодії з користувачами та забезпечення інтероперабельності із зовнішніми інформаційними сервісами; $F_{analytics}$ – сукупність процедур і функцій, що реалізують обчислювально-аналітичні засади інтелектуалізованої обробки даних на основі ML та XAI алгоритмів; $F_{preprocessing}$ – сукупність процедур і функцій, що реалізують обчислювальні засади попередньої статистичної обробки даних; *Data* – просторово-часові дані агромоніторингу, в тому числі історичні; *Factors* – агробіологічні та техніко-технологічні фактори; символ $\circ$ позначає функцію композиції.

Узагальнена композиційна форма представлення процесу багатоетапної трансформації даних агромоніторингу (2.1) дозволяє відобразити інформаційний ланцюг системних перетворень, що відбуваються в досліджуваному програмно-алгоритмічному забезпеченні від моменту детектування вихідних сигналів апаратних пристроїв польового рівня до візуалізації результатів, які можуть бути використанні в якості основних під час автоматизованої підтримки агротехнічних

рішень. Проте задля структуризації та однозначності інтерпретації процесу комплексної трансформації даних агромоніторингу з урахуванням дестабілізуючого впливу факторів агробіологічного та техніко-технологічного характеру ця функціональна залежність потребує подальшої деталізації у відповідності до визначених архітектурних рівнів програмно-алгоритмічного забезпечення.

Узагальнений функціональний опис рівня апаратно-програмного забезпечення польових пристроїв агромоніторингу може бути представлений у такому вигляді:

$$Data^{(0)} \xrightarrow{F_{preprocessing}} Data^{(1)}, \quad (2.2)$$

де $Data^{(0)}$ – неструктурований і необроблений потік просторово-часових даних агромоніторингу; $F_{preprocessing}$ – сукупність процедур і функцій, що реалізують обчислювальні засади попередньої статистичної обробки даних; $Data^{(1)}$ – результати агромоніторингу після попередньої статистичної обробки.

Як було зазначено вище, перелік процедур і функцій попередньої обробки даних, що мають бути реалізовані під час синтезу програмно-алгоритмічного забезпечення комплексної трансформації агромоніторингових інформаційних потоків, є опціональним і повинен визначатись із урахуванням характеристик джерел (апаратні пристрої агромоніторингу), які генерують ці дані. В такому випадку композиційна форма обов'язкового агрегування й опціональної сукупності процедур попередньої обробки може бути представлена у вигляді:

$$F_{preprocessing} = F_{aggregate} \text{ AND Optional } \{ F_{format}, F_{filling}, F_{filter}, F_{averaging}, F_{balance}, F_{normalize}, F_{standartize} \}, \quad (2.3)$$

де $F_{aggregate}$ – функція агрегування даних; F_{format} – функція форматування; $F_{filling}$ –

функція заповнення пропусків у масивах даних; F_{filter} – функція фільтрації; $F_{averaging}$ – функції просторового й часового усереднення даних; $F_{balance}$ – функція балансування даних; $F_{normalize}$ – функція нормалізації даних; $F_{standartize}$ – функція стандартизації даних.

Процедура агрегування даних передбачає їх накопичення у відповідності до таких ознак, як час і локація одержання даних, тип с/г культур, що у загальному випадку описується наступним чином:

$$Data^{(non-aggregated)} \xrightarrow{F_{aggregate}=Features\{Time, Location, Type\}} Data^{(aggregated)}, \quad (2.4)$$

де $Data^{(non-aggregated)}$ – неагреговані дані; $F_{aggregate}$ – функція агрегування даних у відповідності до наступних ознак ($Features$): час ($Time$) і локація ($Location$) одержання даних, тип ($Type$) с/г культур; $Data^{(aggregated)}$ – агреговані у відповідності до зазначених ознак дані.

Призначення функції форматування під час реалізації загального ланцюга попередньої обробки даних (2.3) полягає в перетворенні типів даних і структури їх представлення і в загальному випадку може бути подана у вигляді:

$$Data^{(unformatted)} \xrightarrow{F_{format}} Data^{(formatted)}, \quad (2.5)$$

де $Data^{(unformatted)}$ – невідформатований масив даних; F_{format} – функція форматування; $Data^{(formatted)}$ – відформатовані дані.

Функція заповнення пропусків може бути реалізована із використанням різних підходів [113], як в узагальненому вигляді наведено нижче:

$$Data^{(unfilled)} \xrightarrow{F_{filling}=Optional\{Statistical, Interpolation, Regression, ML-based\}} Data^{(filled)}, \quad (2.6)$$

де $Data^{(unfilled)}$ – масив даних без заповнення пропусків; $F_{filling}$ – функція заповнення пропусків у масивах даних, що може бути реалізована опціонально на основі статистичних, інтерполяційних, регресійних або ML методів; $Data^{(filled)}$ – масив даних із заповненими пропусками.

Процедура фільтрації даних передбачає зменшення (видалення) шумової складової результатів детектування агрокліматичних параметрів, а також виявлення та вилучення аномальних значень у масивах даних [114]. В узагальненому вигляді ця процедура може бути представлена таким чином:

$$Data^{(unfiltered)} \xrightarrow{F_{filter}=Optional\{Linear, Nonlinear, Statistical, Adaptive\}} Data^{(filtered)}, \quad (2.7)$$

де $Data^{(unfiltered)}$ – масив невідфільтрованих даних; F_{filter} – функція фільтрації масивів даних, що може бути реалізована опціонально на основі лінійних, нелінійних, статистичних або адаптивних методів; $Data^{(filtered)}$ – масив відфільтрованих даних.

Усереднення даних відбувається за критеріальними ознаками часу та локацій їх одержання, що в загальному вигляді може бути представлено наступним функціональним ланцюгом:

$$Data^{(unaveraged)} \xrightarrow{F_{averaging}^{time}} LocalData^{(averaged)} \xrightarrow{F_{averaging}^{space}} Data^{(averaged)}, \quad (2.8)$$

де $Data^{(unaveraged)}$ – неусереднені дані; $F_{averaging}^{time}$ – функція усереднення даних у часі; $LocalData^{(averaged)}$ – результати усереднення даних у часі; $F_{averaging}^{space}$ – функція усереднення даних у просторі; $Data^{(averaged)}$ – усереднені в часі та просторі дані.

Процедура балансування даних передбачає усунення або мінімізацію диспропорцій у кількості зразків агрегованих даних під час подальшої розробки програмних компонент прогнозування врожайності на основі ML алгоритмів. Ця процедура може бути реалізована на основі зменшення числа зразків більшого

класу, збільшення числа зразків меншого класу або гібридних методів [115], як в узагальненому функціональному вигляді показано нижче:

$$Data^{(imbalanced)} \xrightarrow{F_{balance}=Optional\{Undersampling, Oversampling, Hybrid\}} Data^{(balanced)}, \quad (2.9)$$

де $Data^{(imbalanced)}$ – ряди результатів спостережень до балансування; $F_{balance}$ – функція балансування даних, що може бути реалізована опціонально на основі зазначених вище методів або шляхом їх комбінування; $Data^{(balanced)}$ – ряди результатів спостережень після балансування.

Процедури нормалізації та стандартизації виконуються для серій результатів спостережень за кожним окремо взятим інформативним агрокліматичним параметром ($Data_i$), що впливає в підсумку на ефективність прогнозування врожайності с/г культур. Нормалізація результатів спостережень за окремо взятим агрокліматичним параметром у заданому діапазоні може бути реалізована наступним чином [116]:

$$Data_i^{(unnormalized)} \rightarrow \frac{Data_i^{(unnormalized)} - \min(Data_i^{(unnormalized)})}{\max(Data_i^{(unnormalized)}) - \min(Data_i^{(unnormalized)})} \rightarrow Data_i^{(normalized)}, \quad (2.10)$$

$Data_i^{(unnormalized)}$ – результати спостережень за кожним окремо взятим інформативним агрокліматичним параметром до нормалізації; $\min(Data_i^{(unnormalized)})$ та $\max(Data_i^{(unnormalized)})$ – мінімальний і максимальний результати спостережень за кожним окремо взятим інформативним агрокліматичним параметром, відповідно; $Data_i^{(normalized)}$ – нормалізовані результати спостережень за кожним окремо взятим інформативним агрокліматичним параметром.

Стандартизація результатів спостережень за окремо взятим інформативним агрокліматичним параметром у заданому діапазоні може бути реалізована у відповідності до виразу [116], який в узагальненому вигляді є таким:

$$Data_i^{(unstandardized)} \rightarrow \frac{Data_i^{(unstandardized)} - \overline{Data_i}}{\sigma_{Data_i}} \rightarrow Data_i^{(standardized)}, \quad (2.11)$$

$Data_i^{(unstandardized)}$ – результати спостережень за кожним окремо взятим інформативним агрокліматичним параметром до стандартизації; $\overline{Data_i}$ – середнє значення серії результатів спостережень за кожним окремо взятим інформативним агрокліматичним параметром; σ_{Data_i} – стандартне відхилення результатів спостережень за кожним окремо взятим інформативним агрокліматичним параметром у серії; $Data_i^{(standardized)}$ – стандартизовані результати спостережень за кожним окремо взятим інформативним агрокліматичним параметром.

Таким чином, функціональні залежності (2.4)–(2.11) являють собою повний перелік опціональних (за виключенням агрегування, що є обов’язковим), виходячи з характеристик джерел генерування даних агромоніторингу, процедур і функцій, які можуть бути реалізовані під час попередньої статистичної обробки результатів агромоніторингу на рівні польових пристроїв у контексті досліджуваного програмно-алгоритмічного забезпечення.

Етап міжрівневої взаємодії апаратно-програмного забезпечення польового та туманного рівнів передбачає мережевий обмін пакетами даних, що дискретизовані в часі з певною періодичністю, як зазначено нижче:

$$Data^{(1)} \xrightarrow{F_{discretization}} Data^{(1)}(time_j), \quad (2.12)$$

де $Data^{(1)}$ – результати агромоніторингу після попередньої статистичної обробки; $F_{discretization}$ – функціональне забезпечення процедури дискретизації під час мережевого обміну даними агромоніторингу; $Data^{(1)}(time_j)$ – дискретизовані пакети даних після попередньої статистичної обробки на рівні апаратно-програмного забезпечення польових пристроїв; $time_j$ – дискретні часові відліки.

Виходячи з цього, узагальнений функціональний опис рівня туманних обчислень із урахуванням одержаних пакетів попередньо оброблених даних та інформації про агробіологічні та техніко-технологічні характеристики агропроцедур, а також масивів історичних агрокліматичних даних, може бути представлений у наступному вигляді:

$$\left. \begin{array}{l} Data^{(1)}(time_j) \\ Data_{historical} \\ Factors \end{array} \right\} \xrightarrow{F_{analytics}} Data^{(2)}, \quad (2.13)$$

де $Data^{(1)}(time_j)$ – дані, що надійшли на рівень туманних обчислень після попередньої обробки на рівні апаратно-програмного забезпечення польових пристроїв; $Data_{historical}$ – масиви історичних агрокліматичних даних; $Factors$ – інформація про агробіологічні та техніко-технологічні характеристики агропроцедур; $F_{analytics}$ – сукупність процедур і функцій, що реалізують обчислювально-аналітичні засади інтелектуальної обробки даних та інтерпретації моделей на основі ML та XAI алгоритмів; $Data^{(2)}$ – результати інтелектуальної аналітики щодо прогнозування врожаю.

Функціональний опис (2.13) у деталізованому вигляді, який реалізується засобами рівня туманних обчислень, є таким:

$$\left. \begin{array}{l} Data^{(1)}(time_j) \\ Data_{historical} \\ Factors \end{array} \right\} \xrightarrow{F_{features}} Data^{(informative)} \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{F_{prediction}} Data^{(2)}, \underbrace{Model^{(structure, parameters)}}_{\substack{\downarrow F_{XAI} \\ Explanations}}, \quad (2.14)$$

де $F_{features}$ – сукупність процедур і функцій, що реалізують методологічні засади факторного і кореляційного аналізу та виявлення інформативних ознак; $Data^{(informative)}$ – результати факторного і кореляційного аналізу та виявлення

інформативних ознак; $F_{prediction}$ – сукупність процедур і функцій, що реалізують методологічні засади прогнозування врожаю с/г культур на основі ML алгоритмів; $Data^{(2)}$ – результати інтелектуальної аналітики щодо прогнозування врожайності; $Model^{(structure, parameters)}$ – параметри та структура предикативних ML моделей; F_{XAI} – сукупність процедур і функцій, що реалізують інтерпретацію та пояснення предикативних ML моделей на основі XAI алгоритмів; $Explanations$ – результати інтерпретації та пояснення предикативних моделей.

Варто зазначити, що логіка роботи розроблюваного програмно-алгоритмічного забезпечення передбачає запис до бази даних одержаних результатів агромоніторингу після попередньої статистичної обробки ($Data^{(1)}$) та результатів інтелектуальної аналітики щодо прогнозування врожаю ($Data^{(2)}$), які дискретизовані в часі з певною періодичністю, внесеної користувачем інформації про агробіологічні й техніко-технологічні характеристики агропроцедур ($Factors$), а також результатів пояснення й інтерпретації прогнозних моделей ($Explanations$). В узагальненому вигляді такий функціонал є наступним:

$$\left. \begin{array}{l} Data^{(1)} \xrightarrow{F_{discretization}} Data^{(1)}(time_j) \\ Data^{(2)} \xrightarrow{F_{discretization}} Data^{(2)}(time_k) \\ Factors \\ Explanations \end{array} \right\} \xrightarrow{F_{write}} DB, \quad (2.15)$$

де $Data^{(1)}$, $Data^{(2)}$ – результати агромоніторингу після попередньої статистичної обробки та прогнозування врожайності, відповідно; $F_{discretization}$ – функціональне забезпечення процедури дискретизації; $Data^{(1)}(time_j)$, $Data^{(2)}(time_k)$ – дискретизовані пакети даних результатів агромоніторингу після попередньої статистичної обробки та інтелектуальної аналітики щодо прогнозування врожаю, відповідно; $Factors$ – інформація про агробіологічні та техніко-технологічні характеристики агропроцедур; $Explanations$ – результати інтерпретації та

пояснення; F_{write} – функціональне забезпечення процедури запису даних і інформації до бази даних; DB – база даних.

Свою чергою, функціональне забезпечення бази даних у межах розроблюваного програмно-алгоритмічного забезпечення агромоніторингу передбачає можливість обробки системних запитів від компонент рівня туманних обчислень та IoT-сервера щодо зчитування даних, які були попередньо до неї внесені (2.15), для їх подальшої обробки та візуалізації (див. рис. 2.6 і 2.7) у відповідності до зазначеного нижче функціонального ланцюга:

$$\begin{aligned} &Fog^{(components)} \xrightarrow{F_{query}(Factors, Data_{historical})} DB \xrightarrow{F_{read}} Factors, Data_{historical} \\ &IoT-server^{(components)} \xrightarrow{F_{query}(Data^{(1)}, Data^{(2)}, Explanations)} DB \xrightarrow{F_{read}} , \quad (2.16) \\ &\xrightarrow{F_{read}} Data^{(1)}, Data^{(2)}, Explanations \end{aligned}$$

де $Fog^{(components)}$, $IoT-server^{(components)}$ – програмно-алгоритмічні компоненти рівня туманних обчислень й IoT-сервера, відповідно; $F_{query}(Factors, Data_{historical})$, $F_{query}(Data^{(1)}, Data^{(2)}, Explanations)$ – функціональне забезпечення обробки системних запитів до бази даних від програмно-алгоритмічних компонент рівня туманних обчислень й IoT-сервера, відповідно; DB – база даних; F_{read} – функціональне забезпечення процедури зчитування даних та інформації з бази даних; $Factors$ – внесена користувачем інформація про агробіологічні та техніко-технологічні характеристики агропроцедур; $Data_{historical}$ – масиви історичних агрокліматичних даних; $Data^{(1)}$, $Data^{(2)}$ – результати агромоніторингу після статистичної обробки та інтелектуальної аналітики прогнозування врожаю, відповідно; $Explanations$ – результати інтерпретації та пояснення;

На рівні IoT-сервера передбачається графічна візуалізація результатів агромоніторингу та їх інтелектуальної обробки щодо прогнозування врожайності с/г культур, які зчитуються з бази даних (2.16), а також надання до них доступу користувачам за відповідними запитами, у відповідності до виразу нижче:

$$DB \xrightarrow{F_{read}} (Data^{(1)}, Data^{(2)}) \xrightarrow{F_{graphs}} \underbrace{Plot(Data^{(1)}, Data^{(2)})}_{\substack{F_{query}^{user} \uparrow \downarrow F_{visualization} \\ User}}, \quad (2.17)$$

DB – база даних; F_{read} – функціональне забезпечення процедури зчитування даних та інформації з бази даних; $Data^{(1)}$, $Data^{(2)}$ – результати агромоніторингу після попередньої статистичної обробки та інтелектуальної аналітики щодо прогнозування врожайності, відповідно; F_{graphs} – функціональне забезпечення процедури графічної інтерпретації результатів трансформації даних агромоніторингу; F_{query}^{user} – функціональне забезпечення обробки користувацьких запитів; $F_{visualization}$ – функціональне забезпечення надання доступу користувачам до графічної інтерпретації результатів агромоніторингу; $User$ – користувач.

Додатково реалізоване функціональне забезпечення має передбачати сповіщення користувачів засобами електронної пошти щодо результатів прогнозової аналітики та інтерпретації предикативних моделей, що відповідає вимогам автоматизованої підтримки прийняття рішень. Це може бути реалізовано у відповідності до наступного логічного ланцюга:

$$DB \xrightarrow{F_{read}} (Data^{(2)}, Explanations) \xrightarrow{F_{email}} \underbrace{Email(Data^{(2)}, Explanations)}_{\substack{\downarrow F_{send} \\ User}}, \quad (2.18)$$

DB – база даних; F_{read} – функціональне забезпечення процедури зчитування інформації з бази даних; $Data^{(2)}$ – результати агромоніторингу після інтелектуальної аналітики щодо прогнозування врожайності; $Explanations$ – результати інтерпретації та пояснення; F_{email} – функціональне забезпечення процедури формування повідомлення засобами електронної пошти; F_{send} – функціональне забезпечення відправлення повідомлення; $User$ – користувач.

Таким чином, запропонований формалізований опис (2.2) – (2.18) являє собою деталізацію інформаційної моделі розроблюваного програмно-

алгоритмічного забезпечення трансформації даних агромоніторингу з урахуванням засад інтелектуалізованості, комплексності та об'єктивності міжрівневого розподілу обчислювально-аналітичних процедур. Запропонована інформаційна модель виступає теоретико-прикладним базисом під час реалізації засад структуризації та параметризації в процесі досліджень програмних компонент збору, обробки й інтерпретації результатів агромоніторингу з урахуванням специфіки задачі прогнозування врожайності с/г культур.

2.5 Висновки за другим розділом

У цьому розділі дисертаційної роботи запропоновано та досліджено підходи до розв'язання одного з основних дослідницьких завдань, що полягало в обґрунтуванні структурно-функціональної організації та інформаційної моделі розроблюваного програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу під час прогнозування врожайності сільськогосподарських культур на основі туманної архітектури. Основні теоретичні та прикладні результати розробки та досліджень цього розділу полягають у наступному:

1. Розроблено узагальнену туманну архітектуру досліджуваного програмно-алгоритмічного забезпечення процесу багаторівневої інтелектуалізованої трансформації даних агромоніторингу та обґрунтовано переваги від її практичної реалізації, що дозволяє реалізувати децентралізований підхід із логікою на основі запитів ініціювання комплексу обчислювально-аналітичних процедур під час прогнозування врожайності сільськогосподарських культур у мінливих агрокліматичних умовах.

2. Запропоновано та проаналізовано структурно-функціональну модель досліджуваного програмно-алгоритмічного забезпечення для цілей інтелектуалізованого прогнозного агромоніторингу, що враховує теоретико-прикладні засади імплементації та функціонування рівнів польових пристроїв, туманних обчислень й IoT-сервера, а також їх міжрівневу взаємодію на етапах

розробки, розгортання та практичного застосування. Це дозволило обґрунтувати методологічні засади узгодження та контрольованості етапів подальшої розробки та тестування програмно-алгоритмічних засобів багаторівневої трансформації та аналітики даних під час прогнозування врожайності сільськогосподарських культур на основі туманної архітектури з урахуванням факторів варіативності агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту.

3. Синтезовано базове методологічне забезпечення програмно-алгоритмічних засобів опрацювання результатів агромоніторингу, а саме: діаграму послідовності обміну даними та інформаційними повідомленнями в програмно-алгоритмічному забезпеченні, діаграму руху інформаційних потоків під час багаторівневої трансформації даних агромоніторингу та діаграму діяльності програмно-алгоритмічного забезпечення комплексної трансформації даних. Це дозволило системно й структуровано деталізувати та інтерпретувати основні сценарії розробки, імплементації та функціонування досліджуваного програмно-алгоритмічного забезпечення під час прогнозування врожайності агрокультур.

4. Розроблено інформаційну модель досліджуваного програмно-алгоритмічного забезпечення, що представлена у вигляді формалізованого опису обчислювальних і аналітичних процедур опрацювання гетерогенних потоків просторово-часових результатів агромоніторингу з урахуванням засад кросрівневої та людино-машинної взаємодії. Це дозволило відобразити черговість і функціональний зв'язок інформаційних процесів і перетворень, що відбуваються на відповідних архітектурних рівнях програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних під час прогнозування врожайності сільськогосподарських культур із урахуванням системних і користувацьких інформаційних запитів.

РОЗДІЛ 3

ПРОГРАМНІ КОМПОНЕНТИ ПОВНОЦИКЛОВОЇ ТРАНСФОРМАЦІЇ ДАНИХ АГРОМОНІТОРИНГУ ПІД ЧАС ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР

3.1 Підходи до розробки програмних компонент повноциклової трансформації даних агромоніторингу

На підставі проведених досліджень попередніх розділів цієї дисертації було встановлено, що класичні статистичні методи моделювання й трансформації даних у більшості випадків виявляються недостатньо ефективними через складність та нелінійність взаємозв'язків між агрокліматичними факторами, агротехнологічними характеристиками та кінцевим результатом – урожайністю. У цьому контексті методи ML забезпечують значно вищу гнучкість і здатність виявляти приховані закономірності.

Разом із тим, у практичному агровиробництві критично важливою є не лише точність прогнозу, а й інтерпретованість моделей. Саме тому останніми роками активно розвивається напрям ХАІ, який дозволяє підвищити прозорість роботи «чорних скриньок» ML-моделей, таких як ансамблі чи нейронні мережі. Враховуючи це, метою цього розділу є розробка та дослідження ефективності програмно-алгоритмічних засобів комплексної трансформації даних агромоніторингу, які реалізують етапи попередньої обробки (фільтрація, агрегування, форматування і стандартизація) та інтелектуальної аналітики (виявлення інформативних ознак, прогнозування на основі ML алгоритмів, інтерпретації та пояснення прогнозних моделей із застосуванням алгоритмів ХАІ) у межах повноциклової туманної архітектури [117, 118].

Отже, задля досягнення високих показників адекватності та об'єктивності обробки даних агромоніторингу реалізовано комплексний підхід, який включає етапи збору, попередньої обробки, аналізу, підготовки навчальних та тестових

наборів даних, навчання та валідації ML моделей, оцінки важливості ознак та пояснювального штучного інтелекту [117], як показано на рисунку 3.1.



Рисунок 3.1 – Узагальнена блок-схема послідовності повноциклової трансформації даних із застосуванням ML та XAI

Під час дослідження було використано Jupyter notebook та низку бібліотек Python (версія $\geq$ 3.12) [119], які забезпечили повний цикл обробки та аналізу даних. Кожна з них відіграє специфічну роль у реалізації запропонованої методології, як це описано нижче та реалізовано в додатку В.1.

Базові інструменти обробки даних включають бібліотеки "numpy ($\geq$ 2.3.2, $<$ 3.0.0)" [120] та "pandas ($\geq$ 2.3.2, $<$ 3.0.0)" [121]. NumPy застосовано для роботи з багатовимірними масивами та ефективних числових обчислень, а також для відпрацювання базових математичних операцій, що забезпечують основу для побудови моделей на основі алгоритмів ML. Бібліотека Pandas використовувалася для обробки табличних даних у вигляді DataFrame, а саме: очищення, агрегація, об'єднання джерел інформації, а також підготовка фінального набору даних.

Для візуалізації та дослідження даних імпортовано бібліотеки "matplotlib ($\geq$ 3.10.5, $<$ 4.0.0)" [122] та "seaborn ($\geq$ 0.13.2, $<$ 0.14.0)" [123]. Matplotlib є базовою бібліотекою під час побудови графіків і діаграм, яку використано для відображення трендів та розподілів даних. Seaborn, своєю чергою, розширює можливості Matplotlib і застосовується для побудови статистичних візуалізацій (кореляційні матриці, розподіли, діаграми розсіювання).

Для машинного навчання використано бібліотеку "scikit-learn ($\geq$ 1.7.1, $<$ 2.0.0)" [124]. Також із цієї бібліотеки для попередньої обробки

застосовано `StandardScaler` (масштабування ознак), `train_test_split` (розділення даних на тренувальну та тестову вибірки). Із алгоритмів регресії в цьому дослідженні було використано: `Linear Regression`, `Random Forest Regressor`, `Gradient Boosting Regressor`, `Decision Tree Regressor`, `SVR`, `KNeighbors Regressor`, `Bagging Regressor`, `MLP Regressor`. Такий вибір ґрунтується на результатах апіорного аналізу відомих робіт, що детально викладено в першому розділі дисертації. Додатково імпортовано `XGBRegressor` (з бібліотеки "`xgboost` ($\geq 3.0.4, < 4.0.0$)") [125]. Для оцінювання роботи ML моделей використано методи `cross_val_score`, `KFold` для крос-валідації, а також метрики R^2 , RMSE, MAE та MAPE.

Відбір ознак виконувався за допомогою спеціалізованої бібліотеки "`mafese` ($\geq 1.0.0, < 2.0.0$)" [126–130], а саме модулів: `Filter Selector`, `Sequential Selector`, `Recursive Selector`, `Mha Selector`. Крім цього, було застосовано `mutual_info_regression` із бібліотеки `sklearn.feature_selection`, що являє собою метод оцінки взаємної інформації, який дозволяє виявити нелінійні залежності між ознаками та цільовою змінною.

Із методів пояснювального штучного інтелекту (XAI), було використано два, які сумісні з версіями інших бібліотек, що були застосовані в цьому дослідженні, а саме: "`eli5` ($\geq 0.16.0, < 0.17.0$)" [131] – забезпечує глобальні та локальні пояснення моделей, зокрема через `Permutation Importance`; "`lime` ($\geq 0.2.0.1, < 0.3.0.0$)" [132, 133] – дозволяє створювати локальні інтерпретації прогнозів, пояснюючи внесок кожної ознаки для конкретного прикладу.

Також у роботі було використано низку технічних бібліотек для інтеграції та збереження: `joblib` [134] – для серіалізації та збереження навчених моделей; `json` [135] – для збереження параметрів попередньої обробки (наприклад, `StandardScaler`); `functools.reduce` [136] – для злиття декількох наборів даних; `IPython.display` ("`ipython` ($\geq 9.4.0, < 10.0.0$)") [137] – для інтеграції з середовищем `Jupyter Notebook`, зручного виведення таблиць та графіків.

Таким чином, набір використаних бібліотек дозволяє охопити всі необхідні етапи повноциклової трансформації даних агромоніторингу в межах єдиної

обчислювальної архітектури: від первинного збору та очищення даних, через побудову та оцінку моделей, до забезпечення прозорості результатів та інтерпретованості моделей. У додатку В.7 наведено зміст `pyproject.toml` файлу, що згенеровано Poetry [138], який використовується в якості базового інструменту для керування залежностями в Python-проектах.

3.2 Програмні компоненти збору та попередньої обробки даних

Під час розв’язання задачі прогнозування врожайності було використано публічно доступний набір даних «Crop Yield Prediction Dataset» [139] із урахуванням відомих програмних реалізацій [140, 141], які одержали подальшого розвитку в цій дисертаційній роботі (див. додаток В.1–В.7). Вихідними джерелами виступали кілька окремих файлів у форматі CSV: `yield.csv` містить інформацію про врожайність сільськогосподарських культур (показник продуктивності в одиницях виробленої продукції на гектар, hg/ha); `rainfall.csv` містить дані про середньорічну кількість опадів, виміряну в міліметрах; `pesticides.csv` містить інформацію про кількість використаних пестицидів у тоннах; `temp.csv` містить дані про середньорічну температуру в $^{\circ}C$. У свою чергу, дані про кількість пестицидів `pesticides.csv` та врожайність `yield.csv` згенеровано на підставі офіційних даних FAO [9]. Дані про кількість опадів `rainfall.csv` та середню температуру `temp.csv` одержано з джерела World Bank Group [142].

Отже, після апріорного аналізу масивів вхідних даних було встановлено, що першим етапом дослідження є формування єдиного узгодженого набору даних, на основі якого здійснюється подальший аналіз та побудова моделей прогнозування врожайності агрокультур.

Первинна структура файлу `yield.csv` окрім інформативних показників, також включала метадані (код країни, одиниці вимірювання та інше), які не були необхідними під час побудови прогнозних моделей, як це показано у вигляді фрагмента в таблиці 3.1.

Таблиця 3.1 – Огляд атрибутів у наборі даних yield.csv

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value
0	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1961	1961	hg/ha	14000
1	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1962	1962	hg/ha	14000
2	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1963	1963	hg/ha	14260
3	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1964	1964	hg/ha	14257
4	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1965	1965	hg/ha	14400
...												
56712	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2012	2012	hg/ha	24420
56713	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2013	2013	hg/ha	22888
56714	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2014	2014	hg/ha	21357
56715	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2015	2015	hg/ha	19826
56716	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2016	2016	hg/ha	18294

На підставі апріорного аналізу таблиці 3.1 було збережено такі атрибути: area (країна), item (культура), year (рік), hg/ha_yield (перейменований стовпець "Value"); видалено такі атрибути: службові коди та дублюючі стовпці. Також було видалено рядки з пропущеними або некоректними значеннями. У результаті отримано компактну таблицю даних, що описує базову залежність врожайності агрокультур від географічних та часових факторів, як фрагментарно показано в таблиці 3.2.

Таблиця 3.2 – Фрагмент результуючої таблиці даних df_yield на основі yield.csv

	area	item	year	value_hg_ha
0	Afghanistan	Maize	1961	14000
1	Afghanistan	Maize	1962	14000
2	Afghanistan	Maize	1963	14260
3	Afghanistan	Maize	1964	14257
4	Afghanistan	Maize	1965	14400
...				

Коротка статистична зведена інформація по числових стовпцях df_yield DataFrame, як результат виконання методу .describe(), наведена в таблиці 3.3, де фігурують такі характеристики: count – кількість непорожніх значень; mean –

середнє арифметичне; std – стандартне відхилення; min – мінімальне значення; 25 % – перший кuartиль (Q1); 50 % – медіана (Q2); 75 % – третій кuartиль (Q3); max – максимальне значення.

Таблиця 3.3 – Статистична зведена інформація по числових стовпцях df_yield

	year	value hg ha
count	56717	56717
mean	1989,67	62094,66
std	16,13	67835,93
min	1961	0,0
25%	1976	15680
50%	1991	36744
75%	2004	86213
max	2016	1000000

Нижче наведено результат виконання в pandas методу .info() для отримання стислої інформації про df_yield DataFrame, а саме про назву класу та кількість рядків і стовпців з їхніми назвами, кількість непорожніх (non-null) значень, тип даних (dtype) та використання пам'яті (скільки RAM займає DataFrame).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56717 entries, 0 to 56716
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   area            56717 non-null  object
1   item            56717 non-null  object
2   year            56717 non-null  int64
3   value_hg_ha     56717 non-null  int64
dtypes: int64(2), object(2)
memory usage: 1.7+ MB
```

Набір даних про опади (rainfall.csv) містив середньорічні значення кількості опадів у різних країнах. Первинно дані були структуровані, як у фрагментарному вигляді показано в таблиці 3.4.

Таблиця 3.4 – Огляд атрибутів у наборі даних rainfall.csv

	Area	Year	average rain fall mm per year
0	Afghanistan	1985	327
1	Afghanistan	1986	327
2	Afghanistan	1987	327
3	Afghanistan	1989	327
4	Afghanistan	1990	327
...			
6723	Zimbabwe	2014	657
6724	Zimbabwe	2015	657
6725	Zimbabwe	2016	657
6726	Zimbabwe	2017	657

В цій таблиці було збережено всі наявні атрибути, а саме: Area, Year, average_rain_fall_mm_per_year. Стовпці Area та Year було стандартизовано як area, year. Рядки, що мали пропуски в даних, було видалено. Таким чином, сформовано таблицю даних df_rain із часовими рядами кількості опадів для кожної країни, як у вигляді відповідного фрагмента показано в таблиці 3.5.

Таблиця 3.5 – Фрагмент таблиці даних df_rain на основі rainfall.csv

	area	year	average rain fall mm per year
0	Afghanistan	1985	327,0
1	Afghanistan	1986	327,0
2	Afghanistan	1987	327,0
3	Afghanistan	1989	327,0
...			

Коротка статистична зведена інформація по числових стовпцях df_rain DataFrame, як результат виконання методу .describe(), наведена у таблиці 3.6.

Таблиця 3.6 – Статистична зведена інформація по числових стовпцях df_rain

	year	average rain fall mm per year
count	5947	5947
mean	2001,37	1124,74
std	9,53	786,26
min	1985	51,0
25%	1993	534,0
50%	2001	1010,0
75%	2010	1651,0
max	2017	3240,0

Нижче наведено результат виконання в pandas методу .info() для отримання стислої інформації про df_rain DataFrame, аналогічно df_yield.

```
<class 'pandas.core.frame.DataFrame'>
Index: 5947 entries, 0 to 5946
Data columns (total 3 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   area                                5947 non-null   object
1   year                                5947 non-null   int64
2   average_rain_fall_mm_per_year      5947 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 185.8+ KB
```

Набір даних про використання пестицидів (pesticides.csv) містив інформацію щодо обсягів використання пестицидів у тоннах. Первинна структура також включала метадані, що підлягали видаленню. Фрагмент початкового файлу pesticides.csv наведено в таблиці 3.7.

Таблиця 3.7 – Огляд атрибутів у наборі даних pesticides.csv

	Domain	Area	Element	Item	Year	Unit	Value
0	Pesticides Use	Albania	Use	Pesticides (total)	1990	tonnes of active ingredients	121,0
1	Pesticides Use	Albania	Use	Pesticides (total)	1991	tonnes of active ingredients	121,0
2	Pesticides Use	Albania	Use	Pesticides (total)	1992	tonnes of active ingredients	121,0
3	Pesticides Use	Albania	Use	Pesticides (total)	1993	tonnes of active ingredients	121,0
4	Pesticides Use	Albania	Use	Pesticides (total)	1994	tonnes of active ingredients	201,0
...							
4344	Pesticides Use	Zimbabwe	Use	Pesticides (total)	2012	tonnes of active ingredients	3375,53
4345	Pesticides Use	Zimbabwe	Use	Pesticides (total)	2013	tonnes of active ingredients	2550,07
4346	Pesticides Use	Zimbabwe	Use	Pesticides (total)	2014	tonnes of active ingredients	2185,07
4347	Pesticides Use	Zimbabwe	Use	Pesticides (total)	2015	tonnes of active ingredients	2185,07
4348	Pesticides Use	Zimbabwe	Use	Pesticides (total)	2016	tonnes of active ingredients	2185,07

В цьому наборі даних було збережено такі атрибути: Area, Year, Value.

Числовий показник Value було приведено до уніфікованої назви pesticides_tonnes. Area та Year перейменовано на area та year, відповідно. Також було вилучено записи з некоректними або відсутніми значеннями. У результаті отримано таблицю, що відображає агротехнічні характеристики для кожної країни в часовій динаміці, як фрагментарно показано в таблиці 3.8.

Таблиця 3.8 – Фрагмент результуючої таблиці даних df_pesticides на основі pesticides.csv

	area	year	pesticides tonnes
0	Albania	1990	121,0
1	Albania	1991	121,0
2	Albania	1992	121,0
3	Albania	1993	121,0
4	Albania	1994	201,0
...			

Коротка статистична зведена інформація по числових стовпцях df_pesticides DataFrame, як результат виконання методу .describe(), наведена нижче в таблиці 3.9.

Таблиця 3.9 – Статистична зведена інформація по числових стовпцях df_pesticides

	year	pesticides tonnes
count	4349	4350,0
mean	2003,14	20300,0
std	7,73	118000,0
min	1990	0,0
25%	1996	93,0
50%	2003	1140,0
75%	2010	7870,0
max	2016	1810000,0

Нижче наведено результат виконання в pandas методу .info() для отримання стислої інформації про df_pesticides DataFrame.

```
<class 'pandas.core.frame.DataFrame'>
Index: 4349 entries, 0 to 4348
```

```

Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   area                   4349 non-null   object
1   year                   4349 non-null   int64
2   pesticides_tonnes      4349 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 135.9+ KB

```

Дані temp.csv включали кліматичні дані щодо середньорічної температури повітря. Його структура була найбільш простою, як у вигляді відповідного фрагмента показано в таблиці 3.10.

Таблиця 3.10 – Огляд атрибутів у наборі даних temp.csv

	year	country	avg_temp
0	1849	Côte D'Ivoire	25,58
1	1850	Côte D'Ivoire	25,52
2	1851	Côte D'Ivoire	25,67
3	1852	Côte D'Ivoire	NaN
4	1853	Côte D'Ivoire	NaN
...			

Всі наявні атрибути (стовпці) було збережено. Стовпець country було перейменовано на area, для приведення у відповідність іншим таблицям даних. Також було виконано видалення рядків із відсутніми значеннями. Таким чином, фрагмент результуючого набору даних df_avg_temperature DataFrame наведено у таблиці 3.11.

Таблиця 3.11 – Фрагмент результуючої таблиці даних df_avg_temperature на основі temp.csv

	year	area	avg_temp
0	1849	Côte D'Ivoire	25,58
1	1850	Côte D'Ivoire	25,52
2	1851	Côte D'Ivoire	25,67
7	1856	Côte D'Ivoire	26,28
8	1857	Côte D'Ivoire	25,17
...			

Коротка статистична зведена інформація по числових стовпцях `df_avg_temperature` DataFrame, як результат виконання методу `.describe()`, наведена у таблиці 3.12.

Таблиця 3.12 – Статистична зведена інформація по числових стовпцях `df_avg_temperature`

	year	avg_temp
count	68764	68764
mean	1908,74	16,18
std	66,06	7,59
min	1743	-14,35
25%	1862	9,75
50%	1914	16,14
75%	1964	23,76
max	2013	30,73

Нижче наведено результат виконання в `pandas` методу `.info()` для отримання стислої інформації про `df_avg_temperature` DataFrame.

```
<class 'pandas.core.frame.DataFrame'>
Index: 68764 entries, 0 to 68763
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   year        68764 non-null  int64
1   area        68764 non-null  object
2   avg_temp    68764 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 2.1+ MB
```

Формування фінального набору даних `df_yield_merged` передбачає, що всі чотири оброблені таблиці, що описані вище, буде інтегровано в єдину таблицю. Об'єднання виконувалося за інформативними змінними `area`, `item` та `year` (для `df_yield`) та `area`, `year` (для решти: `df_rain`, `df_pesticides` та `df_avg_temperature`). Технічно це реалізовано за допомогою послідовних операцій `pd.merge()`, згрупованих через `functools.reduce` (див. додаток В.2). Фінальна структура одержаного набору даних, який використано в якості базового в подальших

дослідженнях, наведена у фрагментарному вигляді в таблиці 3.13.

Таблиця 3.13 – Фрагмент результуючої таблиці даних `df_yield_merged`

	area	year	average_rain_ fall mm per year	pesticides tonnes	avg_tem p	item	value_hg_ha
0	Albania	1990	1485,0	121,0	16,37	Maize	36613
1	Albania	1990	1485,0	121,0	16,37	Potatoes	66667
2	Albania	1990	1485,0	121,0	16,37	Rice, paddy	23333
3	Albania	1990	1485,0	121,0	16,37	Sorghum	12500
4	Albania	1990	1485,0	121,0	16,37	Soybeans	7000
...							

Коротка статистична зведена інформація по числових стовпцях `df_yield_merged`, як результат виконання методу `.describe()`, наведена в таблиці 3.14.

Таблиця 3.14 – Статистична зведена інформація по числових стовпцях `df_yield_merged`

	average_rain_fall_ mm per year	pesticides_tonnes	avg_temp	value_hg_ha
count	25932	25932	25932	25932
mean	1146,84	34782,57	20,42	76978,13
std	717,39	57228,57	6,44	84647,61
min	51,0	0,04	1,3	50,0
25%	593,0	1597,0	16,58	19996,0
50%	1083,0	15373,0	21,42	39530,0
75%	1668,0	45620,0	26,06	104168,0
max	3240,0	367778,0	30,65	501412,0

У таблиці 3.15 наведено статистичну інформацію, де за допомогою методу `.groupby('item')` згруповано всі рядки `df_yield_merged` DataFrame за унікальними значеннями в колонці `item` (за типом с/г культур) та за допомогою методу `.count()` підраховано кількість непорожніх значень (non-null) у кожній колонці окремо для кожної групи. Також нижче таблиці 3.15 наведено результат виконання в pandas методу `.info()` для отримання стислої зведеної інформації про `df_yield_merged` DataFrame.

Таблиця 3.15 – Статистична зведена інформація по с/г культурам
df_yield_merged

item	area	year	average_rain_fall_ mm_per_year	pesticides_ tonnes	avg_tem p	value_hg_ha
Cassava	1889	1889	1889	1889	1889	1889
Maize	3824	3824	3824	3824	3824	3824
Plantains and others	556	556	556	556	556	556
Potatoes	3956	3956	3956	3956	3956	3956
Rice, paddy	3091	3091	3091	3091	3091	3091
Sorghum	2770	2770	2770	2770	2770	2770
Soybeans	2940	2940	2940	2940	2940	2940
Sweet potatoes	2593	2593	2593	2593	2593	2593
Wheat	3539	3539	3539	3539	3539	3539
Yams	774	774	774	774	774	774

Dataframe columns and types:

```
<class 'pandas.core.frame.DataFrame'>
```

Index: 25932 entries, 0 to 25931

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	area	25932 non-null	object
1	year	25932 non-null	int64
2	average_rain_fall_mm_per_year	25932 non-null	float64
3	pesticides_tonnes	25932 non-null	float64
4	avg_temp	25932 non-null	float64
5	item	25932 non-null	object
6	value_hg_ha	25932 non-null	int64

dtypes: float64(3), int64(2), object(2)

memory usage: 1.6+ MB

Таким чином, у результаті реалізації етапу збору та попередньої обробки одержано структурований набір даних, який містить як агротехнічні показники (врожайність і обсяги пестицидів), так і кліматичні фактори (кількість опадів і температура повітря). Кожен рядок відповідає конкретному запису про врожайність певної культури у певній країні та певному році. Всі дані представлені в стандартизованій формі, що дозволяє їх безпосередньо використовувати під час побудови предикативних моделей на основі алгоритмів машинного навчання. Завдяки попередньому очищенню усунуто дублікати,

пропуски та непослідовності. Отриманий набір даних охоплює значну кількість країн, агрокультур та часових періодів, що дає можливість виявити як локальні, так і глобальні закономірності [117].

Також варто зазначити, що реалізація підходів до попереднього статистичного аналізу агрокліматичних даних дозволила отримати результати, що корелюють із раніше відомими [140, 141], що підтверджує об'єктивність розроблених програмних компонент. Проте в отриманих результатах цієї дисертаційної роботи порівняно з тими, що наведені в [140, 141], спостерігається певна відмінність у проміжних статистичних оцінках (типи даних, використана пам'ять та виявлені непорожні значення) і статистично зведених показниках, що обумовлено специфікою відповідної програмної реалізації (див. додаток В.2).

Дослідження цього підрозділу дозволили отримати єдину структуровану таблицю даних [143], що виступає базовим джерелом для подальшого синтезу програмно-алгоритмічного забезпечення прогнозування врожайності.

3.3 Програмні компоненти розвідувального аналізу даних

Після формування єдиного підготовленого набору даних було проведено розвідувальний аналіз даних (EDA), метою якого є виявлення закономірностей, залежностей та аномалій у даних, що можуть впливати на побудову предикативних моделей на основі ML алгоритмів (див. додаток В.3) [117].

Згідно з результатами попереднього аналізу `df_yield_merged.info()`, які наведено вище, кількість записів у підготовленому наборі даних складається з 25932 записів, починаючи з індексу 0 і закінчуючи індексом 25931. Набір даних містить 7 стовпців. Більшість стовпців (5 з 7) мають тип даних `int64` та `float64`. Тільки стовпці `item` та `area` мають тип даних `object`. У наборі даних `df_yield_merged`, немає відсутніх значень, оскільки кожен стовпець має 25932 записи, що не є порожніми (`non-null`).

Для числових ознак (`year`, `average_rain_fall_mm_per_year`, `pesticides_tonnes`, `avg_temp`, `value_hg_ha`) було отримано базові характеристики статистичного

розподілу (див. табл. 3.14), що дозволило встановити наступне: `average_rain_fall_mm_per_year`, середній рівень опадів на рік становить приблизно 1147 мм, при цьому мінімальний рівень опадів становить 51 мм, а максимальний – 3240 мм; `pesticides_tonnes`, середній обсяг використаних пестицидів становить 34783 тонни, при цьому мінімальний обсяг становить 0,04 тонни, а максимальний – 367778 тонн, тобто варіативність є досить значною від майже нульових значень у деяких країнах до значних обсягів у промислово розвинених регіонах; `value_hg_ha`, середня врожайність становить 76978 hg/ha, діапазон коливається від 50 до 501412 hg/ha.

У таблиці 3.16 наведено описову статистику тільки для категоріальних колонок у `DataFrame`, з такими рядками: `count` – кількість непорожніх (non-null) значень; `unique` – кількість унікальних значень; `top` – значення, яке зустрічається найчастіше; `freq` – кількість появ значення, що найчастіше зустрічається.

Таблиця 3.16 – Статистична зведена інформація по категоріальним даним `df_yield_merged`

	area	item
count	25932	25932
unique	101	10
top	India	Potatoes
freq	3384	3956

Аналізуючи дані, які наведено в таблиці 3.16, можна зробити висновок, що є 101 унікальна країна в наборі даних `df_yield_merged`, причому Індія є найчастішою категорією, яка зустрічається 3384 рази із 25932 записів. Крім того, є 10 унікальних видів сільськогосподарських культур, де найпоширенішим видом є картопля, яка зустрічається 3956 разів із 25932 записів.

Під час розвідувального аналізу даних було побудовано матрицю кореляцій Пірсона між числовими змінними в підсумковому наборі даних `df_yield_merged`, як показано на рисунку 3.2. У процесі її побудови було використано маску верхнього трикутника, тому відображено лише нижню частину.

Як видно з аналізу рисунка 3.2, всі коефіцієнти кореляції знаходяться в

межах приблизно від $-0,11$ до $+0,33$. Це вказує, що між цими змінними немає сильних лінійних залежностей. Єдина помітна кореляція спостерігається між середньою температурою (`avg_temp`) та середньорічними опадами (`average_rain_fall_mm_per_year`): коефіцієнт дорівнює $+0,334$, що вказує на наявність помірного позитивного зв'язку. Це може означати, що в досліджуваній вибірці роки з вищими середніми температурами супроводжуються також більшою кількістю опадів. Вплив пестицидів (`pesticides_tonnes`) на врожайність (`value_hg_ha`) практично відсутній: коефіцієнт кореляції складає всього $+0,066$. Це свідчить, що лінійна залежність між цими змінними слабка або незначна та може вказувати, що вплив пестицидів виявляється у нелінійних або відкладених ефектах. Тим не менш, слабкий позитивний зв'язок вказує, що збільшення об'єму пестицидів призводить до збільшення врожайності.

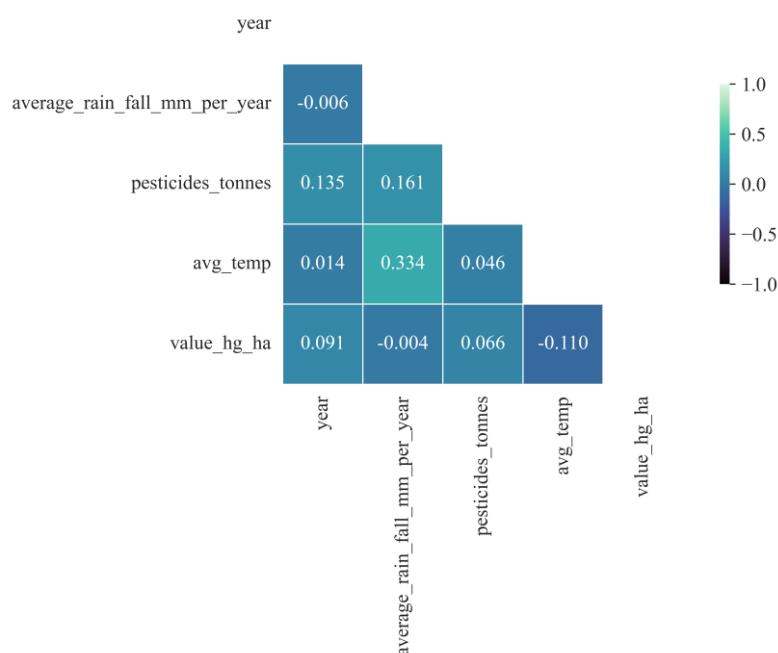


Рисунок 3.2 – Матриця кореляцій Пірсона між числовими змінними в `df_yield_merged`

Коефіцієнт між `average_rain_fall_mm_per_year` та `value_hg_ha` складає $-0,004$, тобто зв'язок дуже слабкий, що є доволі аномальним, адже зазвичай опади мають вагоме значення для забезпечення врожайності. Такий ефект може бути

обумовлено тим, що в наборі даних не враховано фазу вегетації чи деякі регіональні відмінності. Між температурою (`avg_temp`) та врожайністю (`value_hg_ha`) спостерігається слабкий від'ємний зв'язок, коефіцієнт дорівнює $-0,110$. Це може вказувати на те, що зростання середньої температури трохи знижує врожайність, хоча зв'язок слабкий. Часовий фактор (`year`) не має суттєвої кореляції з іншими змінними, але простежується тенденція у даних по роках щодо збільшення `pesticides_tonnes`, `avg_temp` та `value_hg_ha`.

Наступним етапом була побудова графіків-гістограм для всіх числових колонок у DataFrame `df_yield_merged`, як показано на рисунку 3.3.

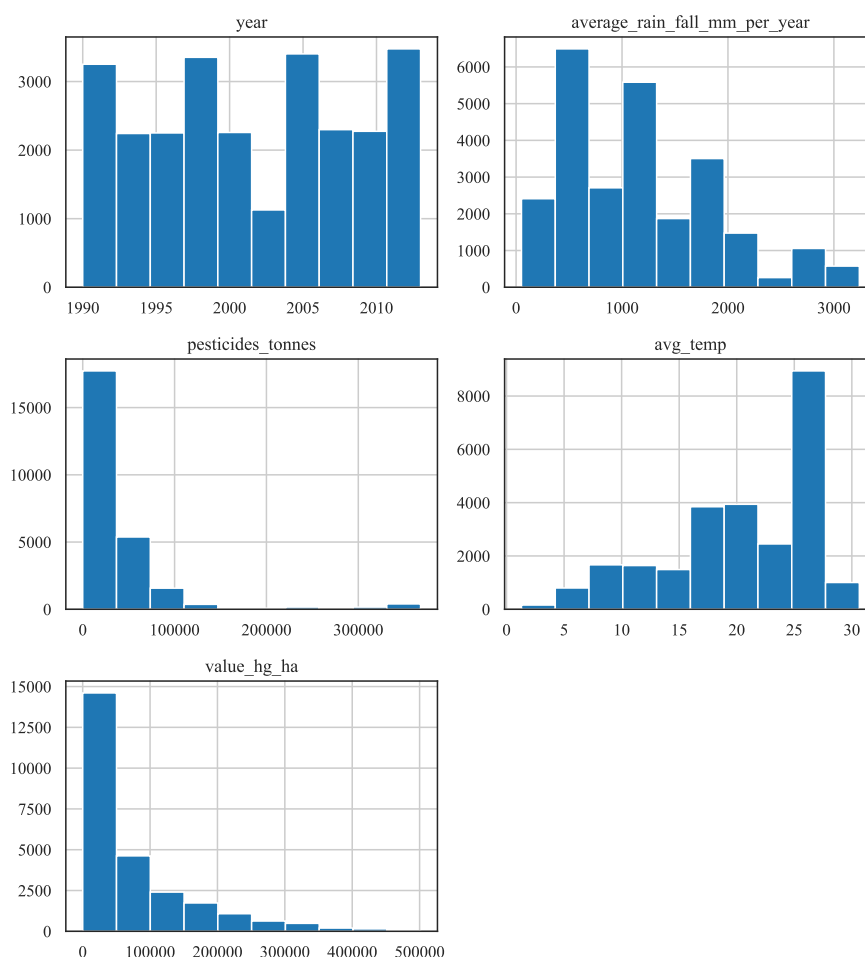


Рисунок 3.3 – Гістограми для всіх числових колонок у DataFrame `df_yield_merged`

Із рисунку 3.3 видно, що розподіл даних по роках відносно рівномірний, хоча спостерігаються періоди з більшою кількістю записів (наприклад, після 2010

року). Це може свідчити про збільшення кількості спостережень або кращу доступність даних у пізніші роки. Розподіл даних по `average_rain_fall_mm_per_year` асиметричний. Більшість спостережень припадає на діапазон до ~1500 мм опадів на рік. Значення понад 2500 мм зустрічаються доволі рідко, що вказує на їх винятковість (можливі екстремальні кліматичні умови чи специфічні регіони). Щодо показника `pesticides_tonnes`, то спостерігається дуже значна асиметрія. Основна маса даних зосереджена при низьких значеннях (<50000 тонн), тоді як поодинокі екстремальні значення сягають понад 300000 тонн, що свідчить про наявність кількох країн чи років із надзвичайно високим використанням пестицидів, які виступають статистичними викидами. Розподіл `avg_temp` має форму, близьку до нормальної, але з незначною асиметрією. Найбільша концентрація даних припадає на діапазон від 18 °C до 27 °C. Це характерно для більшості країн із теплим кліматом, де активно ведеться землеробство. Дані `value_hg_ha` мають яскраво виражений асиметричний розподіл. Більшість значень знаходиться нижче 100000 hg/ha, але зустрічаються і надзвичайно великі (до 500000 hg/ha). Це може бути результатом різниць між культурами, регіонами, площами або методами вимірювання.

Таким чином, аналіз розподілів показників, які наведено на рисунку 3.3, дозволив встановити, що більшість досліджуваних змінних характеризується суттєвою асиметрією. Так, показники використання пестицидів та врожайності мають виражені викиди, що відображає нерівномірність агротехнологічних практик між країнами та культурами. Кількість опадів також варіюється в широкому діапазоні, однак основна маса спостережень припадає на значення нижче 1500 мм/рік. Єдиною змінною з відносно симетричним розподілом є середньорічна температура, з концентрацією даних у діапазоні від 18 °C до 27 °C. Отримані результати спостережень свідчать про необхідність застосування методів аналізу даних, що є стійкими до викидів, а також можливого перетворення змінних для коректного подальшого статистичного та інтелектуального аналізу.

Для більш детального аналізу також було побудовано сімейство характеристик (за видами агрокультур), що відображає динаміку середньої врожайності за роками, як показано на рисунку 3.4.

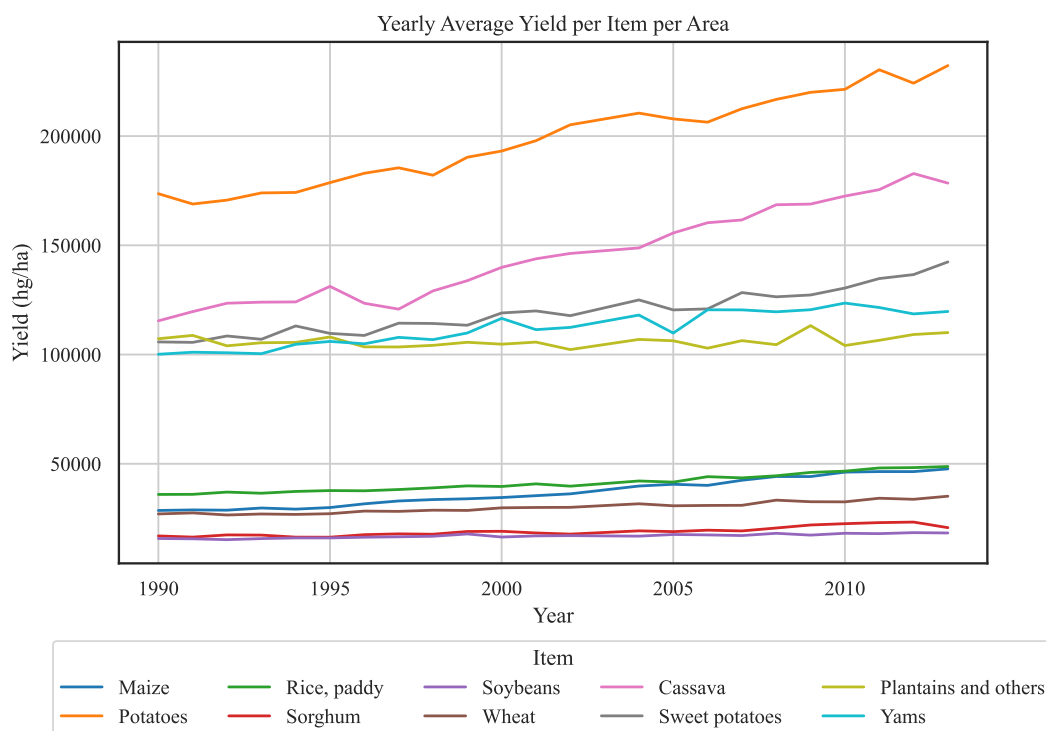


Рисунок 3.4 – Графік сімейства характеристик середньої врожайності за роками для всіх аналізованих видів агрокультур

Для більш глибокого аналізу даних щодо впливу агротехнічних і ґрунтокліматичних факторів на врожайність с/г культур було побудовано матрицю попарних діаграм розсіювання для всіх числових змінних у наборі даних `df_yield_merged`, з групуванням за ознакою `item` (с/г культура), як показано на рисунку 3.5.

На підставі аналізу залежностей, які наведено на рисунках 3.4 і 3.5, можна зробити такі проміжні висновки: для вирощування ямсу використовується значна кількість пестицидів, кількість опадів практично не впливає на продуктивність його вирощування. Пшениця добре зростає в широкому діапазоні середніх температур і опадів за умови застосування пестицидів. Показник обсягів використовуваних пестицидів має сильну праву асиметрію – для багатьох культур

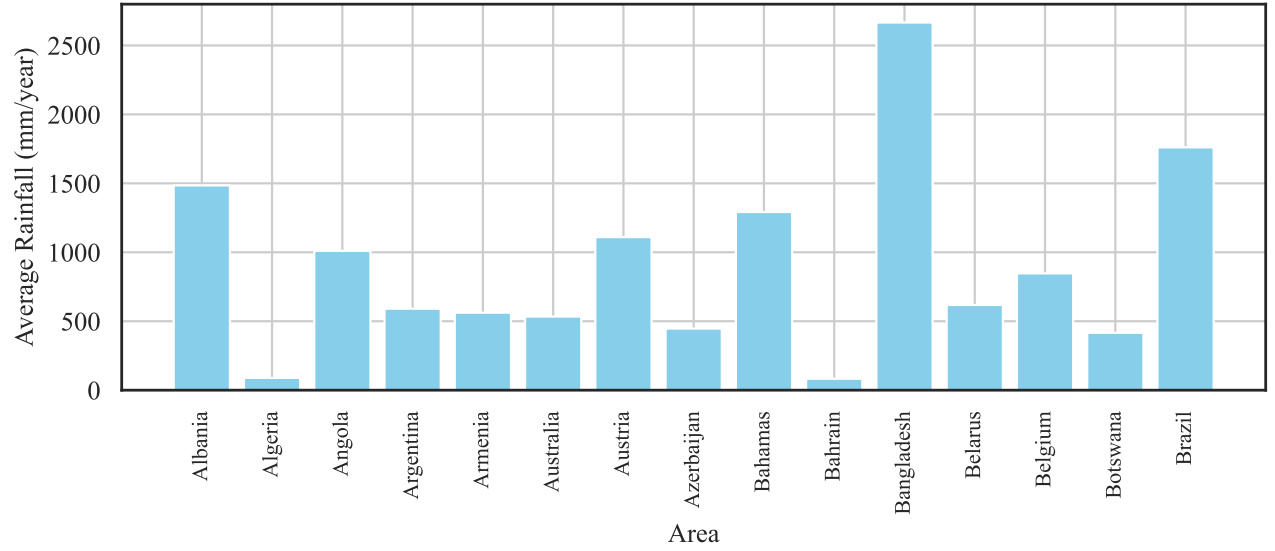
обсяги малі, але окремі культури (наприклад, Cassava, Yams, Soybeans) мають дуже високі значення. У період із 2003 по 2013 рік для підвищення врожайності ямсу використовували все більше пестицидів. Найнижчу врожайність мають сорго та соя, а найпопулярнішою культурою є картопля. Як зазначалося вище, загальної сильної кореляції між показниками обсягів використовуваних пестицидів і врожайності с/г культур не спостерігається, але для окремих культур урожайність зростає зі збільшенням використання пестицидів.



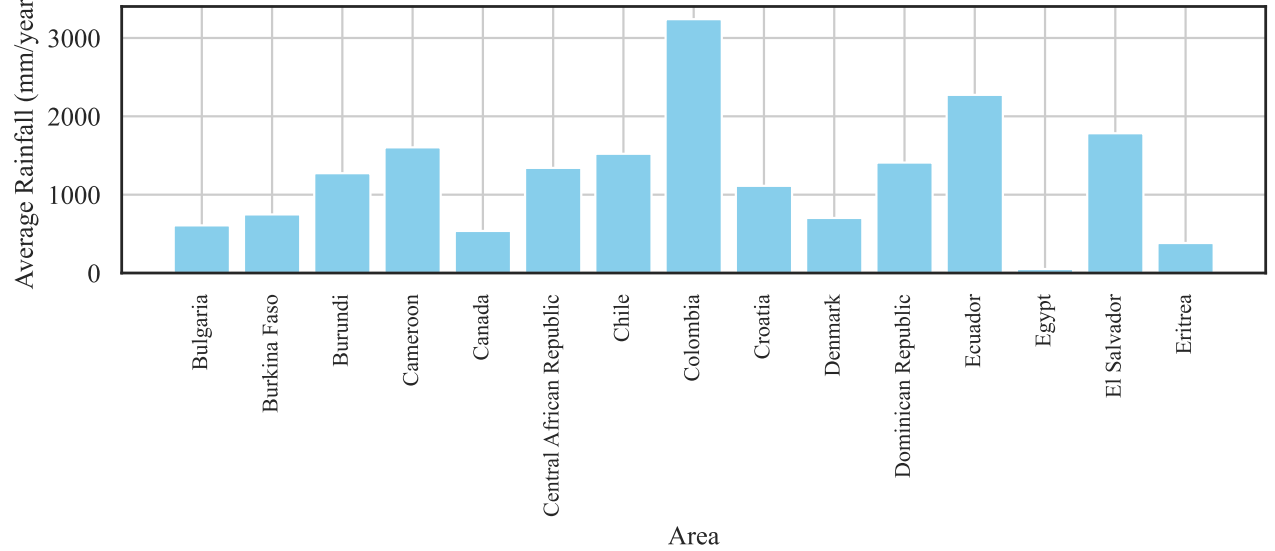
Рисунок 3.5 – Попарні діаграм розсіювання для всіх числових змінних у `df_yield_merged` DataFrame із групуванням за ознакою `item`

З метою аналізу розподілу ґрунтокліматичних показників було побудовано сукупність стовпчикових графіків, які показують середній річний рівень опадів для різних територій (`area`), як показано на рисунку 3.6.

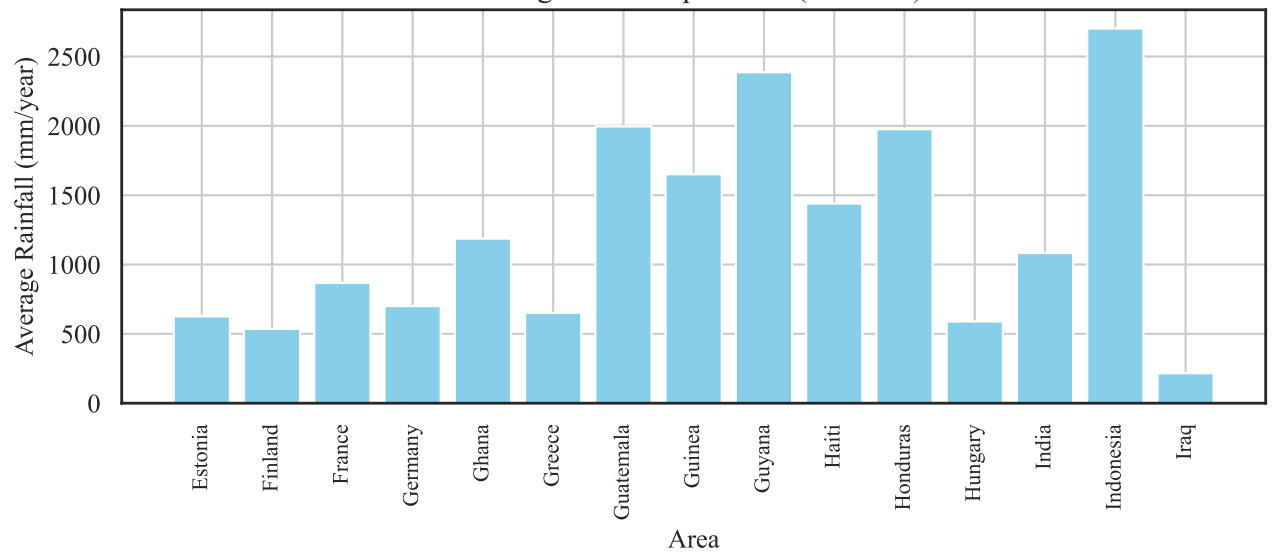
Average Rainfall per Area (Chunk 1)

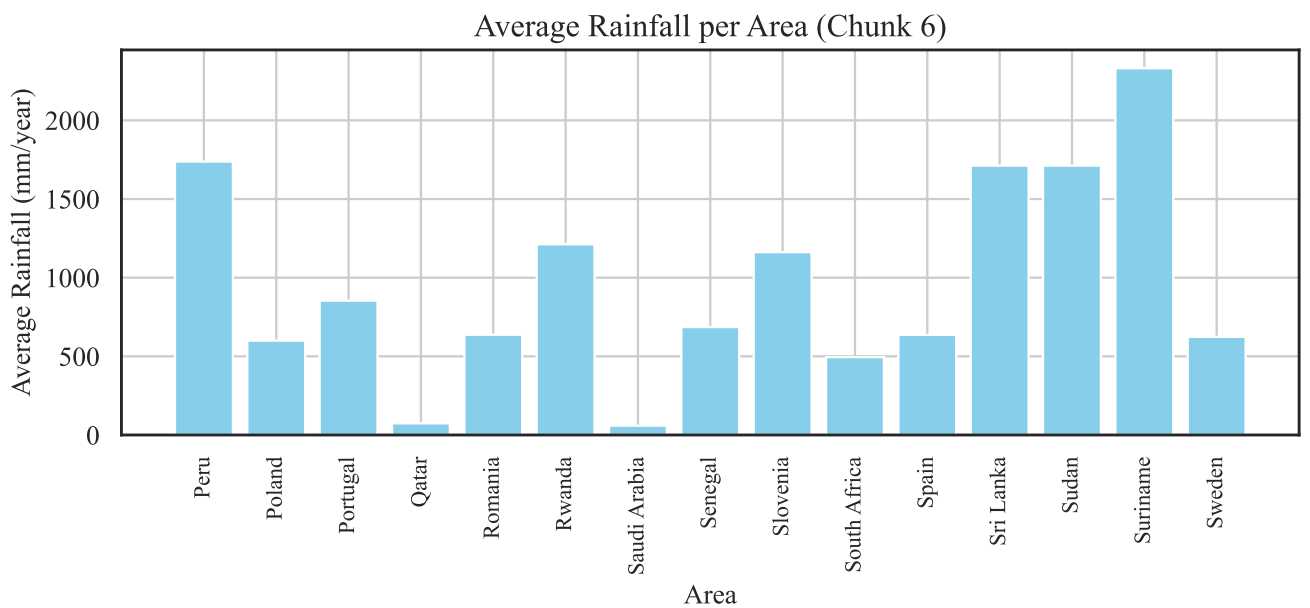
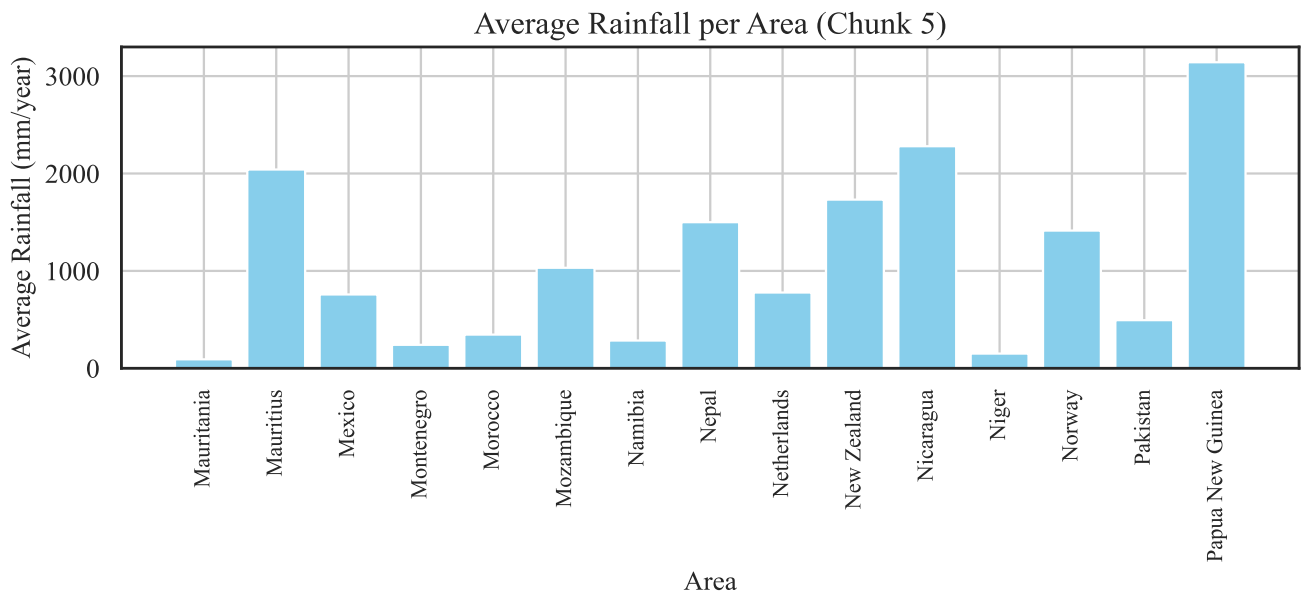
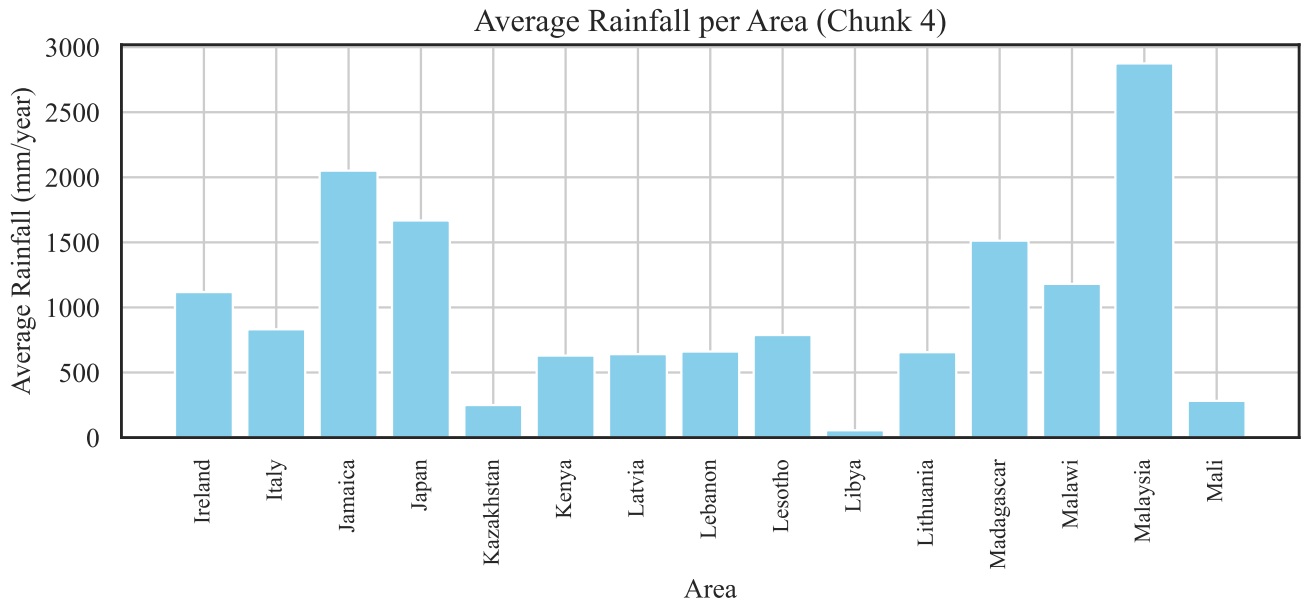


Average Rainfall per Area (Chunk 2)



Average Rainfall per Area (Chunk 3)





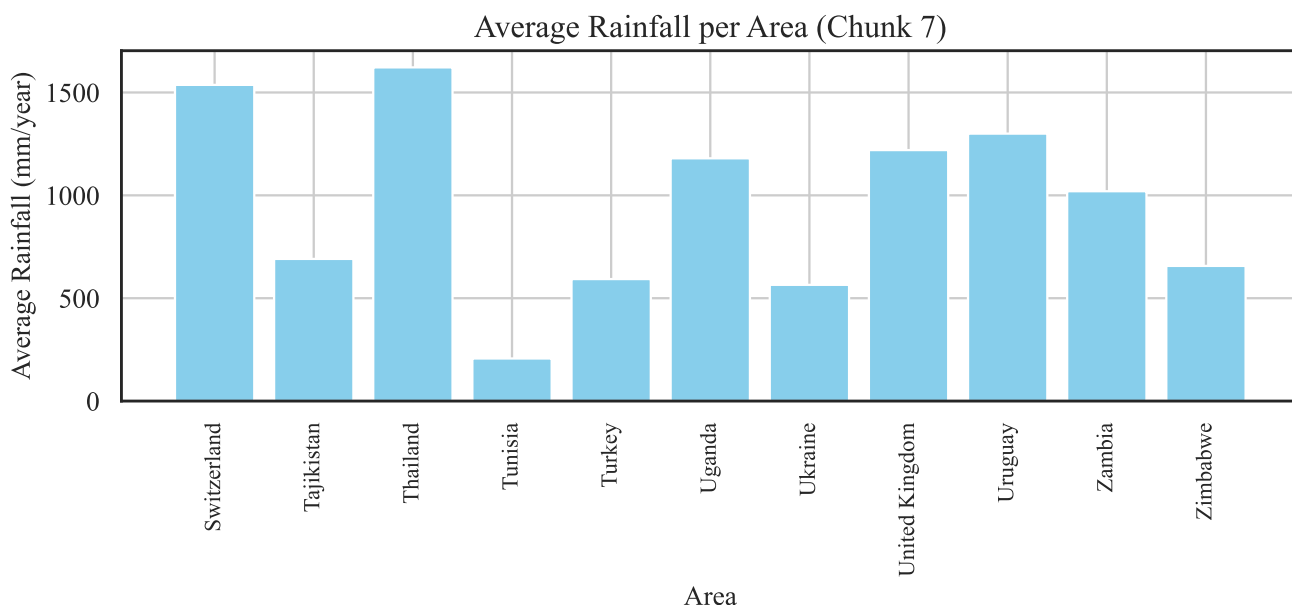


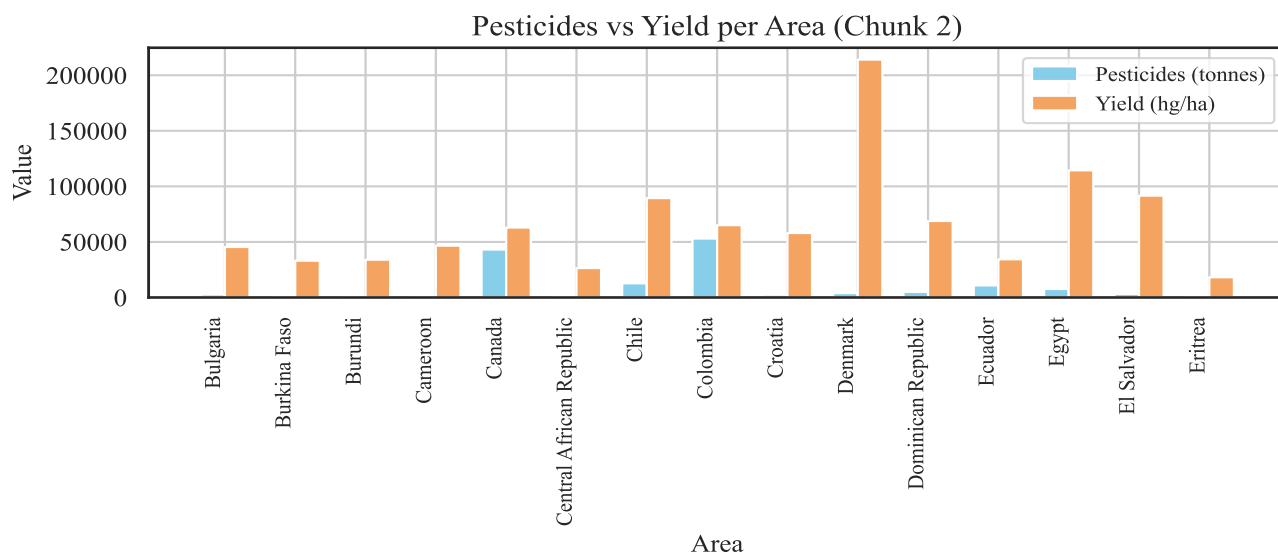
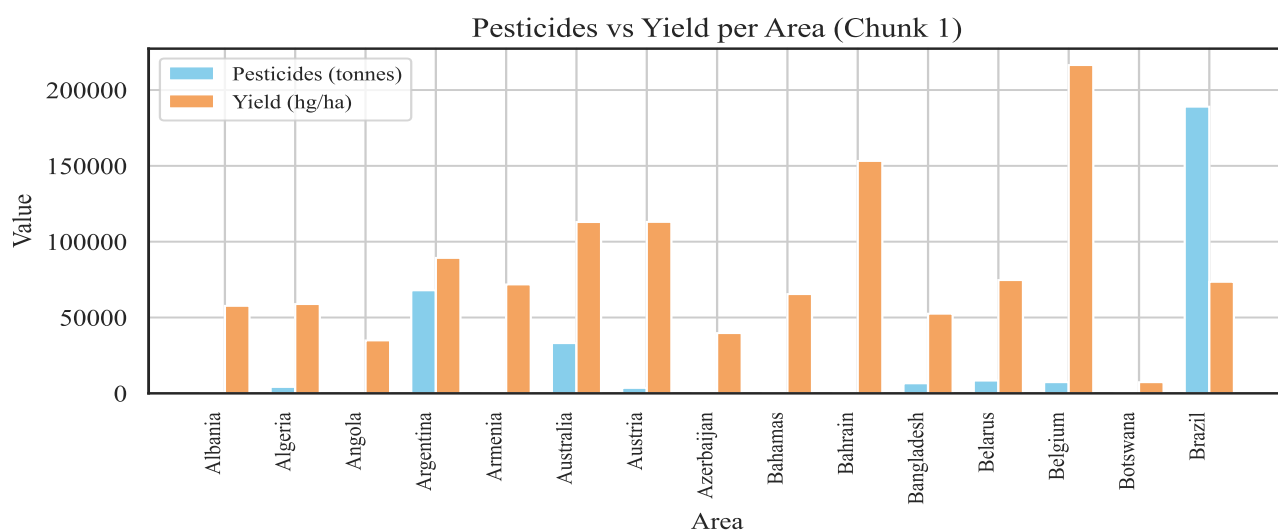
Рисунок 3.6 – Графіки залежності середнього річного рівня опадів для різних країн із `df_yield_merged` DataFrame

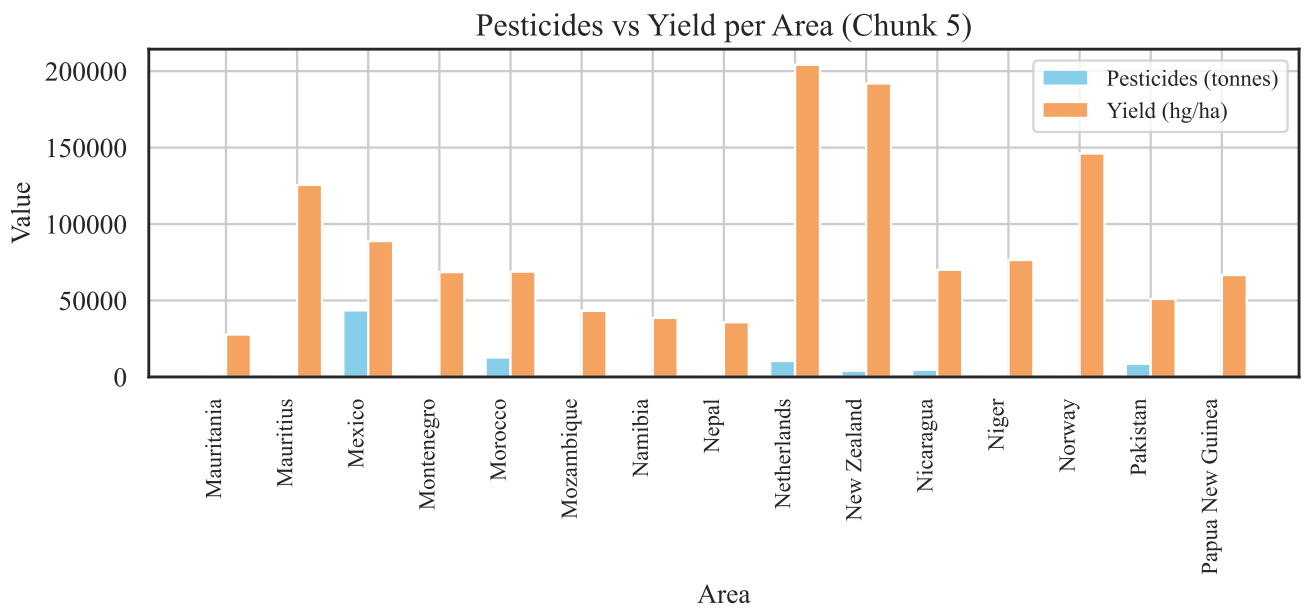
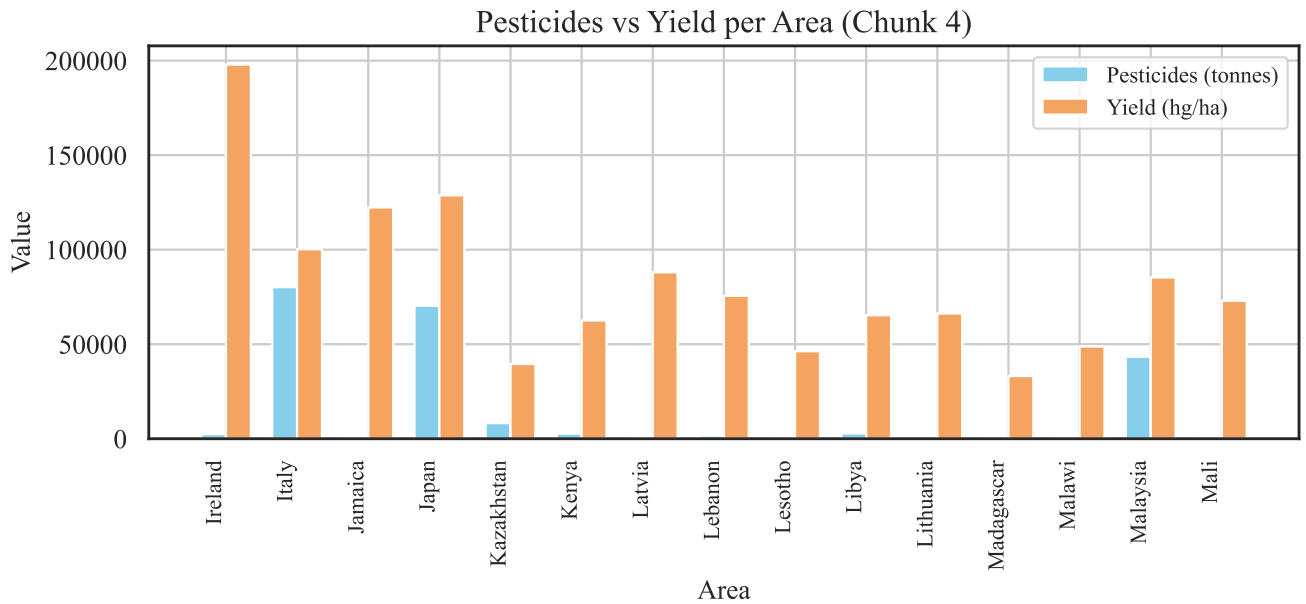
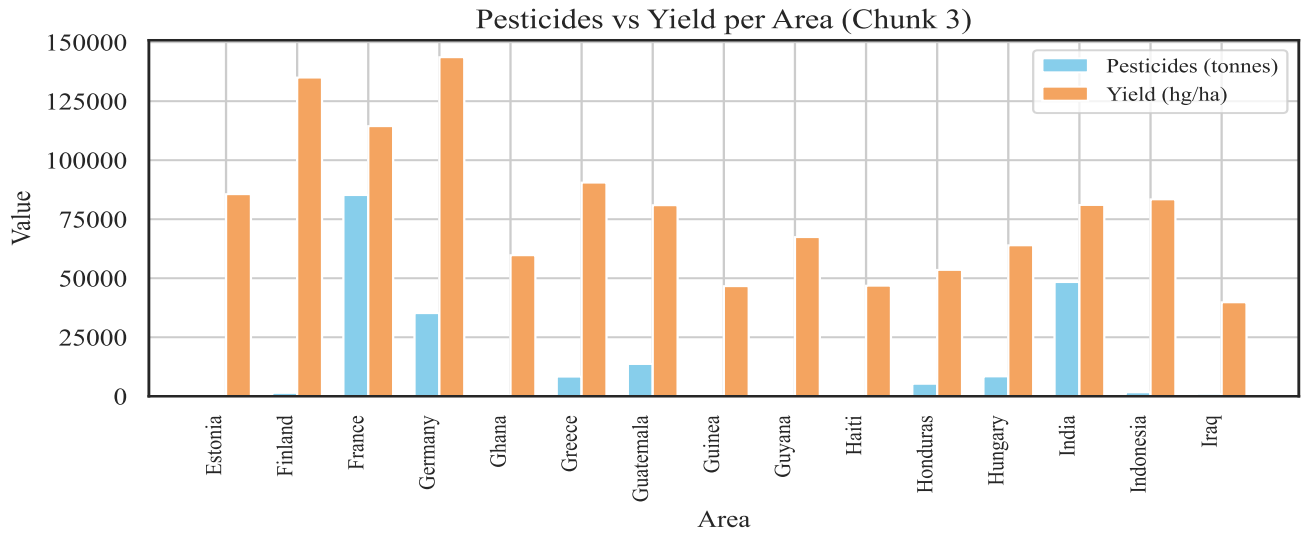
На підставі аналізу рисунку 3.6 встановлено, що спостерігається значна динаміка у кількості опадів у різних країнах, так у аналізованому наборі даних є 13 країн (Бангладеш, Гайана, Гватемала, Гондурас, Еквадор, Індонезія, Колумбія, Маврикій, Малайзія, Нікарагуа, Папуа-Нова Гвінея, Суринам і Ямайка) із показником кількості опадів, що дорівнює або перевищує 2000 мм, проте є також 20 країн із показником менше 500 мм (Азербайджан, Алжир, Бахрейн, Ботсвана, Еритрея, Єгипет, Ірак, Казахстан, Канада, Катар, Лівія, Мавританія, Малі, Намібія, Нігер, Пакистан, Південна Африканська Республіка, Саудівська Аравія, Туніс, Чорногорія).

Додатково було побудовано груповані стовпчикові діаграми, які порівнюють середні значення використання пестицидів і врожайності для різних країн, як показано на рисунку 3.7.

Графічний аналіз значень щодо використання пестицидів і врожайності у різних країнах (див. рис. 3.7) дозволяє зробити наступні висновки. У світі спостерігається значна диференціація країн за співвідношенням врожайність/обсяги пестицидів. Найбільші обсяги застосування пестицидів зафіксовані в Аргентині, Бразилії, Франції, Італії, Індії та Японії. Зокрема такі

країни як Франція, Італія та Японія, що характеризуються високим рівнем врожайності, використовують понад 75 тис. тонн пестицидів на рік, що може мати потенційно негативні наслідки для агроєкосистем та довкілля. Найнижчі показники застосування пестицидів (менше 10 тис. тонн) спостерігаються в значній кількості країн в усьому світі. Зокрема, такі країни, як Данія, Ірландія, Нідерланди, Нова Зеландія та Швеція, характеризуються показниками врожайності понад 150 тис. hg/ha при мінімальному застосуванні пестицидів. Більша частина аналізованих країн світу характеризується співставними показниками застосування пестицидів та обсягів врожайності с/г культур, як наприклад, Еквадор, Німеччина, Марокко, Перу, Польща, Португалія, Велика Британія та Уругвай.





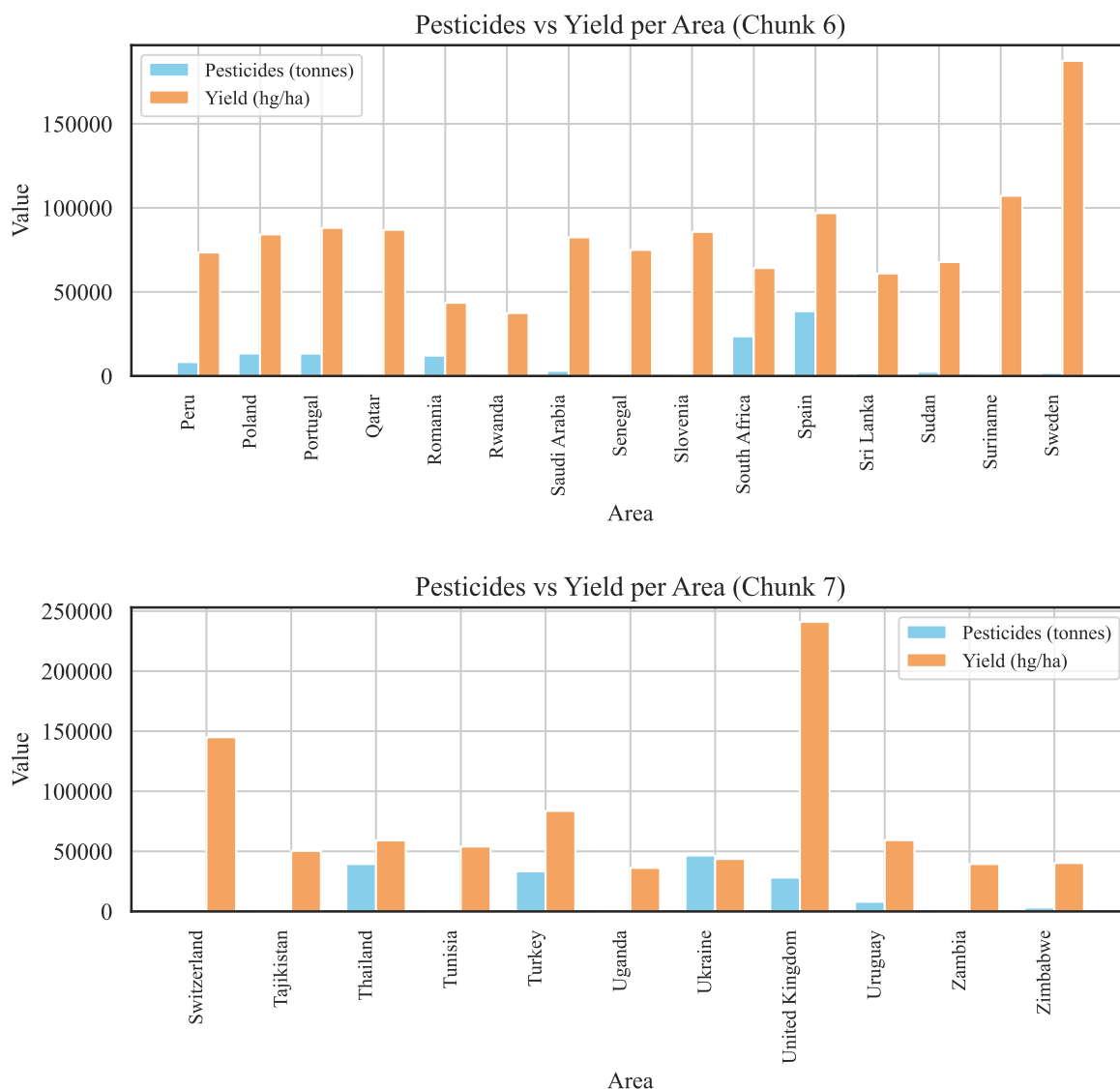


Рисунок 3.7 – Графіки порівняння середніх значень використання пестицидів і врожайності для різних країн з `df_yield_merged` DataFrame

Для більш глибокого аналізу використання пестицидів було побудовано стовпчиковий графік на основі даних із таблиці `df_yield_merged`, де за віссю абсцис відкладається категоріальна змінна `item`, а за віссю ординат – числове значення `pesticides_tonnes`, як показано на рисунку 3.8. Візуалізація даних, наведених на цьому рисунку, показує, що найбільше пестицидів використовується під час вирощування ямсу, а найменше пестицидів використовується під час вирощування бананів. Для більшості зернових культур цей показник становить на рівні 35 тис. тонн. Також графічна залежність, яку наведено на рисунку 3.9, дозволяє простежити зв'язок між середньою температурою і типами агрокультур.

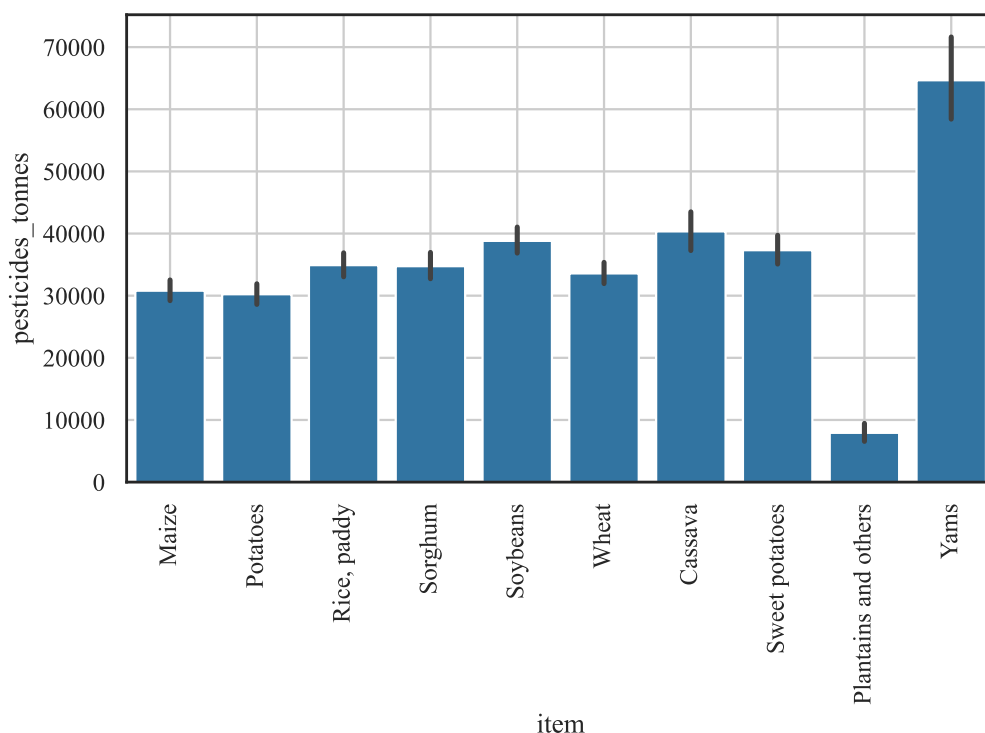


Рисунок 3.8 – Графіки залежності застосованих пестицидів від типу с/г культури

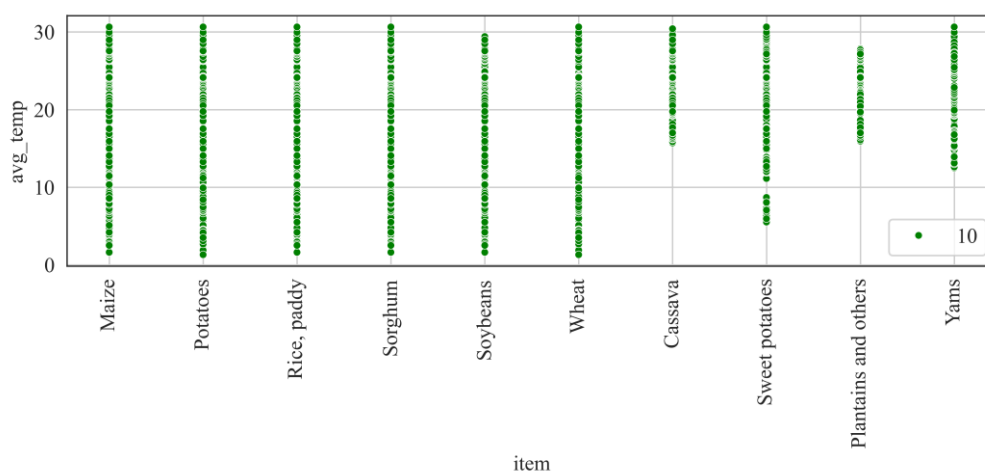


Рисунок 3.9 – Діаграма розкиду точок середньої температури по різних культурах

Як видно з рисунка 3.9 тропічні культури Cassava та Plantains потребують для росту середньої температури повітря понад 15 °C. Усі інші культури можуть зростати в більш широкому діапазоні середньорічних температур. Такі чисельні результати відповідають реальним агрокліматичним умовам вирощування відповідних с/г культур, що апріорно підтверджує адекватність розроблених обчислювальних моделей.

Таким чином, аналіз груп країн свідчить, що зв'язок між використанням пестицидів та рівнем урожайності є неоднозначним. У ряді випадків (наприклад, Данія, Ірландія, Нідерланди та інші) висока врожайність досягалася при низькому або мінімальному застосуванні пестицидів. Натомість у країнах із високим рівнем хімічного навантаження (Італія, Таїланд, Мексика, Туреччина та інші) ефект від використання пестицидів виявився обмеженим. Це свідчить, що ефективність застосування пестицидів залежить не лише від їх кількості, а й від інших факторів, зокрема агротехнологій і ґрунтокліматичних умов.

Аналогічно до реалізації підходів із попереднього статистичного аналізу даних, відповідні програмні рішення, що були створені й протестовані під час досліджень етапу розвідувального аналізу [117], дозволили одержати результати, які корелюють із раніше відомими [140, 141], що підтверджує об'єктивність розроблених у цій дисертації програмних компонент (див. додаток В.3).

Отже, проведені дослідження з розвідувального аналізу даних дозволили сформулювати кілька гіпотез для побудови моделей машинного навчання: врожайність залежить від комбінації кліматичних (температура, опади) та агротехнічних факторів (пестициди); вплив факторів може бути нелінійним, тому важливо застосовувати не лише лінійні моделі, а й ансамблеві підходи; існує виражена варіативність між країнами та культурами, що вимагає врахування категоріальних змінних (area, item) під час моделювання.

3.4 Програмні компоненти предикативного аналізу даних

Після проведення розвідувального аналізу було здійснено попередню обробку даних із метою підготовки їх до застосування під час побудови предикативних моделей на основі алгоритмів машинного навчання (див. додаток В.4). Цей етап є одним із найбільш важливих, оскільки якість підготовки даних безпосередньо впливає на ефективність та стабільність побудованих моделей.

Для об'єктивної оцінки продуктивності моделей дані було розділено на тренувальну та тестову вибірки у співвідношенні 90:10. Тренувальна вибірка

використовувалась для навчання моделей. Тестова вибірка була призначена для незалежної перевірки узагальнюючої здатності моделей до прогнозування. Такий підхід дозволяє уникнути негативного ефекту перенавчання [144] та забезпечує більш надійну оцінку точності прогнозів.

Оскільки числові ознаки мають різні діапазони значень (наприклад, кількість опадів вимірюється у сотнях мм/рік, тоді як урожайність – у десятках тисяч hg/ha), було застосовано метод стандартизації за формулою (3.1) [145]:

$$x' = \frac{x - \mu}{\sigma}, \quad (3.1)$$

де x – вихідне значення; μ – середнє значення; σ – стандартне відхилення.

У результаті всі числові ознаки отримали нульове середнє значення та одиничне стандартне відхилення, що є особливо важливим для алгоритмів із чутливістю до масштабу даних (наприклад, KNN, SVM, нейронні мережі).

Дві категоріальні змінні, *area* та *item*, не можуть напряму використовуватися в більшості алгоритмів машинного навчання через їх тип даних. Тому, для кодування зазначених категоріальних атрибутів застосовується метод One-hot Encoding, який передбачає заміну кожної категорії вектором, що складається з нулів та одиниць, та однозначно ідентифікує відповідну категорію (див. рис. 3.10) [146]. Тобто, кожне унікальне значення категорії перетворюється в окрему бінарну ознаку.

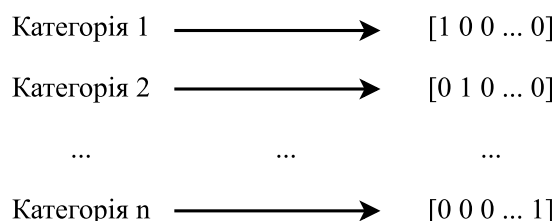


Рисунок 3.10 – Графічне представлення One-hot Encoding [146]

Цей підхід дозволяє алгоритмам працювати з категоріальними змінними без

спотворення відстаней між ними (на відміну від порядкового кодування – послідовної нумерації об'єктів числами натурального ряду).

У результаті попередньої обробки було отримано узгоджений набір даних, який складається з матриць ознак X_{train} , X_{test} , які включають як числові фактори (average_rain_fall_mm_per_year, pesticides_tonnes, avg_temp), так і закодовані категоріальні змінні (area, item), а також цільових змінних y_{train} , y_{test} – value_hg_ha, що представляє врожайність агрокультур.

Таким чином, train/test split гарантує незалежність перевірки моделей, стандартизація запобігає домінуванню ознак із великим масштабом, а One-Hot Encoding дозволяє врахувати регіональні та біологічні особливості під час прогнозування. Попередня обробка створила умови для коректного навчання моделей та забезпечила можливість адекватного порівняння різних алгоритмів.

У межах цього дослідження під час прогнозування врожайності с/г культур було застосовано певну сукупність моделей машинного навчання (див. додаток В.5), кожна з яких має власні математичні засади та відрізняється здатністю моделювати лінійні чи нелінійні залежності.

1. Linear regression є базовою моделлю для знаходження залежності між цільовою змінною y та набором ознак $X = (x_1, x_2, \dots, x_p)$ [147]. Модель описується рівнянням (3.2):

$$y = \beta_0 + \sum_{i=1}^p \beta_i x_i + \varepsilon, \quad (3.2)$$

де β_i – коефіцієнти моделі; ε – похибка.

Навчання реалізується методом найменших квадратів, мінімізуючи функцію втрат (3.3):

$$q = \min_{\beta} \sum_{j=1}^n (y_j - \hat{y}_j)^2, \quad (3.3)$$

де y_j – актуальне значення з набору даних; $\hat{y}_j$ – спрогнозоване значення.

Модель (3.2) в більшості випадків демонструє прийнятні результати за умови лінійності залежностей. Проте у випадку наявності нелінійних зв'язків потребують дослідження складніші моделі, які описані нижче.

2. KNN Regressor відноситься до алгоритмів, які ґрунтуються на припущенні, що об'єкти, розташовані близько в просторі ознак та мають подібні значення цільової змінної [148]. Прогноз для нового об'єкта x обчислюється за (3.4) як середнє значення відповідей його k найближчих сусідів:

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \quad (3.4)$$

де $N_k(x)$ – множина k найближчих сусідів точки x . Вибір гіперпараметра k суттєво впливає на баланс між шумостійкістю та точністю.

3. Decision Tree Regressor виконує рекурсивне бінарне розбиття простору ознак [149]. У вузлах вибирається ознака та порогове значення, яке мінімізує суму квадратичних відхилень у відповідності до (3.5):

$$q = \min_s \sum_{x_i \in R_1(s)} (y_i - \hat{y}_{R_1})^2 + \sum_{x_i \in R_2(s)} (y_i - \hat{y}_{R_2})^2, \quad (3.5)$$

де R_1, R_2 – підмножини після розбиття за правилом s . Древа здатні моделювати складні нелінійні залежності, але схильні до перенавчання.

4. Random Forest Regressor є ансамблем дерев рішень, які будуються на випадкових підмножинах даних і ознак [150]. Прогноз обчислюється за (3.6) як середнє по всіх деревах:

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^B T_b(x), \quad (3.6)$$

де $T_b(x)$ – прогноз b -го дерева; B – кількість дерев у лісі. Цей метод зменшує дисперсію та покращує узагальнюючу здатність.

5. Gradient Boosting Regressor будує послідовність дерев, кожне з яких коригує помилки попередніх [151]. У припущенні, що $F_m(x)$ є ансамблем після m ітерацій, то:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x), \quad (3.7)$$

де $h_m(x)$ – нове дерево, побудоване на залишках; ν – швидкість навчання. Таким чином, модель поступово мінімізує функцію втрат.

6. Support Vector Regression (SVR), метою застосування якого є знаходження функції $f(x)$, яка відхиляється від фактичних значень y_i не більше ніж на ε [152]. Оптимізаційна задача має такий вигляд (3.8):

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{за умови} \quad |y_i - (w \cdot x_i + b)| \leq \varepsilon. \quad (3.8)$$

Завдяки використанню kernel функцій (наприклад, RBF) SVR здатна моделювати складні нелінійні залежності.

7. Multi-layer Perceptron Regressor є моделлю багат шарового перцептрона (MLP), яка складається з послідовних шарів нейронів [153]. Кожен нейрон обчислює показник a_j за формулою (3.9):

$$a_j = \phi \left(\sum_i w_{ij} x_i + b_j \right), \quad (3.9)$$

де $\phi(\cdot)$ – активаційна функція (ReLU, tanh тощо). Навчання здійснюється методом зворотного поширення помилки (backpropagation), мінімізуючи функцію втрат

(наприклад, MSE). MLP здатний апроксимувати довільні нелінійні функції, проте потребує ретельного налаштування гіперпараметрів.

8. Bagging Regressor є методом, який використовує ідею Bootstrap Aggregation (Bagging): кілька моделей навчаються на випадкових підвбірках із поверненням, після чого їхні результати осереднюються. Це дозволяє знизити варіативність і підвищити стабільність прогнозу [154].

9. XGBoost (Extreme Gradient Boosting) – це оптимізована реалізація градієнтного бустингу (Gradient Boosting Regressor), яка включає регуляризацію та ефективні алгоритми обчислень [155]. Функція втрат має вигляд (3.10):

$$\zeta = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad (3.10)$$

де $\Omega(f_k)$ – регуляризаційний член, який контролює складність моделі. Це дозволяє досягати високої точності навіть на великих наборах даних.

Таким чином, використаний набір моделей охоплює як прості лінійні алгоритми, так і складні ансамблеві підходи та нелінійні моделі загального призначення. Це забезпечує можливість порівняння їхньої ефективності під час застосування в задачі прогнозування врожайності агрокультур.

Для оцінювання продуктивності алгоритмів машинного навчання було використано кілька стандартних метрик [156].

Mean Absolute Error (MAE) – середня абсолютна похибка [156]:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.11)$$

де y_i – актуальне значення з набору даних; $\hat{y}_i$ – прогнозоване моделлю значення цільової змінної для i -го екземпляру в наборі даних; n – розмір вибірки даних.

MAPE (Mean Absolute Percentage Error) – середня відносна похибка, що виражена у відсотках [156]:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (3.12)$$

RMSE (Root Mean Squared Error) – квадратична середня похибка [156]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (3.13)$$

R^2 (коефіцієнт детермінації) – частка дисперсії залежної змінної, поясненої моделлю [156]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (3.14)$$

де $\bar{y}$ – середнє значення цільової змінної в даних.

Варто зазначити, що високі значення R^2 (наближені до 1) та низькі значення MAE, MAPE і RMSE свідчать про високу точність моделі. Зокрема параметр R^2 показує, яка частина варіації цільової змінної y пояснюється моделлю.

Також доцільно акцентувати увагу на такому, що оцінювання якості моделей машинного навчання за однією фіксованою тренувальною та тестовою вибірками може призвести до нестабільних результатів, залежних від випадкового поділу даних. Для отримання більш надійної та узагальненої оцінки використовується метод крос-валідації [157]. У цьому дослідженні застосовано K-Fold Cross-Validation з числом блоків $k = 5$. Метод полягає в наступному. Повна навчальна вибірка розділяється на k приблизно рівних блоків $D_1, D_2, \dots, D_k$. На кожній ітерації один блок D_i використовується як валідаційний, а решта – як тренувальні дані. Модель навчається на тренувальних даних та оцінюється на

валідаційних. Після завершення k ітерацій обчислюється середнє значення метрик $\bar{Q}$ за формулою:

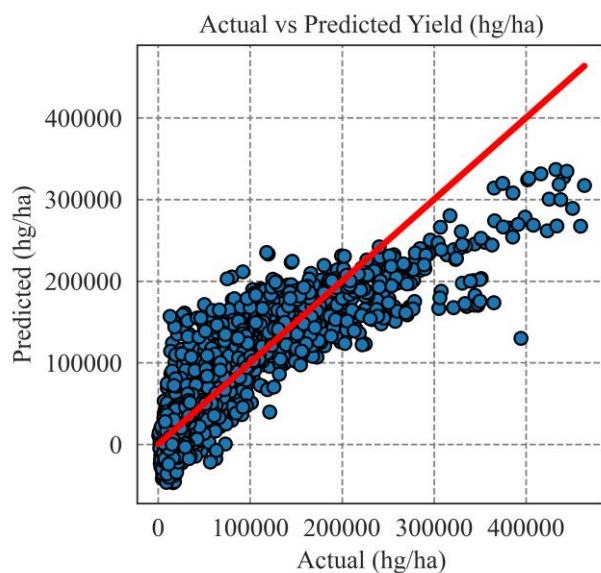
$$\bar{Q} = \frac{1}{k} \sum_{i=1}^k Q_i. \quad (3.15)$$

де Q_i – значення обраної метрики, у даному випадку R^2 на i -й ітерації.

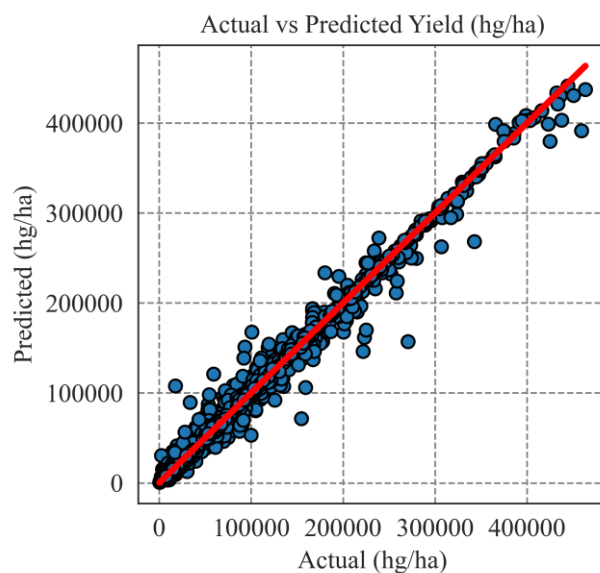
Таким чином, оцінка якості моделі ґрунтується на результатах із усіх підмножин даних, що значно знижує ризик переоцінки або недооцінки її ефективності. У таблиці 3.17 наведено результати оцінки метрик, що характеризують якість досліджуваних ML моделей під час прогнозування врожайності агрокультур (значення R^2 є результатами усереднення за (3.15)). Також на рисунку 3.11 представлено графіки актуальних та прогнозованих значень врожайності на тестових даних для всіх досліджуваних ML моделей.

Таблиця 3.17 – Результати оцінки показників предикативної якості ML моделей на тестових даних

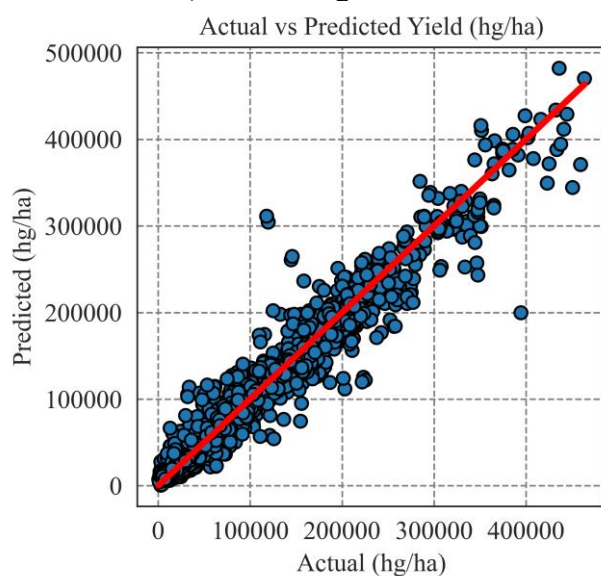
Model	MAE	MAPE	RMSE	R ²
Linear Regression	29605,2892	0,8606	42038,4356	0,7443
Random Forest	3340,5711	0,0789	9630,6006	0,9903
MLP	10508,1369	0,2470	18166,6955	0,9522
Gradient Boost	19285,1882	0,5341	29066,2639	0,8777
SVR	55680,8545	1,1706	90671,11	-0,1894
XGBoost	7898,2407	0,2153	10781,2055	0,9762
Decision Tree	3389,7582	0,0781	10781,2055	0,9831
KNN	4100,3609	0,0827	8813,3099	0,9887
Bagging Regressor	3291,1686	0,0778	8126,2939	0,9904



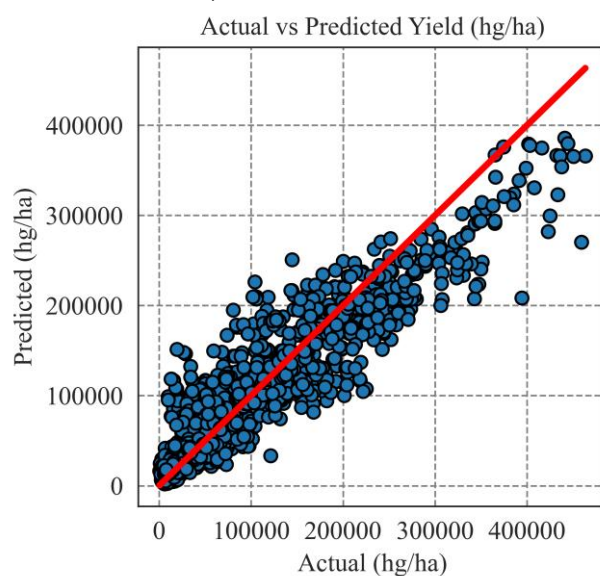
a) Linear Regression



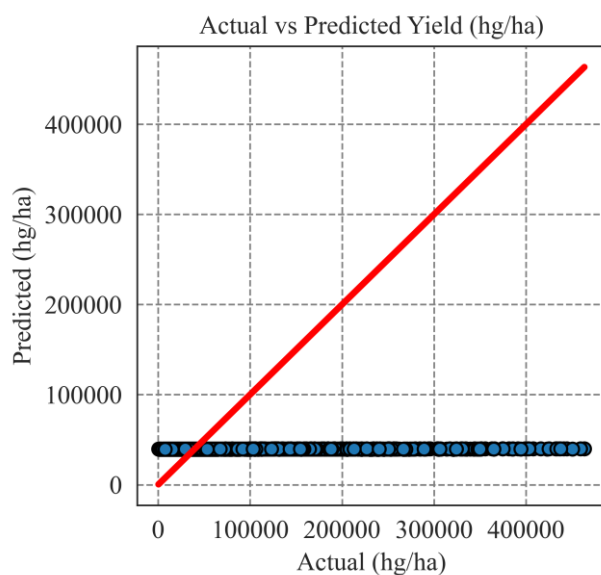
б) Random Forest



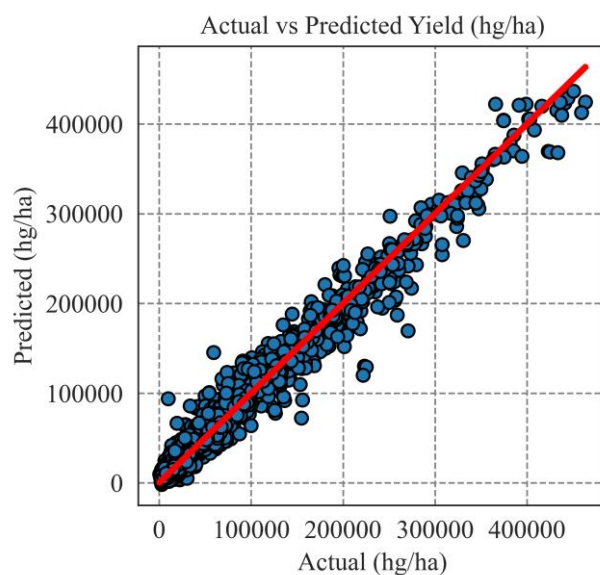
в) Multi-layer Perceptron Regressor



г) Gradient Boosting Regressor



д) Support Vector Regression



е) XGBoost

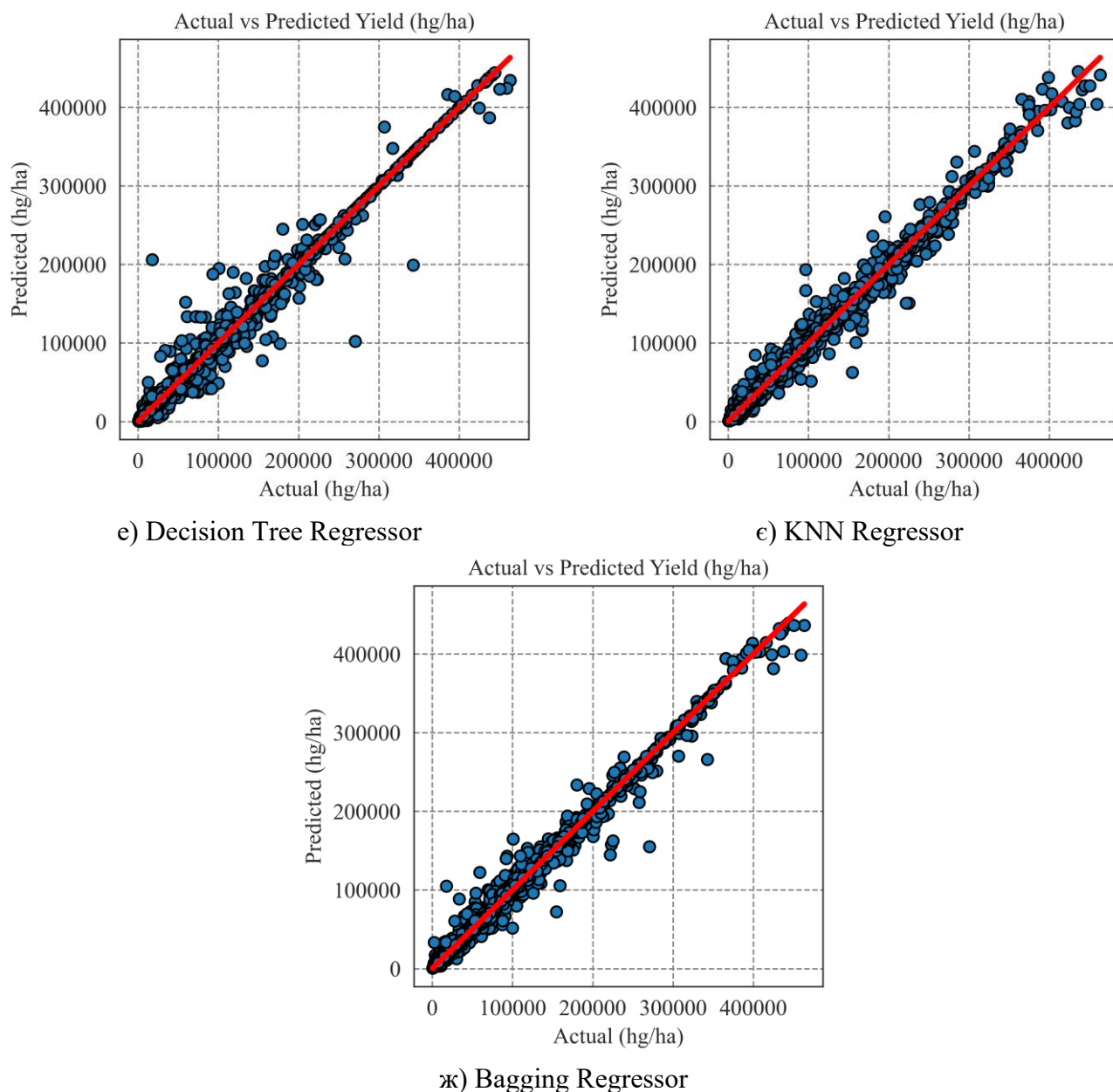


Рисунок 3.11 – Графіки актуальних та прогнозованих значень врожайності на тестових даних для різних ML моделей

У результаті аналізу отриманих результатів, які наведено на рисунку 3.11 і таблиці 3.17, було встановлено наступне. Лінійна регресія показала відносно невисоку якість ($R^2=0,7443$), що свідчить про обмеженість її здатності моделювати складні нелінійні залежності між кліматичними та агротехнічними факторами. Високі значення MAE і $RMSE$ підтверджують значну середню похибку прогнозу. Древа рішень ($R^2=0,9831$) продемонстрували значно вищу точність, зокрема низький MAE (~ 3389) та $MAPE$ ($\sim 0,0781$). Це підтверджує здатність дерев

рішень враховувати нелінійні ефекти та взаємодії між ознаками. Random Forest ($R^2=0,9903$) та Bagging Regressor ($R^2=0,9904$) показали найкращі результати серед усіх аналізованих моделей. Обидва підходи демонструють дуже низькі похибки та стабільність прогнозів, що зумовлено ефектом усереднення ансамблю дерев, яке знижує дисперсію та ризик перенавчання. KNN Regressor також досяг високої якості ($R^2=0,9887$), але має трохи більші похибки в порівнянні з ансамблевими методами. Це свідчить, що локальна подібність між об'єктами є корисною для прогнозування, однак модель чутлива до вибору гіперпараметра k та масштабу даних. XGBoost ($R^2=0,9762$) показав достатньо позитивні результати, завдяки регуляризації він забезпечив високий баланс між точністю та узагальнюючою здатністю. Модель на основі MLP досягла достатньо високого рівня прогнозування ($R^2=0,9522$), проте поступається ансамблевим методам через більшу чутливість до параметрів навчання, обмеженість обсягу даних та ймовірне недосягнення глобального мінімуму в процесі оптимізації. Gradient Boosting у базовій реалізації продемонстрував середні результати ($R^2=0,8777$), що свідчить про необхідність ретельнішої оптимізації гіперпараметрів. Модель на основі SVR показала негативний коефіцієнт детермінації ($R^2=-0,1894$), що свідчить про повну непридатність даної конфігурації моделі до поставленої задачі. Це може бути зумовлено невідповідним масштабуванням, неадекватним вибором ядра або невеликою кількістю навчальних зразків у високорозмірному просторі.

Таким чином, найбільш ефективними для прогнозування врожайності агрокультур на основі агрокліматичних даних виявилися ансамблеві методи (Random Forest та Bagging Regressor), які забезпечили найнижчі похибки та найвищі значення R^2 . Своєю чергою, алгоритми XGBoost і Decision Tree показали високу узгодженість прогнозів, підтверджуючи релевантність деревоподібних моделей під час вирішення предикативних задач для аграрних цілей. Також варто зазначити, що алгоритм MLP має перспективи при використанні більшого обсягу даних і оптимізації архітектури, однак наявні результати свідчать, що традиційні ансамблеві методи є більш стабільними. Модель на основі SVR у поточній конфігурації показала негативний результат, що свідчить про необхідність

ретельнішого налаштування ядра та параметрів регуляризації. Використання підходу на основі 5-Kfolds cross-validation дозволило усереднити коливання, спричинені випадковим поділом даних на тренувальні та тестові множини. У результаті було отримано стабільніші та репрезентативніші метрики, які відображають більш об'єктивну узагальнюючу здатність моделей, а не лише їхню ефективність на конкретному розбитті даних [117].

3.5 Програмні компоненти інтерпретації моделей прогнозування врожайності

Як зазначалося раніше, зокрема в п. 1.3 цієї дисертації, відбір інформативних ознак є важливим етапом під час побудови моделей машинного навчання. Його основна мета полягає у виділенні найбільш інформативних змінних, що мають найбільший вплив на цільову змінну, та одночасному усуненні надлишкових чи недостатньо інформативних ознак. Це може зменшити розмірність простору ознак, знизити ризик перенавчання та, в першу чергу, підвищити інтерпретованість моделей. У цьому дослідженні було застосовано кілька підходів до відбору ознак: Filter Selector, Sequential Selector, Recursive Selector, Mha Selector (PSO) з бібліотеки `mafe` [126–130]. Крім цього, також застосовано `mutual_info_regression`, з бібліотеки `sklearn.feature_selection` (див. додаток В.6).

1. Підходи на основі Filter Selector ґрунтуються на статистичних критеріях, що оцінюють силу залежності між кожною ознакою X_i та цільовою змінною y , незалежно від конкретної моделі. Зокрема, Mutual Information (MI) оцінює, наскільки знання ознаки зменшує невизначеність щодо цільової змінної. Формально взаємна інформація визначається як [158]:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \quad (3.16)$$

де $p(x, y)$ – спільний розподіл ознаки X та цільового значення Y , а $p(x)$, $p(y)$ – їхні маргінальні розподіли.

Програмна реалізація методу Filter Selector (mafese) з урахуванням [126–130] на основі МІ дозволила виділити такі найбільш релевантні ознаки (у порядку зменшення релевантності): 'year', 'average_rain_fall_mm_per_year', 'pesticides_tonnes', 'avg_temp', 'country_Argentina', 'country_Australia', 'country_Brazil', 'country_Canada', 'country_Colombia', 'country_Ecuador', 'country_India', 'country_Indonesia', 'country_Italy', 'country_Japan', 'country_Mexico', 'country_Pakistan', 'country_South Africa', 'country_Spain', 'country_Turkey', 'country_United Kingdom', 'item_Cassava', 'item_Maize', 'item_Plantains and others', 'item_Potatoes', 'item_Rice, paddy', 'item_Sorghum', 'item_Soybeans', 'item_Sweet potatoes', 'item_Wheat', 'item_Yams'.

2. Метод Sequential Selector реалізує ітераційний підхід, де ознаки додаються (Forward Selection) або видаляються (Backward Elimination) поступово, з урахуванням якості моделі [126–130]. У випадку Forward Selection алгоритм починає роботу з порожнього набору ознак. На кожному кроці додається та ознака, яка найбільше покращує метрику моделі (наприклад, R^2). Процес триває, поки не буде досягнуто заданої кількості ознак. У такому випадку критерій оптимізації можна записати як:

$$\max_{S \subseteq \Pi, |S|=k} Q(S), \quad (3.17)$$

де Π – повний набір ознак; S – підмножина вибраних ознак, що дорівнює кількості обраних ознак k ; $Q(S)$ – якість моделі на підмножині S .

Програмна реалізація методу Sequential (Forward) Selector (mafese, estimator="knn") з урахуванням [126–130] дозволила виділити такі релевантні ознаки (у порядку зменшення релевантності): 'average_rain_fall_mm_per_year', 'pesticides_tonnes', 'country_Belgium', 'item_Cassava', 'item_Maize', 'item_Plantains and others', 'item_Potatoes', 'item_Rice, paddy', 'item_Sweet potatoes', 'item_Yams'.

3. Метод Recursive Selector базується на побудові моделі та поступовому видаленні найменш значущих ознак [126–130]. Алгоритм працює наступним чином. На першому етапі будується модель на повному наборі ознак. Потім визначається важливість кожної ознаки (наприклад, у estimator="rf" – за показником Mean Decrease in Impurity). Найменш інформативні ознаки видаляються. Процес повторюється до тих пір, поки не залишиться задана кількість ознак. Математично це можна представити як пошук підмножини S , що мінімізує похибку:

$$\min_{S \subseteq \Pi, |S|=k} Error(S). \quad (3.18)$$

Програмна реалізація методу Recursive Selector (mafese, estimator="rf") з урахуванням [126–130] дозволила виділити такі найбільш релевантні ознаки (у порядку зменшення релевантності): 'year', 'average_rain_fall_mm_per_year', 'pesticides_tonnes', 'avg_temp', 'country_India', 'item_Cassava', 'item_Plantains and others', 'item_Potatoes', 'item_Sweet potatoes', 'item_Yams'.

4. Також у цьому дослідженні застосовано MhaSelector із використанням алгоритму Particle Swarm Optimization (PSO) [126–130]. PSO є методом, який моделює поведінку рою частинок. Кожна «частинка» відповідає можливій підмножині ознак. Її положення оновлюється за певними правилами [159]:

$$\begin{aligned} v_i^{t+1} &= wv_i^t + c_1r_1(p_i^t - x_i^t) + c_2r_2(g^t - x_i^t), \\ x_i^{t+1} &= x_i^t + v_i^{t+1}, \end{aligned} \quad (3.19)$$

де x_i^t – положення частинки (набір ознак); v_i^t – швидкість; p_i^t – найкраще знайдене положення частинки; g^t – найкраще глобальне положення серед рою; w, c_1, c_2 – коефіцієнти інерції та навчання; r_1, r_2 – випадкові значення. Таким

чином, PSO здійснює глобальний пошук підмножини ознак, що в теорії має забезпечувати найкращу якість моделі.

Програмна реалізація методу MhaSelector (mafese, optimizer="OriginalPSO") дозволила виділити такі найбільш релевантні ознаки (у порядку зменшення релевантності) [117]: 'year', 'average_rain_fall_mm_per_year', 'pesticides_tonnes', 'country_Albania', 'country_Algeria', 'country_Angola', 'country_Argentina', 'country_Azerbaijan', 'country_Bahamas', 'country_Bahrain', 'country_Bangladesh', 'country_Belarus', 'country_Belgium', 'country_Botswana', 'country_Brazil', 'country_Bulgaria', 'country_Burkina Faso', 'country_Cameroon', 'country_Chile', 'country_Colombia', 'country_Denmark', 'country_Dominican Republic', 'country_Ecuador', 'country_El Salvador', 'country_Eritrea', 'country_France', 'country_Germany', 'country_Ghana', 'country_Guatemala', 'country_Guinea', 'country_Guyana', 'country_Haiti', 'country_Honduras', 'country_Hungary', 'country_Iraq', 'country_Ireland', 'country_Italy', 'country_Japan', 'country_Kenya', 'country_Latvia', 'country_Lesotho', 'country_Libya', 'country_Lithuania', 'country_Madagascar', 'country_Malawi', 'country_Malaysia', 'country_Mauritania', 'country_Mexico', 'country_Namibia', 'country_Nepal', 'country_New Zealand', 'country_Niger', 'country_Norway', 'country_Pakistan', 'country_Papua New Guinea', 'country_Portugal', 'country_Qatar', 'country_Romania', 'country_Rwanda', 'country_Saudi Arabia', 'country_Slovenia', 'country_South Africa', 'country_Spain', 'country_Sweden', 'country_Tajikistan', 'country_Turkey', 'country_Ukraine', 'country_Uruguay', 'country_Zambia', 'country_Zimbabwe', 'item_Cassava', 'item_Maize', 'item_Potatoes', 'item_Sorghum', 'item_Soybeans', 'item_Sweet potatoes', 'item_Wheat'.

Додатково застосовано окремий метод mutual_info_regression (scikit-learn) [160], який оцінює залежність кожної числової ознаки від цільової змінної. Це дозволило побудувати рейтинг топ-10 ознак, що найбільше впливають на врожайність, як це показано на рисунку 3.12. Цей метод є неконтрольованим і не залежить від конкретної моделі, що робить його зручним для попереднього аналізу.

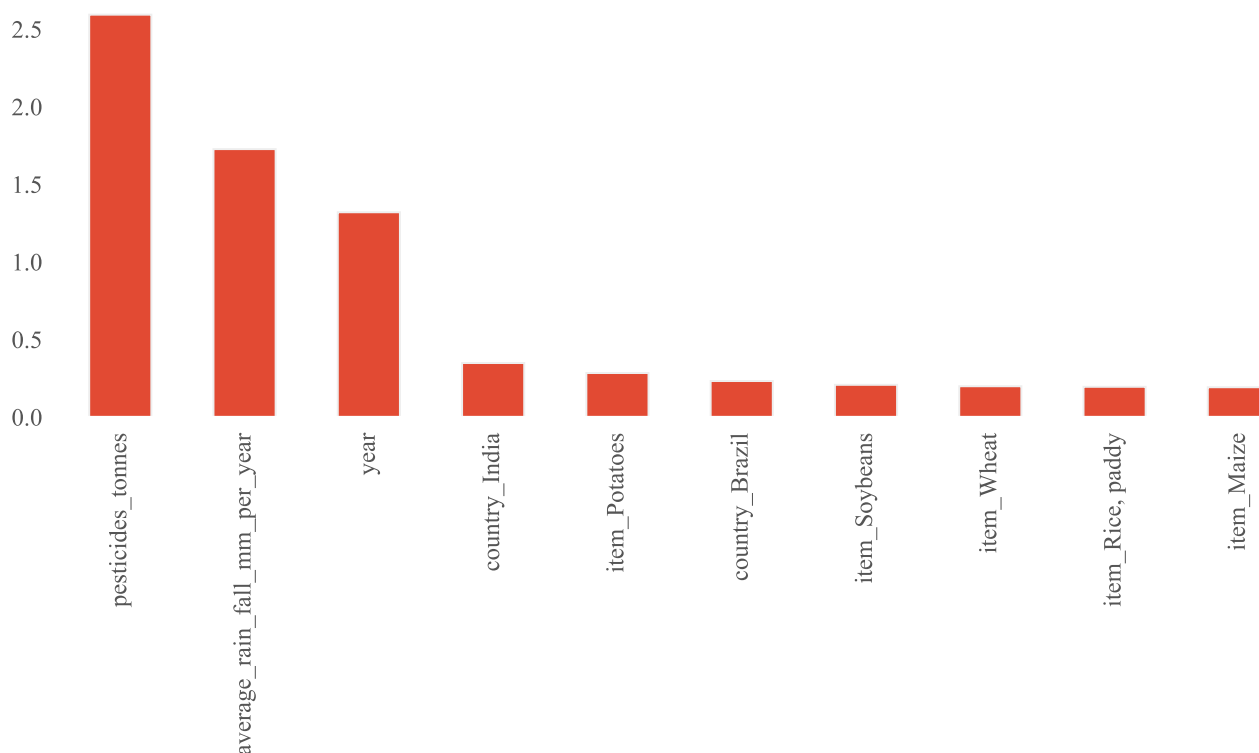


Рисунок 3.12 – Графік важливості ознак відповідно до методу `mutual_info_regression` (scikit-learn)

Таким чином, Filter Selector (MI) показав, що топ-4 релевантними ознаками є 'year', 'average_rain_fall_mm_per_year', 'pesticides_tonnes' та 'avg_temp'. Sequential (Forward) Selector на базі KNN дозволив виділити невелику підмножину ознак (~10), яка забезпечує порівняну якість із повною множиною, і першими двома найважливішими ознаками є 'average_rain_fall_mm_per_year', 'pesticides_tonnes'. Recursive Feature Elimination на базі Random Forest підтвердив важливість факторів 'year', 'average_rain_fall_mm_per_year', 'pesticides_tonnes', 'avg_temp'. Метод PSO дозволив відшукати альтернативні підмножини ознак, які також демонструють високу предикативну здатність, що підтверджує доцільність використання метаевристик для задач високої розмірності. Однак у сенсі зменшення розмірності в порівнянні з початковою, такий підхід є недостатньо ефективним. Mutual Information Regression наочно продемонстрував, які ознаки спричиняють найбільший інформаційний внесок у пояснення врожайності, що узгоджується з результатами ансамблевих моделей [117].

Так, наприклад, алгоритм Random Forest дозволяє не лише здійснювати прогноз, а й оцінювати відносну значущість кожної ознаки в процесі прийняття рішень. Для цього застосовується метрика Mean Decrease in Impurity (MDI), яка визначає, наскільки конкретна ознака сприяє зменшенню невизначеності (Impurity) у вузлах дерев, що входять до складу ансамблю [161]. Формально, важливість ознаки f_j можна записати в такому вигляді [162]:

$$I(f_j) = \frac{1}{T} \sum_{t=1}^T \sum_{n \in N_t, v(n)=f_j} p(n) \Delta i(n), \quad (3.20)$$

де T – кількість дерев у лісі; N_t – множина вузлів дерева t ; $v(n)$ – ознака, за якою відбулося розбиття у вузлі n ; $p(n)$ – частка вибірки, що потрапила у вузол n ; $\Delta i(n)$ – зменшення impurity (наприклад, дисперсії) після розбиття.

У результаті проведеного аналізу було отримано ранжування всіх ознак моделі Random Forest за їхньою значущістю. Серед них виділено сім найбільш впливових факторів, що визначають продуктивність сільськогосподарських культур, як показано на рис. 3.13.

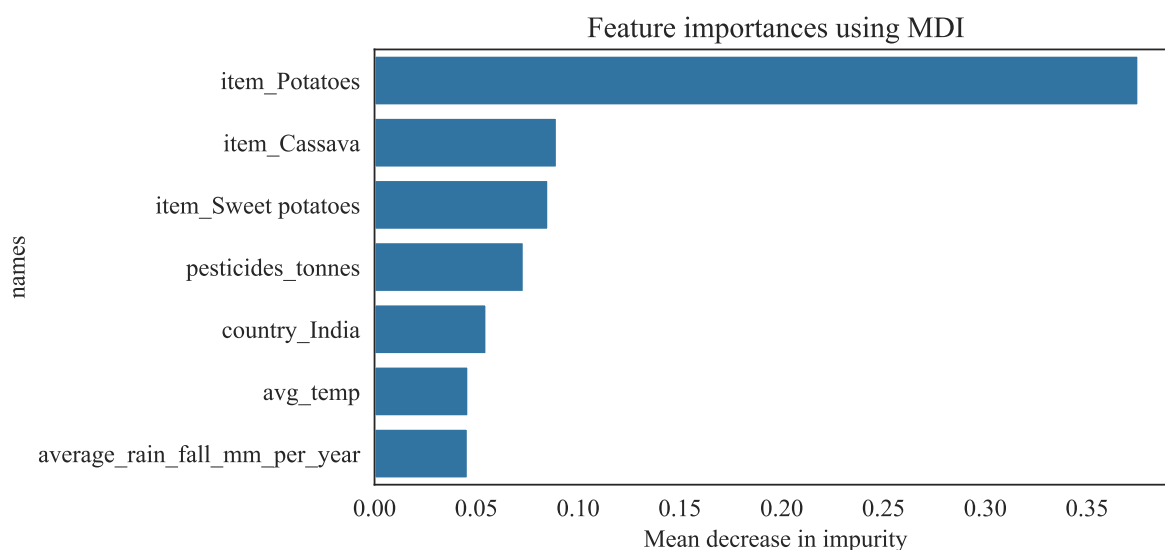


Рисунок 3.13 – Гістограма MDI моделі Random Forest

Згідно з результатами, які наведено на рисунку 3.13, серед семи найбільш впливових ознак найвищу значущість продемонстрував фактор, що відповідає за культуру картоплі `item_Potatoes`. Це узгоджується з тим, що картопля є однією з агрокультур із найбільшою врожайністю у досліджуваному наборі даних. Подібну тенденцію виявлено і для `item_Cassava`, яка також посідає провідне місце за обсягами виробництва. Четвертою за важливістю виявилася ознака, пов'язана з обсягом використання пестицидів, що відповідає очікуванням, оскільки рівень хімічного захисту впливає на врожайність. Серед просторових чинників найбільш значущим виявилось вирощування культур в Індії, що пояснюється найбільшими сумарними обсягами виробництва в цій країні в аналізованому вихідному наборі даних. Також суттєву роль відіграють кліматичні характеристики: середньорічні показники кількості опадів і температури повітря. Отримані результати підтверджують початкові припущення щодо їхнього визначального впливу на очікувану врожайність.

Таким чином, використання кількох методів відбору ознак дозволило підтвердити, що врожайність агрокультур залежить не лише від кліматичних параметрів, а й від категоріальних характеристик (культура, регіон). Використання фільтраційних, рекурсивних, послідовних та метаввристичних методів у сукупності забезпечує високу стабільність результатів і підвищення інтерпретованості предикативних результатів.

Також, у процесі дослідження було застосовано декілька сучасних методів пояснювального штучного інтелекту (XAI), спрямованих на інтерпретацію результатів ансамблевих моделей, зокрема для Random Forest (див. додаток В.7). Основна мета полягала у визначенні відносного внеску кожної ознаки в підсумковий прогноз урожайності, а також у локальній інтерпретації індивідуальних прогнозів [117].

Метод Permutation Importance (PI), який є частиною бібліотеки `eli5.sklearn`, полягає в багаторазовій випадковій перестановці значень конкретної ознаки у тестовій вибірці та обчисленні зниження якості моделі [131, 163]. Важливість

ознаки f_i визначається як середнє зменшення метрики точності (наприклад, коефіцієнта детермінації R^2) після перестановки [164, 165]:

$$PI(f_i) = \frac{1}{K} \sum_{k=1}^K \left(Q(\hat{y}, y) - Q(\hat{y}_{f_i}^{(k)}, y) \right), \quad (3.21)$$

де $Q(\cdot)$ – функція якості; $\hat{y}$ – прогноз моделі на початкових даних; $\hat{y}_{f_i}^{(k)}$ – прогноз після випадкової перестановки ознаки f_i на k -й ітерації; K – кількість повторів.

Цей метод не залежить від внутрішньої структури моделі, що робить його універсальним для будь-яких алгоритмів. У даному дослідженні PI реалізовано з використанням бібліотеки eli5 [131, 163], результат наведено в таблиці 3.18.

Таблиця 3.18 – Результати дослідження Permutation Importance

Weight	Feature
1.0981 ± 0.0643	item_Potatoes
0.3961 ± 0.0409	item_Cassava
0.2706 ± 0.0151	item_Sweet potatoes
0.1875 ± 0.0221	pesticides_tonnes
0.1195 ± 0.0064	avg_temp
0.1126 ± 0.0126	country_India
0.1005 ± 0.0056	average_rain_fall_mm_per_year
0.0667 ± 0.0094	year
0.0558 ± 0.0041	item_Yams
0.0457 ± 0.0034	item_Plantains and others
0.0396 ± 0.0057	country_Canada
0.0308 ± 0.0046	country_Japan
0.0220 ± 0.0073	country_Australia
0.0139 ± 0.0012	item_Rice, paddy
0.0124 ± 0.0008	country_United Kingdom
0.0121 ± 0.0009	item_Maize
0.0075 ± 0.0001	country_Netherlands
0.0074 ± 0.0007	country_Germany
0.0065 ± 0.0014	country_Ecuador
0.0058 ± 0.0010	country_Switzerland
... 95 more ...	

Аналіз методом Permutation Importance виявив найбільш значущі ознаки, що визначають прогноз врожайності с/г культур (див. табл. 3.18). Найвищу вагу отримала ознака item_Potatoes, що підтверджує її визначальну роль у структурі

даних. Значними предикторами також виявилися культури `item_Cassava` та `item_Sweet potatoes`. Високу важливість продемонструвала ознака, яка характеризує обсяг використання пестицидів, що узгоджується з теоретичними очікуваннями щодо її впливу на врожайність. Серед кліматичних змінних найбільш релевантними виявилися середньорічна температура та кількість опадів. Крім того, суттєву роль відіграє географічний фактор, особливо для спостережень з Індії, яка характеризується найбільшими обсягами виробництва в аналізованій вибірці. Таким чином, отримані результати підтверджують як агрономічну, так і просторово-кліматичну обумовленість формування врожайності, а також ідентичні тим, що були отримані відповідно до метрики Mean Decrease in Impurity (MDI) моделі Random Forest (див. рис. 3.13).

Бібліотека `Eli5` забезпечує також інструменти для інтерпретації моделей машинного навчання, включаючи візуалізацію глобальної важливості ознак та локального внеску ознак під час прогнозування. У випадку локальної інтерпретації для об'єкта x_i модель представляє прогноз як лінійну комбінацію впливів ознак за наступним виразом [166]:

$$\hat{y}_i = b + \sum_{j=1}^p w_j \cdot x_{ij}, \quad (3.22)$$

де w_j – ваговий внесок ознаки x_{ij} ; b – зміщення.

У випадку нелінійних моделей, як Random Forest, що розглядається, `Eli5` застосовує апроксимаційний підхід для відображення внеску кожної ознаки в конкретний прогноз врожайності агрокультур. Далі в таблиці 3.19 наводиться приклад, в якому було здійснено пояснення передбачень для підмножини даних (зокрема, для культури `Maize` (кукурудза) в Україні), що дозволило оцінити, які фактори відіграли ключову роль у відповідному рішенні моделі.

Аналіз локального передбачення (див. табл. 3.19) моделі Random Forest за допомогою бібліотеки `Eli5` дозволив визначити внесок окремих ознак у прогноз

врожайності для вибраного спостереження (культура кукурудза в Україні). Найбільший позитивний внесок забезпечує базовий зсув моделі (BIAS), що відображає середнє значення прогнозованої величини. Суттєвий додатковий внесок забезпечили ознаки середньорічна температура avg_temp (зі значенням нижче середнього) та належність до культури кукурудза item_Maize, які разом суттєво підвищують прогноз врожайності. Негативний внесок спостерігається для ряду ознак, серед яких належність до змінних рік year та середньорічні опади average_rain_fall_mm_per_year, що знижували прогнозоване значення.

Таблиця 3.19 – Результати пояснення вибраного спостереження з Eli5

y (score **41363.430**) top features

Contribution [?]	Feature	Value
+76876.215	<BIAS>	1.000
+20705.117	avg_temp	-1.793
+14336.291	item_Maize	1.000
+3627.502	item_Soybeans	0.000
+396.773	country_Romania	0.000
+77.717	country_Turkey	0.000
+41.464	country_Lithuania	0.000
+22.531	country_Armenia	0.000
+13.993	country_Montenegro	0.000
-16.788	country_Belarus	0.000
-26.627	country_Azerbaijan	0.000
-32.370	country_Bulgaria	0.000
-131.504	country_Kazakhstan	0.000
-152.467	country_Tajikistan	0.000
-255.424	country_Australia	0.000
-318.616	country_Spain	0.000
-428.492	country_Poland	0.000
-535.265	country_Hungary	0.000
-721.324	year	0.062
-1415.051	country_Canada	0.000
-2626.790	item_Plantains and others	0.000
-3403.842	country_Ukraine	1.000
-3545.024	item_Yams	0.000
-8576.135	item_Cassava	0.000
-8782.905	pesticides_tonnes	-0.301
-10672.040	item_Sweet potatoes	0.000
-11329.261	average_rain_fall_mm_per_year	-0.813
-21764.247	item_Potatoes	0.000

Таким чином, локальна інтерпретація підтверджує релевантність

кліматичних характеристик та типу культури як ключових детермінант у прогнозуванні врожайності.

Другий XAI метод, що був використаний з тих самих цілей, як і eli5 – це LIME (Local Interpretable Model-agnostic Explanations) [132, 167]. LIME дозволяє інтерпретувати рішення моделі шляхом побудови локальної апроксимації у вигляді інтерпретованої моделі (наприклад, лінійної регресії) навколо точки зацікавленості. Для об'єкта x формується множина збурених вибірок $\{x'_1, x'_2, \dots, x'_n\}$, які зважуються за відстанню до вихідного об'єкта x . На цій вибірці навчається проста модель $g \in G$, що мінімізує функцію нижче [168]:

$$L(f, g, \pi_x) + \Omega(g), \quad (3.23)$$

де f – чорна скринька (оригінальна модель); π_x – міра близькості між збуреними зразками та x ; $\Omega(g)$ – регуляризація складності моделі g . У результаті формується набір коефіцієнтів, що описують локальний вплив ознак на прогноз для конкретного об'єкта.

Нижче наводиться приклад, у якому було здійснено пояснення передбачень для підмножини даних (зокрема, для культури Maize (кукурудза) в Україні) за допомогою LIME, як показано на рисунку 3.14. Перелік ознак досить широкий, отже, подання результату візуально (див. рис. 3.14) не є зручним, так як вдалося показати лише їх частину. Тому ознаки, які вплинули на прогноз визначеного рядка з набору даних (такий самий як і для eli5 вище), наводяться у вигляді Python List: ('avg_temp <= -0.60', 18494.59), ('-0.34 < pesticides_tonnes <= 0.19', 10051.981671985182), ('item_Maize > 0.00', 8321.551740490433), ('average_rain_fall_mm_per_year <= -0.77', 4165.4163362300405), ('country_Ukraine > 0.00', -3968.5059557740665), ('-0.08 < year <= 0.91', 3769.2106084926004).

Метод LIME для тієї ж самої вибірки (Maize, Україна) показав, що найбільш вагомим фактором під час прогнозування врожайності є середньорічна температура ($\text{avg_temp} \leq -0.60$), внесок якої становив близько +18,5 тис. hg/ha.

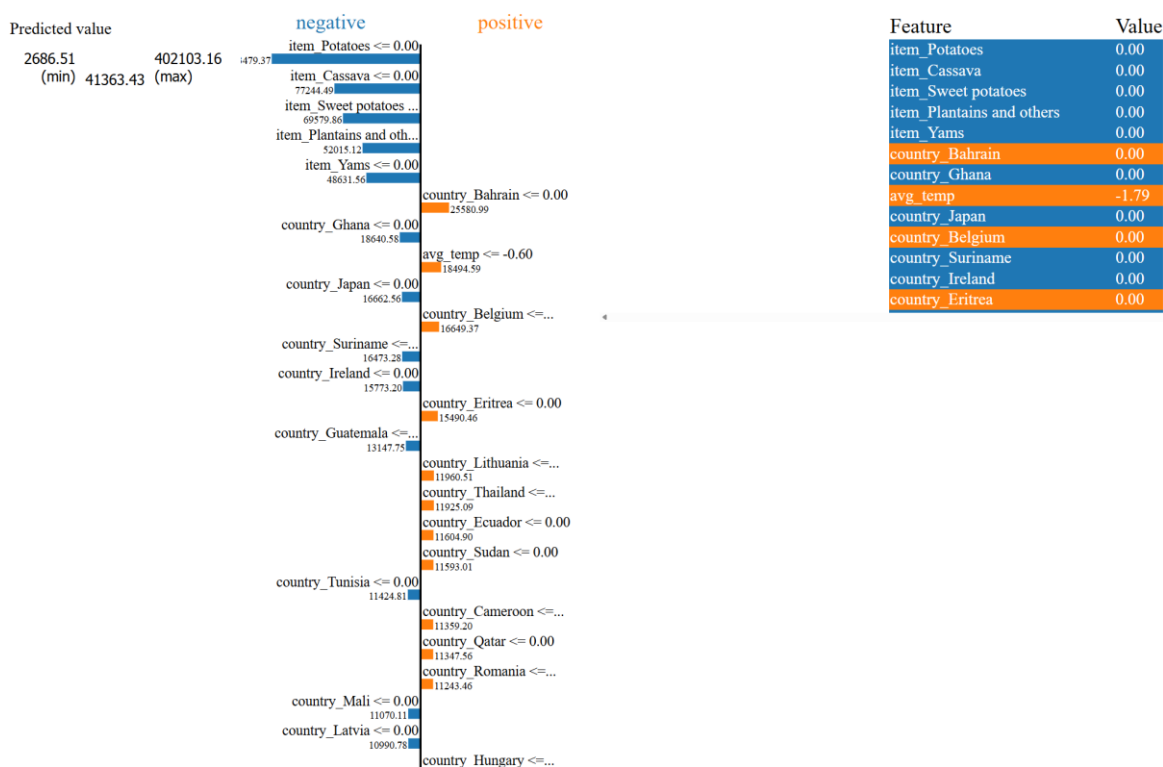


Рисунок 3.14 – Приклад LIME інтерпретації результату прогнозування

Додатково значущий позитивний ефект забезпечили: обсяг використання пестицидів у середньому діапазоні ($-0,34 < \text{pesticides_tonnes} \leq 0,19$) з внеском понад +10 тис. hg/ha; ознака належності до культури кукурудза ($\text{item_Maize} > 0,00$) з внеском близько +8,3 тис. hg/ha; низький рівень середньорічних опадів ($\text{average_rain_fall_mm_per_year} \leq -0,77$) з внеском близько +4,1 тис. hg/ha. Водночас було виявлено негативний внесок просторового чинника country_Ukraine (близько -4,0 тис. hg/ha), що вказує на нижчі відносні значення врожайності у межах країни порівняно з глобальним середнім рівнем. Додатковим позитивним фактором виявився часовий параметр ($-0,08 < \text{year} \leq 0,91$), що підвищив прогнозне значення врожайності приблизно на +3,8 тис. hg/ha.

3.6 Потокова модель програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур

На підставі виконаних досліджень зі збору, попередньої обробки, аналізу, підготовки навчальних (НВ) та тестових (ТВ) наборів даних, навчання та валідації

моделей машинного навчання, оцінки важливості ознак та пояснювального штучного інтелекту, уточнені кроки дослідження в цьому розділі представлені у вигляді деталізованої потокової структурно-алгоритмічної схеми, як показано на рисунку 3.15.

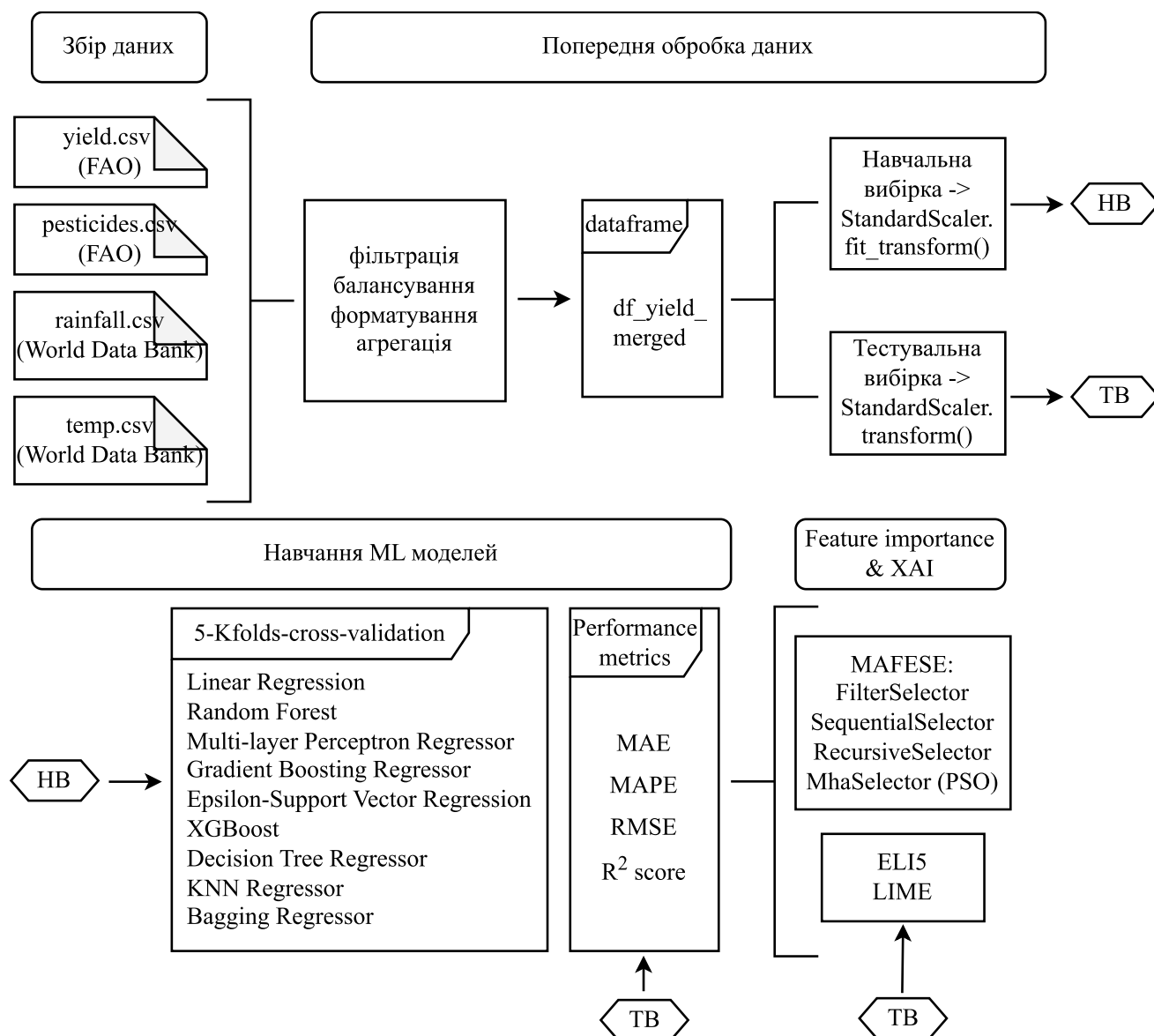


Рисунок 3.15 – Деталізована структурно-алгоритмічна послідовність інтелектуальної трансформації даних із застосуванням ML та XAI

Таким чином, отримана деталізована структурно-алгоритмічна послідовність інтелектуальної трансформації даних із урахуванням інтерпретації результатів, що наведені в п. 3.5 за допомогою XAI, підтверджує провідну роль

кліматичних змінних (температура, опади), а також факторів, пов'язаних із типом культури та використанням пестицидів, що раніше було встановлено методами Permutation Importance й MDI Random Forest, та іншими Feature Importance методами [117]. Обидва підходи, Eli5 і LIME, вказують на ключову роль ознаки середньорічної температури для одного з рядів набору даних (Maize, Україна). Водночас метод LIME дозволив кількісно оцінити ефект інтервальних значень змінних (наприклад, конкретні діапазони для температури, опадів та пестицидів), що підвищує інтерпретованість моделі для агрономічних застосувань. Ці інтервальні значення є стандартизованими, і у разі необхідності можна скористатися тією ж бібліотекою StandardScaler для отримання оригінальних значень. Комбінація глобальних (MDI, PI) і локальних методів (Eli5, LIME) демонструє високу узгодженість у визначенні основних чинників урожайності агрокультур, що підтверджує релевантність ML моделей для прогнозування в аграрному секторі та забезпечує необхідний рівень прозорості їх роботи.

3.7 Висновки за третім розділом

У цьому розділі дисертації вирішено одне з основних дослідних завдань, що полягало в розробці та дослідженні ефективності програмно-алгоритмічних засобів комплексної трансформації даних агромоніторингу, які реалізують етапи попередньої обробки, інтелектуальної аналітики, а також інтерпретації та пояснення предикативних моделей задля автоматизованої підтримки прийняття агротехнічних рішень. Під час проведених досліджень одержано такі результати:

1. Обґрунтовано можливість і доцільність застосування широкого спектра бібліотек Python, які забезпечують повний цикл обробки даних: від базових наукових інструментів (NumPy, Pandas, Matplotlib, Seaborn) до спеціалізованих засобів машинного навчання (Scikit-learn, XGBoost), оптимізації (mafese), збереження моделей (joblib), а також пояснювального штучного інтелекту (Eli5, LIME). Такий підхід дозволив реалізувати відтворюваний та масштабований обчислювально-аналітичний процес у межах повноциклової програмно-

алгоритмічної архітектури прогнозування врожайності агрокультур.

2. Програмно реалізовано поетапне злиття кількох різних джерел даних: показників урожайності, кліматичних даних та агротехнічних характеристик. Здійснено процедури попередньої статистичної обробки цих даних, а саме: попередню очистку, уніфікацію форматів і приведення числових характеристик до єдиної структури. У результаті сформовано узгоджений табличний набір даних, що є придатним до подальшого аналізу та побудови предикативних моделей прогнозування врожайності.

3. Виконано комплексний аналіз вхідних даних, включаючи групування за культурами та країнами, оцінку кореляційних зв'язків між агрономічними та кліматичними показниками, побудову гістограм, діаграм розсіювання та часових рядів. На підставі цього було встановлено суттєву залежність врожайності від типу агрокультур, кліматичних характеристик (опадів і температури) та просторового чинника.

4. Реалізовано програмними засобами процедури стандартизації числових ознак та кодування категоріальних змінних методом One-hot. Це забезпечило сумісність та співставність ознак, а також коректність подальшого використання моделей машинного навчання.

5. Програмно реалізовано та проаналізовано ефективність дев'яти регресійних моделей на основі алгоритмів машинного навчання, а саме: Linear Regression, Random Forest, MLP, Gradient Boosting, SVR, XGBoost, Decision Tree, KNN, Bagging Regressor. Виконано навчання моделей на тренувальній вибірці та тестування на відкладених даних. Найкращі результати продемонстрували ансамблеві алгоритми, зокрема Bagging Regressor та Random Forest, що забезпечили коефіцієнт детермінації $R^2 > 0,99$. Лінійна регресія показала нижчу точність ($R^2 = 0,74$), а метод опорних векторів (SVR) виявився непридатним у даній постановці задачі ($R^2 < 0$). Візуалізація співвідношення фактичних та прогнозованих значень підтвердила адекватність роботи ансамблевих моделей та доцільність їх подальшого застосування під час прогнозування врожайності агрокультур на основі результатів агромоніторингу.

6. Для всіх моделей застосовано п'ятикратну крос-валідацію, що дозволило усереднити метрики та уникнути переоцінки якості моделі. Теоретично це відповідає обчисленню очікуваного значення похибки на різних підмножинах даних, що підвищує надійність зроблених висновків щодо предикативних можливостей програмно реалізованих моделей на основі аналізованих алгоритмів машинного навчання.

7. Досліджено низку програмно реалізованих стратегій відбору інформативних ознак: фільтраційні методи (Mutual Information), послідовний відбір (Sequential (Forward) Selection), рекурсивне видалення ознак (RFE) та метаевристичний алгоритм (Particle Swarm Optimization). Усі підходи підтвердили провідну роль кліматичних змінних (середньорічна температура повітря та кількість опадів), показників використання пестицидів і просторових характеристик під час прогнозування врожайності агрокультур. Це дозволило зменшити розмірність предикативної моделі без суттєвої втрати точності прогнозування.

8. Застосовано глобальні (Random Forest MDI, Permutation Importance) та локальні методи пояснювального штучного інтелекту (Eli5, LIME) задля інтерпретації моделей прогнозування врожайності агрокультур. Локальні пояснення показали, як поєднання цих ознак впливає на окремий прогноз на прикладі вирощування кукурудзи в агрокліматичних умовах України, що підтверджує узгодженість глобальних та локальних оцінок і забезпечує прозорість запропонованих предикативних моделей.

9. Запропонована повноциклова потокова модель програмно-алгоритмічного забезпечення із застосуванням ML та ХАІ продемонструвала високу ефективність під час вирішення задачі прогнозування врожайності сільськогосподарських культур. Використання ансамблевих моделей у поєднанні з методами відбору ознак та ХАІ забезпечує не лише високу точність прогнозу ($R^2 > 0,99$), але і його інтерпретованість, що має принципове значення для практичних застосувань у системах агромоніторингу із автоматизованою підтримкою прийняття рішень.

РОЗДІЛ 4

КОМП'ЮТЕРНА ТЕХНОЛОГІЯ КРОСРІВНЕВОЇ ВЗАЄМОДІЇ ПРОГРАМНО-АЛГОРИТМІЧНИХ КОМПОНЕНТ ПРОГНОЗУВАННЯ ВРОЖАЙНОСТІ АГРОКУЛЬТУР ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

4.1 Опис застосованих компонент та ролей імпортованих модулів

У цьому розділі комп'ютерну технологію прогнозування врожайності агрокультур реалізовано як Flask-додаток [169], що поєднує модулі вебінтерфейсу, бази даних, локального виконання навченої прогнозової моделі (інференсу), засобів пояснювального штучного інтелекту та інтерфейсу взаємодії з користувачами. На початковому етапі користувач через вебформу задає набір вхідних параметрів (географічна ознака, тип агрокультури, кліматичні та агротехнічні характеристики), на підставі яких формується структурований вхідний вектор. Числові ознаки стандартизуються з використанням StandardScaler, відновленого за збереженими статистичними атрибутами, що забезпечує консистентність етапів попередньої обробки та інференсу. Прогнозування здійснюється попередньо навченою моделлю Random Forest, завантаженою через joblib. Для підвищення інтерпретованості результатів використано локальні пояснення за допомогою ELI5. Також реалізовано механізм, за допомогою якого HTML-репрезентація пояснення додається до тіла електронного листа, що надсилається користувачу у фоновому режимі. Для збереження користувацьких облікових даних і історичних спостережень використано SQLAlchemy ORM. Для інтеграції зовнішніх джерел даних реалізовано захищений API-маршрут, що приймає пакетні вставки даних у форматі JSON. Запропонована реалізація акцентує увагу на відтворюваності попередньої обробки, прозорості моделей та практичній інтеграції результатів під час підтримки прийняття рішень [170].

Програмна реалізація сервісу прогнозування врожайності побудована із використанням низки зовнішніх бібліотек і модулів (див. додаток Г.1, Г.3), які

забезпечують вебфункціональність, надійну роботу з даними, розгортання й відпрацювання модулів на основі алгоритмів машинного навчання і пояснювального штучного інтелекту та взаємодію з користувачем через графічний інтерфейс.

Базовим середовищем розробки обрано фреймворк Flask [169], що являє собою легковаговий WSGI фреймворк, який реалізує маршрутизацію HTTP-запитів, управління контекстом запиту та рендеринг шаблонів. Для забезпечення зручності роботи з формами користувача застосовано розширення Flask-WTF у поєднанні з WTForms, що надає декларативний спосіб опису форм, здійснює валідацію введених даних і вбудовує захист від CSRF-атак. Також використано модуль Flask-Bootstrap, який забезпечує інтеграцію з CSS-фреймворком Bootstrap для формування адаптивного та зручного у використанні інтерфейсу.

Суттєвою складовою архітектури є підсистема управління користувацькими сесіями та аутентифікацією. Вона реалізована через Flask-Login, що дозволяє зберігати стан користувача між запитами, надає механізми для входу, виходу та обмеження доступу до приватних маршрутів. Під час реалізації застосовано callback `user_loader` для відновлення користувача з БД, декоратор `login_required` для приватних маршрутів. Для збереження облікових даних і результатів агромоніторингу використовується Flask-SQLAlchemy, який забезпечує доступ до реляційної БД через об'єктно-реляційне відображення (ORM). Моделі користувачів і даних визначено у вигляді Python-класів із декларативним описом типів колонок. Для підвищення безпеки аутентифікації паролі не зберігаються у відкритому вигляді, а хешуються з використанням функцій `generate_password_hash` і `check_password_hash` з пакету Werkzeug, які реалізують криптографічно стійкі алгоритми на основі PBKDF2-SHA256.

Робота з вхідними даними під час прогнозування врожайності реалізована за допомогою бібліотек `pandas` та `NumPy`, що забезпечують створення, трансформацію та зручну обробку табличних структур. На етапі попередньої обробки числові ознаки стандартизуються за допомогою `StandardScaler` із пакету `scikit-learn`, при цьому статистичні параметри (середнє значення, дисперсія та

масштаби) відновлюються із попередньо збережених JSON-файлів. Такий підхід гарантує консистентність між етапами навчання та інференсу моделі. Сама модель машинного навчання завантажується з використанням бібліотеки `joblib`, що дозволяє відновлювати її параметри в серіалізованій формі.

Для підвищення прозорості функціонування вебзастосунку та забезпечення інтерпретованості прогнозних моделей застосовано бібліотеку `eli5`. Вона надає можливість формувати локальні пояснення у вигляді вагових коефіцієнтів внеску окремих ознак у прийняте рішення моделі. Пояснення відображаються у форматі HTML та інтегруються до електронного листа з результатами обробки запиту для користувача. Додаткову функціональність із візуалізації часових рядів і трендів забезпечує `Plotly`, що генерує інтерактивні графіки у форматі HTML і дозволяє аналізувати залежності між кліматичними показниками та врожайністю.

Механізм взаємодії з користувачем передбачає відправку прогнозів та пояснень на електронну пошту. Для цього використано бібліотеки `smtplib` та `email.message.EmailMessage`, які реалізують клієнт SMTP і забезпечують формування мультимедійних листів із plain-text та HTML-вмістом. Конфігураційні параметри, зокрема SMTP-логін, паролі, секретні ключі API, шляхи до моделей та ключ бази даних, зчитуються за допомогою модуля `python-dotenv` зі змінних середовища, що відповідає вимогам безпечного управління конфігурацією. Для асинхронної відправки повідомлень і розвантаження основного потоку запитів використано стандартний модуль `threading` у поєднанні з декоратором `copy_current_request_context`, що дозволяє коректно передавати контекст Flask-запиту у фоновий потік.

В якості базової апаратної платформи, на якій виконано розгортання зазначених вище модулів, стеку технологій та розроблених програмних компонент, використано мікрокомп'ютерну платформу `Raspberry Pi 4`, що відповідає вимогам реалізації туманної архітектури.

Отже, кожен із використаних модулів і бібліотек виконує чітко визначену роль: `Flask` формує основу вебсервісу, `Flask-WTF` та `WTForms` реалізують взаємодію з користувачем, `SQLAlchemy` керує даними, `scikit-learn` і `joblib`

забезпечують відтворюваність машинного навчання, eli5 формує пояснення, Plotly генерує візуалізацію, а smtplib інтегрує результати із зовнішніми поштовими сервісами. Сукупне застосування цих компонентів забезпечує узгоджену архітектуру, що поєднує вебдоступ, інтелектуальну аналітику та пояснюваність результатів у єдиному середовищі.

4.2 Архітектура програмного забезпечення та її основні функціональні сценарії

Під час досліджень із проєктування комп'ютерної технології, що реалізує архітектурну взаємодію запропонованих програмно-алгоритмічних рішень, було розроблено та імплементовано серверну архітектуру класичного вебзастосунку, побудовану на базі фреймворка Flask [170, 171], як показано на рисунку 4.1. Архітектурно це відповідає підходу «edge ML as a service» [172]: запити з клієнтських пристроїв опрацьовуються сервером, який виконує локальний виклик моделі та повертає результати.

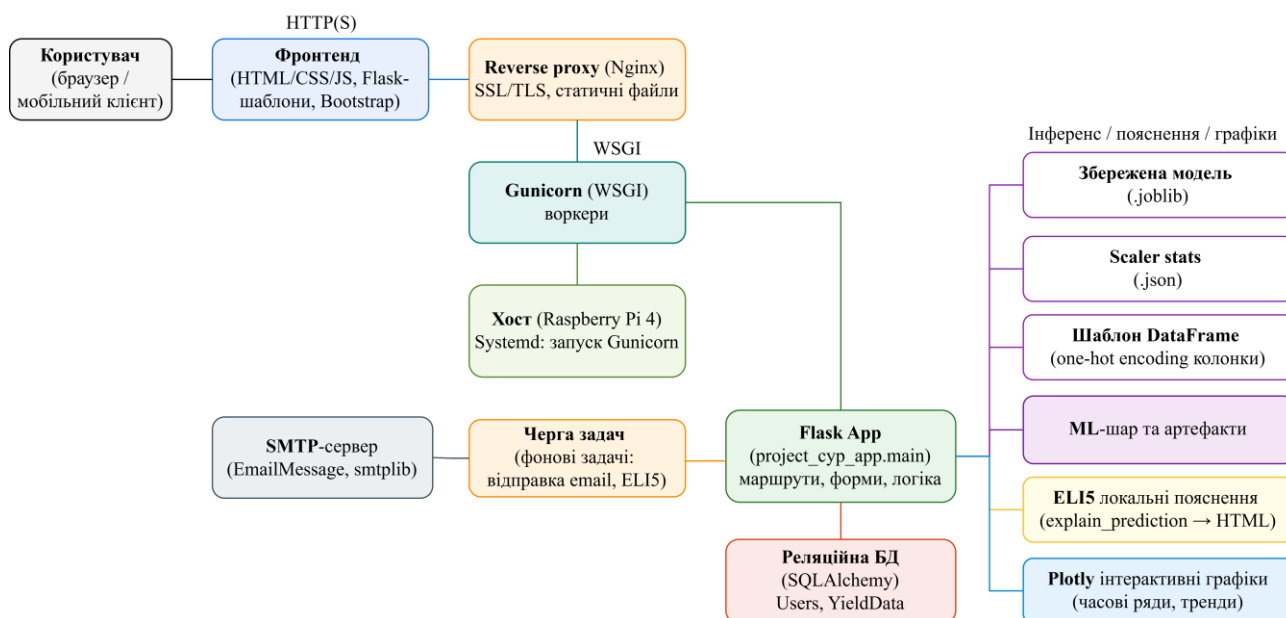


Рисунок 4.1 – Архітектура програмно-апаратного забезпечення комп'ютерної технології прогнозування врожайності сільськогосподарських культур

Архітектура програмно-апаратного забезпечення розроблюваної комп'ютерної технології прогнозування врожайності сільськогосподарських культур, яку наведено на рисунку 4.1, реалізована у вигляді багаторівневої модульної структури, яка забезпечує цілісний цикл обробки агрокліматичних даних: введення користувачем даних, формування прогнозу, пояснення моделей, інформування користувача та візуалізація результатів.

На верхньому рівні функціонує користувацький інтерфейс (див. додаток Г.2), побудований на основі Flask-WTF та Bootstrap, що забезпечує введення вихідних параметрів агромоніторингу (рік, країна, вид агрокультури, середньорічна температура, кількість опадів та обсяг використаних пестицидів). Дані уніфіковано надходять до шару Flask-сервера (див. додаток Г.1), який виконує маршрутизацію запитів, управління сесіями користувачів і взаємодію з іншими компонентами.

Наступним рівнем є шар обробки даних, що включає процедури попередньої підготовки: стандартизацію числових змінних за допомогою збережених статистичних дескрипторів, відновлення структури набору даних з урахуванням попереднього one-hot кодування категоріальних ознак, а також перевірку узгодженості вхідних даних із форматом навчальних/тестових вибірок. Це забезпечує відтворюваність результатів і коректність застосування розроблених ML моделей.

Центральним компонентом є ML-шар, де виконується безпосереднє застосування навченої моделі на основі алгоритмів машинного навчання для прогнозування врожайності. Для цього використовуються заздалегідь збережені артефакти: серіалізована модель у форматі .joblib, статистичні параметри масштабування та набір даних простору ознак. Сукупність цих артефактів гарантує стабільність та узгодженість процесу інференсу моделі.

Після отримання прогнозу викликається блок пояснюваного штучного інтелекту (XAI), що реалізує інтерпретацію прогнозних моделей на рівні окремих ознак. Для цього застосовується інструмент ELI5, який дозволяє кількісно оцінити внесок змінних у фінальний прогноз та надати користувачу пояснення отриманих

модельних даних. Додатково використовується підсистема візуалізації на основі бібліотеки Plotly, яка генерує інтерактивні графіки, що відображають історичні дані, збережені у БД на основі SQLAlchemy [173].

Заключним елементом архітектури є шар інтеграції із зовнішніми сервісами, який забезпечує автоматизоване надсилання результатів користувачу засобами електронної пошти через SMTP-протокол. До повідомлення інтегруються як числові прогнози, так і HTML-пояснення згенеровані ELI5, що підвищує практичну цінність та зручність використання розроблюваної комп'ютерної технології (див. приклад у додатку Г.4).

Отже, запропонована архітектура поєднує вебсервісну логіку, аналітичний шар машинного навчання, інструменти інтерпретації моделей та механізми взаємодії з користувачем, утворюючи завершену технологічну організацію інтелектуалізованої обробки даних щодо прогнозування врожайності агрокультур [174]. Основні функціональні сценарії взаємодії користувача з системою (Use-cases) наведено на рисунку 4.2.

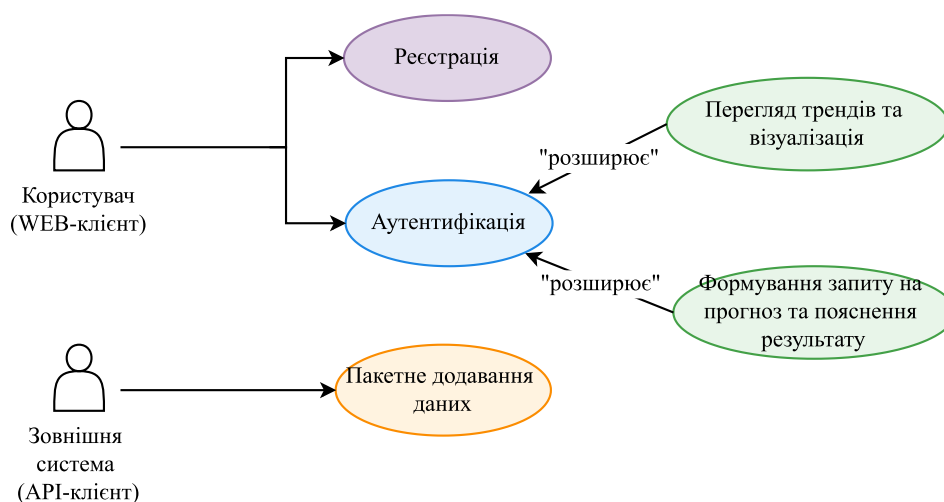


Рисунок 4.2 – Діаграма прецедентів

Нижче подано формальні описи основних сценаріїв використання (виконавці: неавторизований користувач; авторизований користувач; зовнішня система з API-ключем).

1. Прецедент «Реєстрація нового користувача».

Виконавець: неавторизований користувач.

Передумови: відкрита сторінка вебзастосунку.

Основний потік: маршрут /register (POST/GET) → заповнення форми реєстрації → валідація (хешування пароля через werkzeug.security.generate_password_hash методом PBKDF2-SHA256) → створення запису в таблиці users → автоматичний логін → перенаправлення до сторінки прогнозів.

Постумови: можливе подальше виконання прецедентів «Перегляд трендів та візуалізація» та «Формування запиту на прогноз та пояснення».

Альтернативні потоки: якщо користувач із введеною електронною поштою вже існує в БД, то виконується автоматичне перенаправлення до сторінки авторизації.

2. Прецедент «Аутентифікація (Авторизація)».

Виконавець: зареєстрований користувач.

Основний потік: маршрут /login → введення адреси електронної пошти і пароля → перевірка існування запису → порівняння хеша пароля (check_password_hash) → виклик login_user.

Постумови: можливе подальше виконання прецедентів «Перегляд трендів та візуалізація» та «Формування запиту на прогноз та пояснення результату».

Альтернативні потоки: у разі помилок при введенні даних, повідомлення flash.

3. Прецедент «Формування запиту на прогноз та пояснення результату».

Виконавець: авторизований користувач.

Основний потік: маршрут /predictions → користувач заповнює форму CreateCYPForm (поля: area, year, average_rain_fall_mm_per_year, pesticides_tonnes, avg_temp, item), форма визначена у forms.py. Сервер формує структуру вхідних ознак: викликає recreate_df(new_submission), копіює шаблонний рядок DF_STRUCT і підставляє значення (створює one-hot ознаки country_* та item_* для відповідних значень). Виконується масштабування числових полів через функцію scale_df(), StandardScaler відновлюється зі збережених статистичних метрик (scaler_stats.json). Прогноз: loaded_model.predict(scaled_df), модель

завантажена через `joblib.load`. Інтерпретація: `eli5.explain_prediction(loader_model, scaled_df.values[0], feature_names=...)` → `eli5.format_as_html(exp)` → HTML-пояснення. Доставка результату: у фоні (`thread`) викликається `notify_via_email(form, y_pred, html)` → побудова `EmailMessage` з plain-text та HTML-альтернативою → відправка через SMTP.

Постумови: автоматичне перенаправлення користувача на сторінку `secrets.html`, де сказано, що результати прогнозу та пояснення незабаром будуть надіслані йому на електронну пошту, вказану під час реєстрації.

Альтернативні потоки: помилка при введенні некоректних даних, наприклад, с/г культура та країна введені такі, що в процесі навчання моделі вони були відсутні в наборі даних.

4. Прецедент «Перегляд трендів та візуалізація».

Виконавець: авторизований користувач.

Основний потік: маршрут `/trends` → введення країни і культури у форму `CreateTrendQueryForm` → БД-запит `YieldData` → конструювання графіка `Plotly` (`traces: temp, rainfall, pesticides, yield`) → вбудовування HTML із графіка у шаблон.

Постумови: графіки успішно відображені на сторінці користувача з можливістю інтерактивної взаємодії завдяки інструментам бібліотеки `Plotly`.

Альтернативні потоки: помилка при введенні некоректних даних, наприклад, с/г культура та країна введені такі, що відсутні в БД.

5. Прецедент «Пакетне додавання даних».

Виконавець: зовнішня система (API-клієнт).

Основний потік: POST JSON на `/add_via_json` з полем `sup_data` → маршрут захищено методом `api_key_required` (перевірка X-API-Key заголовка) → створення екземплярів `YieldData`

Постумови: завершення операції та збереження змін у БД → повернення JSON із повідомленням про успіх виконання операції.

Альтернативні потоки: повернення JSON із повідомленням про помилку при обробці даних або невірний/відсутній API-Key.

На підставі наведеного вище опису було синтезовано схему послідовності

поступової обробки запиту /predictions (основний сценарій), як показано на рисунку 4.3. Послідовність обробки користувацького запиту в цій архітектурі реалізована як багатокроковий конвеєр взаємодії між основними компонентами розроблюваної комп'ютерної технології, що забезпечує узгодженість етапів введення, трансформації, прогнозування та інтерпретації [174].

На початковому етапі авторизований користувач через вебінтерфейс здійснює введення параметрів агромоніторингу, зокрема країни вирощування, року, виду агрокультури, середньорічної температури, кількості опадів та обсягів застосованих пестицидів та надсилає POST запит на /predictions із полями заповненої форми (CreateCYPForm). Дані передаються на Flask-сервер у вигляді HTTP-запиту. Сервер виконує первинну маршрутизацію, валідацію коректності введених значень полів через WTForms (включно з CSRF) та ініціює подальший цикл обробки.

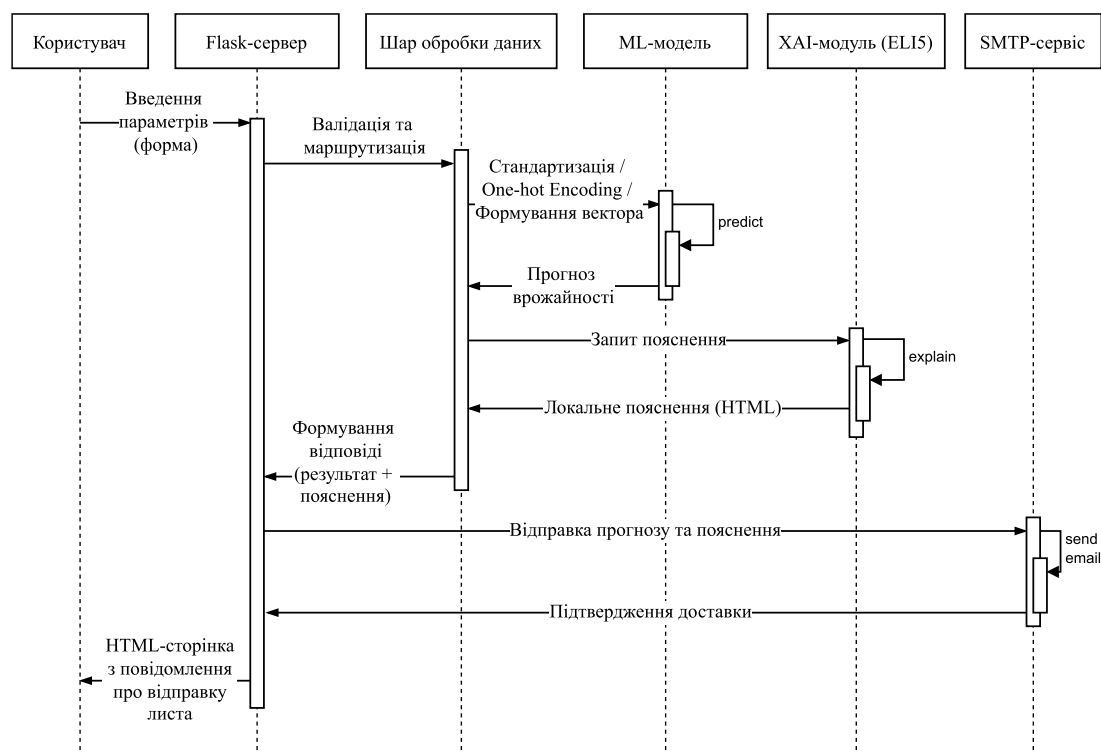


Рисунок 4.3 – Діаграма потоків для прецеденту «Формування запиту на прогноз та пояснення результату»

Далі дані спрямовуються у шар попередньої обробки, де реалізуються

процедури нормалізації, стандартизації та кодування ознак. Для числових параметрів застосовуються масштабувальні перетворення із використанням збережених статистичних характеристик навчальної вибірки, що забезпечує відтворюваність процесу інференсу. Категоріальні змінні відновлюються у вигляді one-hot представлення згідно зі структурою набору даних, сформованою на етапі навчання моделей. У результаті формується узгоджений вектор ознак, який передається на вхід моделі машинного навчання.

У ML шарі виконується безпосереднє прогнозування врожайності. Навчена модель, завантажена із серіалізованого артефакту, обчислює значення очікуваної врожайності агрокультури у відповідних одиницях, `loaded_model.predict(scaled_df)` повертає прогноз `y_pred`. Отриманий результат повертається до шару обробки даних, де він зберігається для подальшої інтерпретації та візуалізації.

З метою забезпечення прозорості роботи системної організації розроблюваної комп'ютерної технології, сформований вектор ознак додатково подається до модулів пояснюваного штучного інтелекту. Інструмент `eli5.explain_prediction(...)` визначає вагу окремих змінних у процесі прийняття рішення та формує інтерпретацію прогнозних моделей у вигляді таблиць та графічних HTML-звітів. Це дозволяє ідентифікувати найсуттєвіші фактори, які зумовили прогнозовану врожайність.

Заключний етап передбачає формування комплексної відповіді для користувача. Flask-сервер отримує прогнозні значення врожайності, пояснення від XAI-модуля та додаткові графічні візуалізації, після чого повертає інтегрований HTML-документ як вебсторінку. Паралельно формується електронний лист, що включає числові результати та пояснення у форматі HTML, який автоматично відправляється на адресу користувача через SMTP-сервіс.

Таким чином, запропонована послідовність обробки запиту забезпечує наскрізний рух даних від моменту введення до отримання користувачем прогнозу з поясненнями, при цьому інтегруючи процедури форматування, обробки, прогнозування та інтерпретації у єдиний системний логічний процес.

4.3 Результати розробки інтерфейсу користувача

Вебінтерфейс розробленої системної організації комп'ютерної технології прогнозування врожайності реалізований як частина Flask-застосунку, що забезпечує взаємодію між користувачем і серверною логікою (див. додаток Г.2). HTML-сторінки формуються із використанням шаблонізатора Jinja2 [175], що дає можливість динамічно відображати дані та візуалізації в залежності від контексту користувацького сеансу [170, 176]. Результати розробки інтерфейсу користувача у вигляді основних HTML-сторінок наведено на рисунках 4.4–4.7.

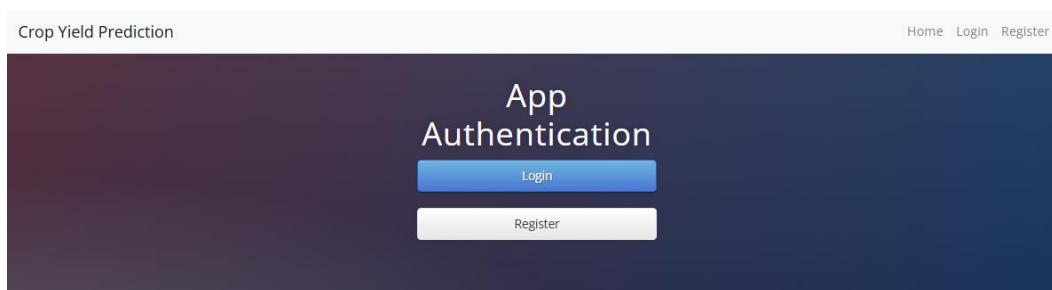


Рисунок 4.4 – HTML-сторінка для route(“/home”)

The screenshot shows the 'Crop Yield Prediction' form page. The page has a dark blue gradient background. At the top, there is a navigation bar with 'Home', 'Trends', 'Predictions', and 'Log Out' links. The main heading is 'Please Fill in the Details Below to Get Your Crop Yield Prediction'. Below the heading, there are several input fields with labels: 'Country' (with 'Ukraine' entered), 'Year' (with '2025' entered), 'Average rain fall mm per year' (with '600' entered), 'Pesticides tonnes' (with '86781.8' entered), 'Average temperature' (with '12.1' entered), and 'Crop name' (with 'Maize' entered). At the bottom, there is a blue button labeled 'Email me with prediction'.

Рисунок 4.5 – HTML-сторінка для route(“/predictions”)

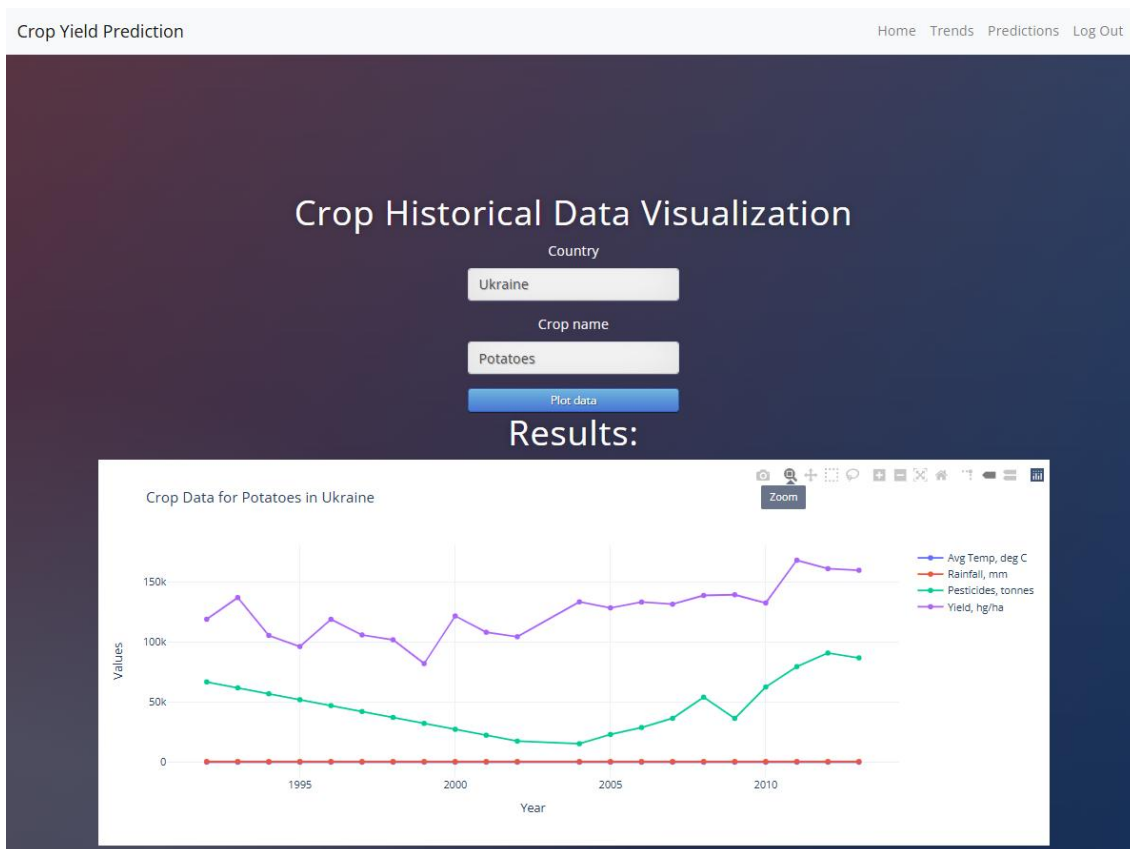


Рисунок 4.6 – HTML-сторінка для route(“/trends”)

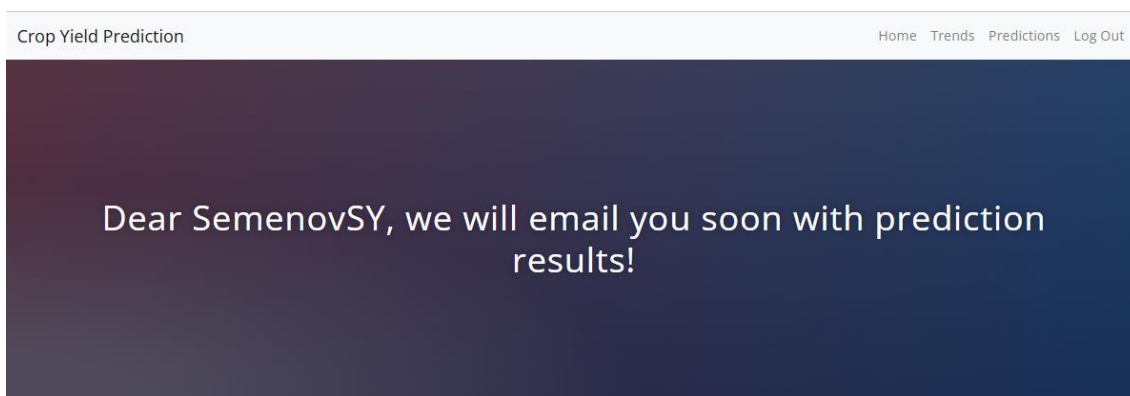


Рисунок 4.7 – HTML-сторінка для route(“/secrets”)

Інтерфейс головної сторінки (/home), яку наведено на рисунку 4.4, реалізує механізм аутентифікації та реєстрації користувачів. Використання шаблонів Jinja дозволяє інтегрувати Flask-WTF форми для вводу облікових даних та забезпечує зв'язок між клієнтською та серверною частинами. Інтерфейс містить два основні варіанти дій: вхід до вже зареєстрованого облікового запису (Login) або створення нового облікового запису (Register). У разі реєстрації створюється новий

користувачський профіль у БД, тоді як при вході здійснюється перевірка валідності введених облікових даних. Це відповідає стандартній архітектурі Flask із застосуванням сесій для збереження стану користувача та відповідає вимогам безпеки та персоналізації, оскільки подальші результати прогнозів та візуалізацій пов'язуються із конкретним користувачем.

Основний функціональний модуль вебзастосунку представлений формою введення параметрів агромоніторингу (країна, рік, середньорічна кількість опадів, використання пестицидів, середньорічна температура та назва агрокультури) на сторінці формування запиту прогнозу (/predictions), як показано на рисунку 4.5. Поля форми інтегровані у HTML-шаблон із використанням Jinja-тегів для відображення змінних, валідації введених значень та їх передачі до Flask-контролера. Після відправки форми дані надходять на сервер, де відбувається масштабування, підготовка та передача до прогнозової моделі на основі алгоритмів машинного навчання. Результати прогнозування ML моделлю та пояснення від XAI (ELI5) передаються до SMTP-модуля для надсилання користувачеві електронного листа (див. рис. 4.7 та додаток Г.4).

Сторінка аналізу історичних даних (/trends), яку наведено на рисунку 4.6, реалізує візуалізацію часових рядів агрокліматичних показників та врожайності. Користувач обирає країну та агрокультуру за допомогою інтегрованої Flask-WTF форми, після чого сервер здійснює обробку даних і передає їх у Jinja-шаблон. Відображення здійснюється через інтерактивні графіки, побудовані на базі бібліотеки візуалізації Plotly, що інтегрується у HTML за допомогою Jinja. Таким чином, графічний компонент відображає динаміку середньорічної температури, кількості опадів, застосування пестицидів та врожайності, що дозволяє користувачу проводити детальний порівняльний аналіз.

Усі сторінки вебзастосунку дотримуються принципів Model-View-Controller (MVC) [177] у Flask або близькі до нього: HTML-шаблони формуються динамічно через Jinja, серверні потоки обробляють основну логіку, а модельні об'єкти відповідають за машинне навчання та взаємодію з БД. Таким чином, застосований у процесі розробки підхід забезпечує відокремлення рівнів вебзастосунку,

гнучкість у масштабуванні та розширенні функціоналу.

Під час розробки інтерфейсу для взаємодії з серверною частиною комп'ютерної технології було реалізовано спеціалізований API-ендпоінт `/add_via_json`, призначений для автоматизації процесу масового завантаження структурованих даних щодо врожайності с/г культур. Така точка доступу забезпечує ефективне збереження значних обсягів агрономічних даних через RESTful інтерфейс, що дозволяє інтегрувати запропоновану реалізацію комп'ютерної технології із зовнішніми джерелами інформації та автоматизувати процеси збору статистичних даних.

Розроблений механізм підтримує авторизацію через API-Key (див. рис. 4.8) та транзакційну обробку множинних записів, гарантуючи цілісність даних під час виконання операцій масового додавання (див. рис. 4.9). Використання JSON як формату обміну даними забезпечує стандартизацію та сумісність із сучасними вебтехнологіями. Опис API-ендпоінту `/add_via_json` наведено в таблиці 4.1, а структуру вхідних даних – у таблиці 4.2.

Таблиця 4.1 – Опис API-ендпоінту `/add_via_json`

Параметр	Значення
URL	<code>/add_via_json</code>
HTTP-метод	POST
Авторизація	Обов'язкова (декоратор <code>@api_key_required</code>)
Тип контенту	<code>application/json</code>

Таблиця 4.2 – Структура вхідних даних API-ендпоінту `/add_via_json`

Поле	Тип	Опис
<code>area</code>	string	Географічна область/регіон
<code>year</code>	integer	Рік спостереження
<code>average_rain_fall_mm_per_year</code>	float	Середня річна кількість опадів (мм)
<code>pesticides_tonnes</code>	float	Використання пестицидів (тонни)
<code>avg_temp</code>	float	Середня річна температура (°C)
<code>item</code>	string	Назва сільськогосподарської культури
<code>value_hg_ha</code>	float	Врожайність (hg/ha)

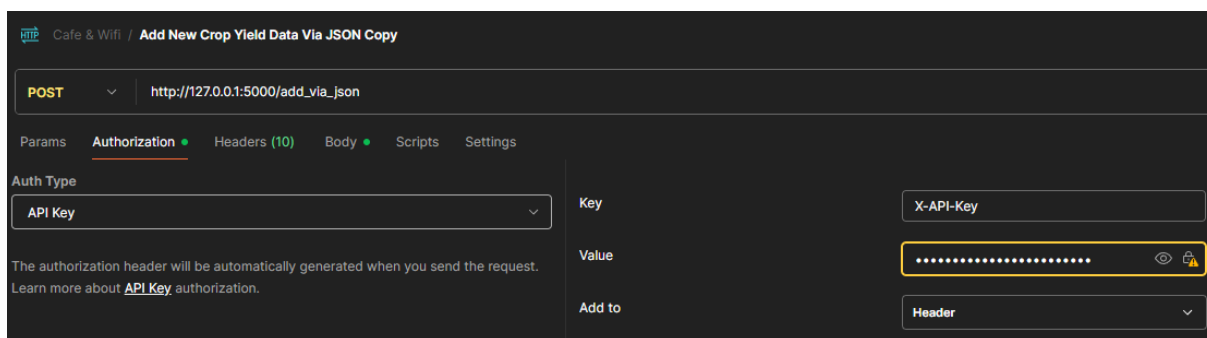


Рисунок 4.8 – API-ендпоінт /add_via_json Authorization в Postman

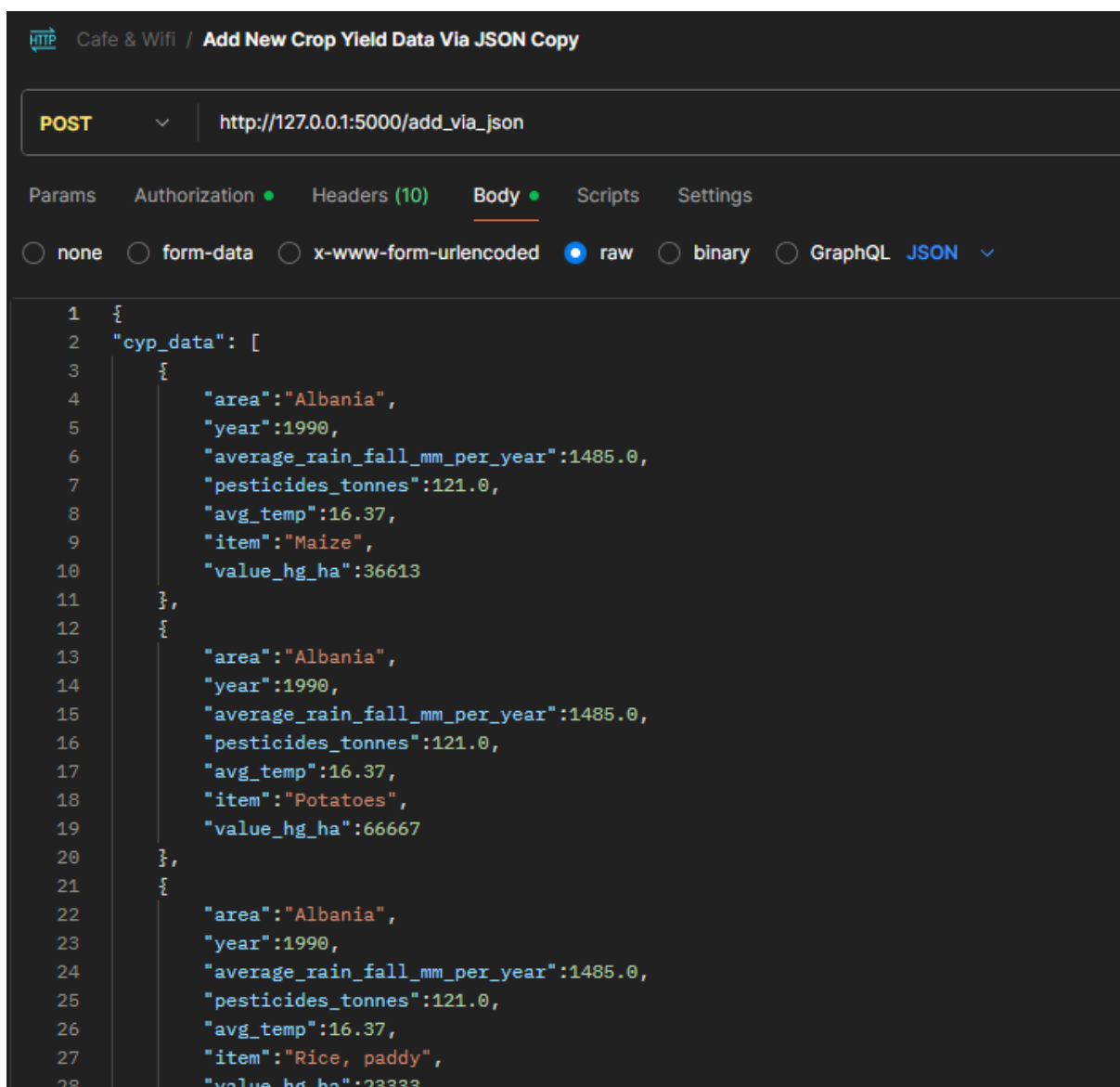


Рисунок 4.9 – API-ендпоінт /add_via_json, тіло запиту в Postman

Точка доступу API-ендпоінт /add_via_json призначена як для масового, так і одиничного додавання записів щодо врожайності с/г культур до БД через JSON-

запит. Ця точка доступу очікує JSON-об'єкт із ключем `cyp_data`, що містить масив об'єктів із полями (див. табл. 4.2). Успішна відповідь має вид `{ "response": { "success": "Successfully added cyp_data from json." } }`. Цей API-ендпоінт використовує модель `YieldData` з ORM `Flask-SQLAlchemy` для збереження інформації щодо врожайності агрокультур із урахуванням кліматичних та агротехнічних факторів (див. додаток Г.1).

Реалізований API-ендпоінт `/add_via_json` демонструє ефективний підхід до масового завантаження структурованих даних у розподіленій організації розробленої комп'ютерної технології прогнозування врожайності агрокультур. Основними перевагами розробленого рішення є: підтримка одночасної обробки множинних записів через єдиний HTTP-запит; обов'язкова авторизація через API-ключ запобігає несанкціонованому доступу; JSON-формат забезпечує сумісність із широким спектром клієнтських застосунків [170].

Таким чином, розроблений API-ендпоінт створює надійну основу для автоматизації процесів збору та збереження агрономічних даних, що є важливим для функціонування вебзастосунку прогнозування врожайності агрокультур. Запропоноване архітектурне рішення комп'ютерної технології дозволяє розширювати функціональність та інтегрувати додаткові джерела даних без суттєвих модифікацій існуючого програмного коду.

4.4 Аналіз обмежень реалізації та перспективи подальших досліджень

Під час розробки та валідації запропонованої комп'ютерної технології, що реалізована у вигляді вебзастосунку прогнозування врожайності сільськогосподарських культур, було визначено певні потенційні обмеження, пов'язані з архітектурними та алгоритмічними аспектами. Основним джерелом вразливостей виступає компонент автентифікації та управління сесіями, що у випадку неналежної реалізації може створювати загрозу несанкціонованого доступу до персоніфікованих результатів. Отже, під час переходу від запропонованої в цій роботі реалізації до промислового впровадження

рекомендується використовувати механізми на основі проксі-серверних конфігурацій, наприклад, Nginx як reverse proxy, або статичні файлові структури. Аналогічно, використання SMTP-сервісу для відправлення електронних листів вимагає ретельного дотримання політик безпеки, оскільки перехоплення облікових даних може призвести до витоку чутливої інформації. Наразі SMTP пароль та інші секрети зчитуються з .env, що є достатньо коректним підходом, але в процесі подальших досліджень слід поглибити аспекти реалізації механізмів керування доступом до файлів конфігурації та обмеження прав на виконання. Важливим є також ризик некоректної валідації даних на стороні користувача та/або сервера, що у разі цілеспрямованих атак може спровокувати SQL-ін'єкції або спотворення результатів прогнозу. Щодо API/add_via_json, його захищено заголовком X-API-Key, але заголовок у передачі може бути перехоплений. Отже, під час подальшого розвитку запропонованої комп'ютерної технології доречним є застосування HTTPS (TLS). Крім цього, перспективним є використання цифрових токенів із терміном дії (JWT) або HMAC підписів запиту, вести журнал доступів і логувати IP та час викликів.

З апаратно-технічної точки зору основним обмеженням запропонованої реалізації комп'ютерної технології є залежність від обчислювальних ресурсів мікрокомп'ютерної платформи Raspberry Pi 4, що під час динамічного збільшення обсягів оброблюваних даних може ускладнювати виконання ресурсомістких алгоритмів машинного навчання та побудову складних візуалізацій у реальному часі. Наявність висококорельованих ознак та значні обсяги шуму у вхідних даних без належної попередньої обробки можуть призвести до створення ризику перенавчання окремих моделей, що може знизити достовірність прогнозів. Щодо програмної реалізації засобів ХАІ, то генерування локального пояснення може бути обчислювально інтенсивною процедурою. На Raspberry Pi або в обмеженому середовищі це може уповільнити обробку. Тому перспективним напрямком пріоритетних досліджень під час масштабування одержаних результатів є реалізація механізмів асинхронних черг та/або кешування результатів для ідентичних вхідних даних.

Таким чином, для мінімізації зазначених обмежень та потенційних ризиків доцільним є застосування сучасних методів забезпечення інформаційної безпеки, включаючи використання захищених протоколів автентифікації, двофакторної перевірки та шифрування трафіку. Доцільним напрямом є інтеграція надійної системи логування та моніторингу, що дозволить своєчасно виявляти аномальні дії та відхилення у роботі сервісу. З архітектурної точки зору перспективним є поступовий перехід від монолітної структури Flask-застосунку до мікросервісної архітектури, що підвищить показники масштабованості й надійності реалізованої комп'ютерної технології під час її застосування в реальних практичних сценаріях. Додатково можна зазначити, що використання механізмів контейнеризації та оркестрації потенційно дозволить стандартизувати процеси розгортання і спростить управління програмним середовищем.

Із позицій подальших перспективних напрямків наукових досліджень доцільно розвивати модуль інтелектуальної аналітики шляхом інтеграції ансамблевих і гібридних моделей, що поєднують різні алгоритмічні підходи, як наприклад, Random Forest, Gradient Boosting, XGBoost та нейронні мережі. Перспективним також є впровадження методів онлайн-навчання прогнозних моделей, які дозволять системній реалізації комп'ютерної технології адаптивно оновлювати параметри моделей на основі потокових даних агромоніторингу. Важливим науковим завданням у перспективі є також розширення функціоналу програмного модуля на основі алгоритмів пояснювального штучного інтелекту за рахунок побудови глобальних моделей-метаінтерпретаторів, що здатні узагальнювати закономірності для різних культур, регіонів і часових інтервалів.

У ширшій перспективі доцільно поєднувати прогнозування врожайності з оптимізаційними моделями управління агротехнічними процесами, включаючи вибір систем зрошення, норм внесення добрив та пестицидів, а також планування сівозмін. Такий підхід дозволить створити основу для побудови комплексних систем підтримки прийняття рішень у сільському господарстві, що інтегрують програмні засоби аналітики і прогнозування, а також рекомендаційні модулі в єдину програмно-апаратну платформу.

4.5 Висновки за четвертим розділом

У цьому розділі дисертації розв'язано основні дослідні завдання, які полягали в створенні та валідації комп'ютерної технології у вигляді вебзастосунку, що в межах єдиної архітектури реалізує комплексну трансформацію результатів агромоніторингу, а також критичному аналізу отриманих результатів щодо розробки та дослідження програмно-алгоритмічного забезпечення агротехнічного призначення прогнозування врожайності сільськогосподарських культур із формулюванням перспектив подальших досліджень. Під час виконання досліджень одержано такі результати:

1. Розроблено архітектуру Flask-застосунку, яка включає інтеграцію модулів автентифікації, обробки форм введення, обчислювального шару для машинного навчання та підсистеми візуалізації. Застосунок розгортається на мікрокомп'ютерній платформі Raspberry Pi 4 та реалізує чітко структуровану взаємодію між користувачем, серверними обчислювальними модулями та базовими сервісами, що забезпечує повний цикл обробки даних агромоніторингу під час прогнозування врожайності агрокультур у широкому діапазоні варіювання просторово-часових ознак.

2. Проведено детальний аналіз ролі імпортованих бібліотек та програмних компонентів, зокрема Flask, Flask-WTF, SQLAlchemy, Pandas, Scikit-learn та Eli5. Доведено, що їх системне поєднане використання забезпечує комплексне середовище для інтеграції баз даних, обробки вхідних даних, реалізації моделей машинного навчання та інтерпретації прогнозних моделей на основі пояснювального штучного інтелекту.

3. Описано основні функціональні сценарії взаємодії користувача із комп'ютерною технологією, що охоплюють етапи автентифікації, введення параметрів агромоніторингу, отримання прогнозів врожайності та перегляду історичних даних. Встановлено, що реалізація цих сценаріїв забезпечує цілісну користувацьку взаємодію з вебзастосунком, що орієнтований на практичне використання в умовах агропідприємств рослинництва відкритого ґрунту.

4. Досліджено послідовність обробки системних і користувацьких запитів, що охоплюють надсилання даних через інтерфейс, їх обробку із застосуванням форм Flask-WTF, масштабування та перетворення, обчислення прогнозу ML-моделлю та формування відповіді із прогнозом та поясненням предикативних моделей. Такий підхід підтвердив коректність архітектурного рішення розробленого програмно-алгоритмічного забезпечення агротехнічного призначення та узгодженість взаємодії між усіма його компонентами.

5. Виконано критичний аналіз обмежень і потенційних вразливостей розробленої архітектури програмно-алгоритмічного забезпечення, реалізованої у вигляді вебзастосунку. Серед виявлених обмежень і ризиків основними є ті, що пов'язані з автентифікацією, безпечністю SMTP-взаємодії, а також можливим перенавантаженням апаратних ресурсів Raspberry Pi. Запропоновано практичні шляхи мінімізації цих ризиків, включаючи механізми двофакторної автентифікації, контейнеризації, оркестрації та впровадження підходів до онлайн-навчання прогнозних моделей.

6. Встановлено, що запропонована архітектура є потенційно придатною до масштабування та подальшої інтеграції з реальними комплексами агромоніторингу, в тому числі завдяки застосуванню механізмів API, які реалізовано в досліджуваному програмно-алгоритмічному забезпеченні вебзастосунку. Перспективними напрямками розвитку визначено впровадження ансамблевих та гібридних ML-моделей для підвищення точності прогнозування, удосконалення моделей на основі пояснювального штучного інтелекту для глобальних інтерпретацій та поєднання прогнозів із оптимізаційними моделями управління агротехнічними процесами.

Таким чином, запропонована комп'ютерна технологія, що реалізована у вигляді відповідного вебзастосунку, може розглядатися як науково обґрунтоване і практично значуще програмно-алгоритмічне забезпечення комплексів агромоніторингу, що забезпечує інтегровану обробку агрокліматичних даних та їх трансформацію у прогностичні та інтерпретовані результати, придатні для використання в агротехнічній практиці.

ВИСНОВКИ

У цій дисертації розв'язано актуальну наукову задачу практично-орієнтованого характеру, яка полягала в розвитку інформаційних і комп'ютерних технологій агротехнічного призначення шляхом розробки, дослідження та застосування програмно-алгоритмічного забезпечення комплексної інтелектуалізованої трансформації даних агромоніторингу на основі туманної архітектури, що сфокусоване на прогнозуванні врожайності сільськогосподарських культур із обліком значної мінливості агрокліматичних умов, типів вирощуваних агрокультур та застосовуваних практик їх захисту. Це дозволило довести доцільність реалізації підходів із підвищення продуктивності агротехнологічних практик рослинництва відкритого ґрунту завдяки підвищенню рівня цифровізації агротехнічних процесів, зокрема шляхом застосування запропонованих програмних засобів предикативної аналітики агрокліматичних факторів, що впливають на врожайність вирощуваних сільськогосподарських культур, із автоматизованою підтримкою прийняття рішень. Основні науково-прикладні результати цієї дисертаційної роботи є такими:

1. На підставі аналізу сучасного стану програмно-алгоритмічних рішень із цифровізації та інтелектуалізації рослинництва відкритого ґрунту встановлено, що інформаційно-орієнтована модернізація сільського господарства є об'єктивним трендом сталого розвитку в умовах зростання ризиків природного і антропогенного характеру. Локалізовані обмеження наявних науково-практичних рішень дозволили довести, що існуючі підходи до прогнозування врожайності, попри їх значні функціональні можливості, потребують подальшого розвитку в контексті реалізації засад повноциклової трансформації, основні етапи якої мають охоплювати агрегування просторово-часових даних, їх структуризацію, статистичну обробку, предикативний аналіз та інтерпретацію з автоматизованою підтримкою прийняття рішень із використанням технологій туманних обчислень, машинного навчання і пояснювального штучного інтелекту. Також обґрунтовано, що задля підвищення продуктивності обчислювально-аналітичних процесів ці

трансформаційні етапи мають бути об'єктивно розподілені за структурними рівнями досліджуваного програмно-алгоритмічного забезпечення на основі децентралізованої архітектури та реалізовані з комплексним урахуванням кліматичних умов, типів агрокультур і застосовуваних агротехнічних практик.

2. У результаті формалізації процесу комплексної інтелектуалізованої трансформації часово-просторових даних агромоніторингу розроблено методологічну й модельну основу для побудови програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур. Обґрунтовано структурно-функціональне забезпечення туманної архітектури багаторівневої трансформації даних, що дозволяє реалізувати децентралізований підхід із логікою на основі запитів щодо ініціювання комплексу процедур збору, агрегації й обчислювально-аналітичного опрацювання даних в умовах мінливості агрокліматичних і агротехнологічних факторів. Запропонована структурно-функціональна модель дозволила формалізувати кросрівневу взаємодію польових пристроїв, програмних засобів рівня туманних обчислень та IoT-сервера з урахуванням варіативності умов вирощування агрокультур. Синтезовані структурно-функціональна та інформаційна моделі, а також запропоновані сценарії прикладного застосування розробки, дозволили локалізувати наскрізну логіку і послідовність інформаційних перетворень на всіх архітектурних рівнях програмно-алгоритмічного забезпечення під час прогнозування врожайності сільськогосподарських культур.

3. Розроблено та доведено ефективність програмно-алгоритмічного забезпечення повноциклової обробки даних агромоніторингу, яке включає попередню обробку даних, розвідувальний аналіз, інтелектуальну аналітику на основі моделей машинного навчання, а також інтерпретацію предикативних моделей на основі пояснювального штучного інтелекту. Доведено, що ансамблеві алгоритми, зокрема Bagging Regressor та Random Forest, забезпечують найвищий коефіцієнт детермінації серед досліджуваних алгоритмів при одночасному зниженні похибки прогнозування. Застосування програмно реалізованих методів відбору ознак дозволило зменшити розмірність вхідних даних без істотної втрати точності, підтвердивши домінантний вплив кліматичних параметрів, показників

застосування пестицидів і просторових ознак на формування врожайності.

4. Засобами комп'ютерного експерименту доведено доцільність інтеграції методів пояснювального штучного інтелекту до структури запропонованого програмно-алгоритмічного забезпечення, що забезпечило підвищення інтерпретованості та практичної цінності реалізованих моделей прогнозування врожайності агрокультур. Комбіноване використання глобальних методів оцінки важливості ознак та локальних ХАІ-методів продемонструвало узгоджені результати щодо визначення домінантних факторів урожайності, зокрема для окремих агрокультур і регіонів, а також дозволило кількісно оцінити вплив ґрунтокліматичних змінних. Це підтверджує можливість використання розробленого програмно-алгоритмічного забезпечення як функціональної складової цифрових систем прийняття рішень у сільському господарстві, орієнтованих на адаптацію агротехнічних заходів до мінливих умов вирощування.

5. Створено комп'ютерну технологію у вигляді вебзастосунку, що реалізує взаємодію між користувачем, серверними обчислювальними модулями, базовими сервісами та розробленими програмно-алгоритмічними компонентами прогнозування врожайності агрокультур. Результати тестування та валідації продемонстрували стабільність обчислювально-аналітичних процесів і відтворюваність результатів прогнозування врожайності агрокультур за умов варіативності агрокліматичних параметрів і практик захисту рослин.

6. Критичний аналіз результатів досліджень підтвердив доцільність застосування запропонованого програмно-алгоритмічного забезпечення під час прогнозування врожайності агрокультур, а також його потенційну сумісність із комплексами агромоніторингу, в тому числі завдяки реалізації механізмів API. Разом із тим локалізовано ряд обмежень розробленого програмно-алгоритмічного забезпечення і перспективних напрямків досліджень із мінімізації їх впливу, зокрема в напрямі інтеграції вискоєфективних механізмів кіберзахисту, розробки та застосування підходів до онлайн-навчання прогнозних моделей, а також удосконалення інтелектуальних моделей глобальних інтерпретацій та поєднання прогнозів із оптимізаційними моделями управління агротехнічними процесами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. National Centers for Environmental Information: Global Climate Report. URL: <https://www.ncei.noaa.gov/access/monitoring/monthly-report/global/202113> (дата звернення 05.01.2026).

2. World Economic Forum: Our resources are running out. These charts show how urgently action is needed. URL: <https://www.weforum.org/stories/2024/03/sustainable-resource-consumption-urgent-un/> (дата звернення 05.01.2026).

3. United Nations: Global Land Outlook 2. URL: <https://www.unccd.int/resources/global-land-outlook/glo2> (дата звернення 04.01.2026).

4. Worldometer: Current World Population. URL: <https://www.worldometers.info/world-population/> (дата звернення 04.01.2026).

5. World Health Organization: Hunger numbers stubbornly high for three consecutive years as global crises deepen: UN report. URL: <https://www.who.int/news/item/24-07-2024-hunger-numbers-stubbornly-high-for-three-consecutive-years-as-global-crises-deepen--un-report> (дата звернення 06.01.2026).

6. European Commission: CAP in my country. URL: https://agriculture.ec.europa.eu/cap-my-country_en (дата звернення 06.01.2026).

7. Eurostat: Performance of the agricultural sector. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Performance_of_the_agricultural_sector (дата звернення 08.01.2026).

8. Statista: The Value of Agriculture to European Economies. URL: <https://www.statista.com/chart/30483/agriculture-sector-as-share-of-gdp-in-european-countries/> (дата звернення 08.01.2026).

9. Food and Agriculture Organization: Statistics. URL: <https://www.fao.org/statistics/en> (дата звернення 08.01.2026).

10. Агробізнес України: Інфографічний довідник 2023/2024. URL: https://agribusinessinukraine.com/get_file/id/the-infographics-report-ukrainian-

agribusiness-2024.pdf (дата звернення 07.01.2026).

11. G7G20 Documents Database: G20 Agriculture Ministers Declaration. URL: <https://g7g20-documents.org/database/document/2024-g20-brazil-sherpa-track-agricultural-ministers-ministers-language-g20-agriculture-ministers-declaration> (дата звернення 09.01.2026).

12. United Nations: THE 17 GOALS. URL: <https://sdgs.un.org/goals> (дата звернення 09.01.2026).

13. Food and Agriculture Organization: Agro-informatics. URL: <https://openknowledge.fao.org/server/api/core/bitstreams/a3b0d0e6-0842-4827-9955-7887888ccb50/content> (дата звернення 11.01.2026).

14. IEG: Agricultural Innovation and Technology in World Bank Projects. URL: https://ieg.worldbankgroup.org/sites/default/files/Data/Evaluation/files/EIN_Agricultural-Innovation_and_Technology.pdf (дата звернення 11.01.2026).

15. European Commission: Common agricultural policy. URL: https://agriculture.ec.europa.eu/common-agricultural-policy/cap-overview_en (дата звернення 11.01.2026).

16. European Commission: EU Digital Strategy. URL: <https://eufordigital.eu/discover-eu/eu-digital-strategy/> (дата звернення 12.01.2026).

17. Про схвалення Стратегії розвитку сільського господарства та сільських територій в Україні на період до 2030 року та затвердження операційного плану заходів з її реалізації у 2025-2027 роках: Розпорядження Кабінету Міністрів України від 15.11.2024 р. № 1163-р. URL: <https://zakon.rada.gov.ua/laws/show/1163-2024-%D1%80#n12> (дата звернення 12.01.2026).

18. UNDP Україна: Стратегія сталого розвитку України до 2030 року. URL: https://www.undp.org/sites/g/files/zskgke326/files/migration/ua/UNDP_Strategy_v06-optimized.pdf (дата звернення 12.01.2026).

19. Про схвалення Стратегії розвитку сфери інноваційної діяльності на період до 2030 року: Розпорядження Кабінету Міністрів України від 10.07.2019 р. № 526-р. URL: <https://zakon.rada.gov.ua/laws/show/526-2019-%D1%80#Text> (дата звернення 13.01.2026).

20. NES: Національна економічна стратегія 2030. URL: <https://nes2030.org.ua/#rec246061582> (дата звернення 14.01.2026).

21. Iqbal N., Shahzad M.U., Sherif E.-S.M., Tariq M.U., Rashid J., Le T.-V., Ghani A. Analysis of Wheat-Yield Prediction Using Machine Learning Models under Climate Change Scenarios. *Sustainability*. 2024. Vol. 16 (16). P. 1–26.

22. Muruganantham P., Wibowo S., Grandhi S., Samrat N.H., Islam N. A Systematic Literature Review on Crop Yield Prediction with Deep Learning and Remote Sensing. *Remote Sensing*. 2022. Vol. 14 (9). P. 1–21.

23. Darra N., Anastasiou E., Kriezi O., Lazarou E., Kalivas D., Fountas S. Can Yield Prediction Be Fully Digitized? A Systematic Review. *Agronomy*. 2023. Vol. 13 (9). P. 1–53.

24. Лактіонов І., Семенов С. Компаративний аналіз програмно-алгоритмічних рішень інтелектуалізованого моніторингу в рослинництві відкритого ґрунту: сучасний стан і перспективи розвитку. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2025. Вип. 82 (2). С. 372–379.

25. Лактіонов І., Семенов С., Дяченко Г., Євстрат'єв М. Результати критичного аналізу та систематизації програмно-алгоритмічних рішень інформаційно-орієнтованого прогнозування врожайності сільськогосподарських культур // «Наукова весна» 2025: матеріали XV Міжн. наук.-техн. конф. аспірантів та молодих вчених, 26–28 березня 2025 р.: Дніпро: НТУ «ДП», 2025. С. 196–197.

26. Food and Agriculture Organization: Review of Crop Yield Forecasting Methods and Early Warning Systems by Basso B., Cammarano D., Carfagna E. URL: https://www.fao.org/fileadmin/templates/ess/documents/meetings_and_workshops/GS_SAC_2013/Improving_methods_for_crops_estimates/Crop_Yield_Forecasting_Methods_and_Early_Warning_Systems_Lit_review.pdf (дата звернення 15.01.2026).

27. Schauburger B., Jägermeyr J., Gornott C. A systematic review of local to regional yield forecasting approaches and frequently used data resources. *European Journal of Agronomy*. 2020. Vol. 120. P. 1–15.

28. Abu Jabed Md., Murad M.A.A. Crop yield prediction in agriculture: A comprehensive review of machine learning and deep learning approaches, with insights

for future research and sustainability. *Heliyon*. 2024. Vol. 10 (24). P. 1–19.

29. Estes L.D., Bradley B.A., Beukes H., Hole D.G., Lau M., Oppenheimer M.G., Schulze R., Tadross M.A., Turner W.R. Crop model intercomparison. *Global Ecology and Biogeography*. 2013. Vol. 22. P. 1007–1018.

30. Hegedus P.B., Maxwell B.D., Mieno T. Assessing performance of empirical models for forecasting crop responses to variable fertilizer rates using on-farm precision experimentation. *Precision Agriculture*. 2023. Vol. 24. P. 677–704.

31. Ahmar A.S., Singh P.K., Ruliana R., Pandey A.K., Gupta S. Comparison of ARIMA, SutteARIMA, and Holt-Winters, and NNAR Models to Predict Food Grain in India. *Forecasting*. 2023. Vol. 5 (1). P. 1–15.

32. Food and Agriculture Organization: Decision-Support System for Agrotechnology Transfer. URL: <https://www.fao.org/land-water/land/land-governance/land-resources-planning-toolbox/category/details/en/c/1027490/> (дата звернення 20.01.2026).

33. APSIM: The Leading Software Framework for Agricultural Systems Modelling and Simulation. URL: <https://www.apsim.info/> (дата звернення 20.01.2026).

34. Food and Agriculture Organization: World Food Studies Simulation Model. URL: <https://www.fao.org/land-water/land/land-governance/land-resources-planning-toolbox/category/details/fr/c/1236431/> (дата звернення 20.01.2026).

35. Belmahi M., Hanchane M., Krakauer N.Y., Kessabi R., Bouayad H., Mahjoub A., Zouhri D. Analysis of Relationship between Grain Yield and NDVI from MODIS in the Fez-Meknes Region, Morocco. *Remote Sensing*. 2023. Vol. 15 (11). P. 1–12.

36. Xu C., Katchova A.L. Predicting Soybean Yield with NDVI Using a Flexible Fourier Transform Model. *Journal of Agricultural and Applied Economics*. 2019. Vol. 51 (3). P. 402–416.

37. Kryvoshein O., Kryvobok O., Kozhushko O. Correlation of satellite-based LAI and actual crop yield. *Ukrainian Journal of Remote Sensing*. 2023. Vol. 10 (3). P. 21–27.

38. Zhuo W., Fang S., Gao X., Wang L., Wu D., Fu S., Wu Q., Huang J. Crop yield prediction using MODIS LAI, TIGGE weather forecasts and WOFOST model: A case study for winter wheat in Hebei, China during 2009–2013. *International Journal of Applied Earth Observation and Geoinformation*. 2022. Vol. 106. P. 1–14.
39. Yield Prophet: Make better decisions with Yield Prophet. URL: <https://www.yieldprophet.com.au/yp/Home.aspx> (дата звернення 24.01.2026).
40. Han, W., Yang Z., Di L., Mueller R. CropScape: A Web service based application for exploring and disseminating US conterminous geospatial cropland data products for decision support. *Computers and Electronics in Agriculture*. 2012. Vol. 84. P. 111–123.
41. Dissanayake D.M.P.W., Rathnayake R.M.K.T., Gihan Chathuranga L.L. Crop Yield Forecasting using Machine Learning Techniques - A Systematic Literature Review. *KDU Journal of Multidisciplinary Studies*. 2023. Vol. 5 (1). P. 54–65.
42. Aslan M.F., Sabanci K., Aslan B. Artificial Intelligence Techniques in Crop Yield Estimation Based on Sentinel-2 Data: A Comprehensive Survey. *Sustainability*. 2024. Vol. 16 (18). P. 1–23.
43. Goel M., Pandey M. Crop Yield Prediction Using AI: A Review // 2024 2nd International Conference on Disruptive Technologies (ICDT), March 15–16: Greater Noida, India, 2024. P. 1547–1553.
44. Diachenko G., Laktionov I., Vinyukov O., Likhushyna H. A Decision Support System for Wheat Powdery Mildew Risk Prediction Using Weather Monitoring, Machine Learning and Explainable Artificial Intelligence. *Computers and Electronics in Agriculture*. 2025. Vol. 230. P. 1–24.
45. Лактіонов І.С., Гнатушенко В.В., Каштан В.Ю., Дяченко Г.Г. Інтелектуальні інформаційні та комп'ютерні технології збору та аналізу даних агромоніторингу: монографія. Міністерство освіти і науки України, НТУ «Дніпровська політехніка». Дніпро: НТУ «ДП», 2025. 235 с.
46. Diachenko G., Laktionov I., Vizniuk A., Gorev V., Kashtan V., Khabarлак K., Shedlovska Y. An Improved Approach to Prediction of Maize Disease Occurrence Based on Weather Monitoring and Machine Learning: Case of the Forest-Steppe and

Northern Steppe of Ukraine. *Baltic J. of Modern Comp.* 2024. Vol. 12 (4). P. 387–414.

47. Sun Q., Zhang Y., Che X., Chen S., Ying Q., Zheng X., Feng A. Coupling Process-Based Crop Model and Extreme Climate Indicators with Machine Learning Can Improve the Predictions and Reduce Uncertainties of Global Soybean Yields. *Agriculture*. 2022. Vol. 12 (11). P. 1–15.

48. Zare H., Weber T.K.D., Ingwersen J., Nowak W., Gayler S., Streck T. Combining Crop Modeling with Remote Sensing Data Using a Particle Filtering Technique to Produce Real-Time Forecasts of Winter Wheat Yields under Uncertain Boundary Conditions. *Remote Sensing*. 2022. Vol. 14 (6). P. 1–26.

49. Dlamini L., Crespo O., van Dam J., Kooistra L. A Global Systematic Review of Improving Crop Model Estimations by Assimilating Remote Sensing Data: Implications for Small-Scale Agricultural Systems. *Remote Sensing*. 2023. Vol. 15 (16). P. 1–25.

50. Hoefsloot P., Ines A., van Dam J., Duveiller G., Kayitakire F., Hansen J. Combining crop models and remote sensing for yield prediction: Concepts, applications and challenges for heterogeneous, smallholder environments: Report of CCFAS-JRC Workshop at JRC. Luxembourg: Publications Office of the EU, 2012. 52 p.

51. Food and Agriculture Organization: AquaCrop. Calculation scheme. URL: <https://www.fao.org/aquacrop/overview/calculationscheme/en/> (дата звернення 03.02.2026).

52. Pandit P., Krishnamurthy K.N., Bakshi B. Chapter 22 – Prediction of crop yield and pest-disease infestation. *Intelligent Data-Centric Systems, AI, Edge and IoT-based Smart Agriculture*. 2022. P. 375–393.

53. Xu R., Dai J., Luo W., Yin X., Li Y., Tai X., Han L., Chen Y., Lin L., Li G., Zou C., Du W., Diao M. A photothermal model of leaf area index for greenhouse crops. *Agricultural and Forest Meteorology*. 2010. Vol. 150 (4). P. 541–552.

54. Bannayan M., Hoogenboom G., Crout N.M.J. Photothermal impact on maize performance: a simulation approach. *Ecological Modelling*. 2004. Vol. 180 (2–3). P. 277–290.

55. USGS: Soil-Water-Balance (SWB): A modified Thornthwaite-Mather model

for estimating groundwater recharge. URL: <https://www.usgs.gov/centers/upper-midwest-water-science-center/science/soil-water-balance-swb-a-modified-thornthwaite> (дата звернення 04.02.2026).

56. Rana G., Katerji N. Measurement and estimation of actual evapotranspiration in the field under Mediterranean climate: a review. *European Journal of Agronomy*. 2000. Vol. 13 (2–3). P. 125–153.

57. Прядкіна Г.О., Стасик О.О., Капітанська О.С., Ярмольська О.Є., Цукренко Н.В. Ефективність використання фотосинтетично активної радіації посівами озимої пшениці. *Вісник Харківського національного аграрного університету. Серія Біологія*. 2019. Вип. 1 (46). С. 23–34.

58. Hatfield J.L. Radiation Use Efficiency: Evaluation of Cropping and Management Systems. *Agronomy Journal*. 2014. Vol. 106. P. 1820–1827.

59. Liang W., Lu H., Wang G., Connor D.J., Rimmington G.M. A simulation model-assisted study on water and nitrogen dynamics and their effects on crop performance in the wheat-maize system I: The model. *Frontiers of Agriculture in China*. 2007. Vol. 1. P. 155–165.

60. Ines A.V.M., Droogers P., Makin I.W., Das Gupta A. Crop growth and soil water balance modeling to explore water management options. *IWMI Working Papers H029557*. 2001. P. 1–26.

61. You Y., Wang Y., Fan X., Dai Q., Yang G., Wang W., Chen D., Hu X., Progress in joint application of crop models and hydrological models. *Agricultural Water Management*. 2024. Vol. 295. P. 1–11.

62. Rokade A., Singh M., Malik P.K., Singh R., Alsuwian T. Intelligent Data Analytics Framework for Precision Farming Using IoT and Regressor Machine Learning Algorithms. *Applied Sciences*. 2022. Vol. 12 (19). P. 1–20.

63. Liu Y., Bao Y. Review on automated condition assessment of pipelines with machine learning. *Advanced Engineering Informatics*. 2022. Vol. 53. P. 1–22.

64. Bryk O., Mudryk I., Holubovskyi M., Stoianov Y. Machine learning models and methods aspects of processing unstructured data // BAIT'2024: The 1st International Workshop on “Bioinformatics and applied information technologies”,

October 02–04: Zboriv, Ukraine, 2024. P. 1–10.

65. Posoldova A. Machine Learning Pipelines: From Research to Production. *IEEE Potentials*. 2020. Vol. 39 (6). P. 38–42.

66. Laktionov I., Diachenko G., Kashtan V., Vizniuk A., Gorev V., Khabarлак K., Shedlovska Y. A Comprehensive Review of Recent Approaches and Hardware-Software Technologies for Digitalisation and Intellectualisation of Open-Field Crop Production: Ukrainian Case Study in the Global Context. *Computers and Electronics in Agriculture*. 2024. Vol. 225. P. 1–31.

67. Pudjihartono N., Fadason T., Kempa-Liehr A.W., O'Sullivan J.M. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. *Frontiers in Bioinformatics*. 2022. Vol. 2. P. 1–17.

68. Barbieri M.C., Grisci B.I., Dorn M. Analysis and comparison of feature selection methods towards performance and stability. *Expert Systems with Applications*. 2024. Vol. 249 (B). P. 1–32.

69. Pandya H., Jaiswal K., Shah M. A Comprehensive Review of Machine Learning Algorithms and Its Application in Groundwater Quality Prediction. *Archives of Computational Methods in Engineering*. 2024. Vol. 31. P. 4633–4654.

70. Woodman R.J., Mangoni A.A. A comprehensive review of machine learning algorithms and their application in geriatric medicine: present and future. *Aging Clinical and Experimental Research*. 2023. Vol. 35. P. 2363–2397.

71. Yang W., Wei Y., Wei H., Chen Y., Huang G., Li X., Li R., Yao N., Wang X., Gu X., Bilal Amin M., Kang B. Survey on Explainable AI: From Approaches, Limitations and Applications Aspects. *Human-Centric Intelligent Systems*. 2023. Vol. 3. P. 161–188.

72. Saeed W., Omlin C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*. 2023. Vol. 263. P. 1–24.

73. Abusitta A., Li M.Q., Fung B.C.M. Survey on Explainable AI: Techniques, challenges and open issues. *Expert Systems with Applications*. 2024. Vol. 255 (C). P. 1–18.

74. Zhai Z., Martínez J.F., Beltran V., Martínez N.L. Decision support systems for agriculture 4.0: Survey and challenges. *Computers and Electronics in Agriculture*. 2020. Vol. 170. P. 1–16.
75. Gutiérrez F., Htun N.N., Schlenz F., Kasimati A., Verbert K. A review of visualisations in agricultural decision support systems: An HCI perspective. *Computers and Electronics in Agriculture*. 2019. Vol. 163. P. 1–28.
76. Vizniuk A., Diachenko G., Laktionov I., Siwocha A., Xiao M., Smola J. A Comprehensive Survey of Retrieval-Augmented Large Language Models for Decision Making in Agriculture: Unsolved Problems and Research Opportunities. *Journal of Artificial Intelligence and Soft Computing Research*. 2025. Vol. 15 (2). P. 115–146.
77. Badshah A., Yousef Alkazemi B., Din F., Zamli K.Z., Haris M. Crop Classification and Yield Prediction Using Robust Machine Learning Models for Agricultural Sustainability. *IEEE Access*. 2024. Vol. 12. P. 162799–162813.
78. Bassine F.Z., Epule Epule T., Kechchour A., Chehbouni A. Recent applications of machine learning, remote sensing, and iot approaches in yield prediction: a critical review. *ArXiv*. 2023. P. 1–35.
79. Piekutowska M., Niedbała G. Review of Methods and Models for Potato Yield Prediction. *Agriculture*. 2025. Vol. 15 (4). P. 1–31.
80. Morales A., Villalobos F.J. Using machine learning for crop yield prediction in the past or the future. *Frontiers in Plant Science*. 2023. Vol. 14. P. 1–13.
81. Karthikeyan J., Murugan D.A. A Study on Enhancing Crop Yield Prediction with Accuracy through Machine and Deep Learning Models. *South Eastern European Journal of Public Health*. 2024. Vol. XXIV (S4). P. 1217–1227.
82. Nikhil U.V., Pandiyan A.M., Raja S.P., Stamenkovic Z. Machine Learning-Based Crop Yield Prediction in South India: Performance Analysis of Various Models. *Computers*. 2024. Vol. 13 (6). P. 1–27.
83. Yuan J., Zhang Y., Zheng Z., Yao W., Wang W., Guo L. Grain Crop Yield Prediction Using Machine Learning Based on UAV Remote Sensing: A Systematic Literature Review. *Drones*. 2024. Vol. 8 (10). P. 1–27.
84. Гнатієнко В.Г., Гнатієнко Г.М., Зозуля О.Л., Снитюк В.Є. Метод

прогнозування врожайності сільськогосподарських культур з використанням мультифакторного аналізу та нейронних мереж. *Науковий вісник Ужгородського університету. Серія «Математика і інформатика»*. 2024. Том 44 (1). С. 93–105.

85. Shawon S.M., Ema F.B., Mahi A.K., Niha F.L., Zubair H.T. Crop yield prediction using machine learning: An extensive and systematic literature review. *Smart Agricultural Technology*. 2025. Vol. 10. P. 1–17.

86. APSIM: Documentation. URL: <https://www.apsim.info/documentation/> (дата звернення 01.02.2026).

87. Food and Agriculture Organization: Land & Water. CropSyst. URL: <https://www.fao.org/land-water/land/land-governance/land-resources-planning-toolbox/category/details/en/c/1236450/> (дата звернення 01.02.2026).

88. Climate Fieldview: Технології Climate Fieldview. URL: <https://www.climatefieldview.com.ua/> (дата звернення 02.02.2026).

89. EOS DATA ANALYTICS: Прогнозування врожайності тютюну та бавовни з WOFOST. URL: <https://eos.com/uk/blog/prohnozuvannia-vrozhainosti-tiutiunu-ta-bavovny-z-wofost/> (дата звернення 03.02.2026).

90. CORTEVA agriscience: Granular Insights. URL: <https://www.corteva.us/products-and-solutions/digital-solutions/granular-insights.html> (дата звернення 03.02.2026).

91. Yotov O., Aleksieva-Petrova A. Data-Driven Prediction Model for Analysis of Sensor Data. *Electronics*. 2024. Vol. 13 (10). P. 1–15.

92. Abioye E.A., Abidin M.S.Z., Mahmud M.S.A., Buyamin S., AbdRahman M.K.I, Otuoze A.O., Ramli M.S.A., Ijike O.D. IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation experiment. *Information Processing in Agriculture*. 2021. Vol. 8 (2). P. 270–283.

93. Al Masarweh M., Alwada'n T., Afandi W. Fog Computing, Cloud Computing and IoT Environment: Advanced Broker Management System. *Journal of Sensor and Actuator Networks*. 2022. Vol. 11 (4). P. 1–17.

94. Pisani F., Martins do Rosario V., Borin E. Fog vs. Cloud Computing: Should I Stay or Should I Go? *Future Internet*. 2019. Vol. 11 (2). P. 1–31.

95. Жураковський Б.Ю., Зенів І.О. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: навчальний посібник для студентів спеціальності 126 Інформаційні системи та технології, спеціалізації «Інформаційне забезпечення робототехнічних систем». Київ: КПІ ім. І. Сікорського, 2021. 271 с.

96. CISCO: The Internet of Things Vertical Solutions. URL: <https://www.cisco.com/web/offer/emear/38586/images/Presentations/P11.pdf> (дата звернення 07.02.2026).

97. Fernando N., Shrestha S., Loke S.W., Lee K. On Edge-Fog-Cloud Collaboration and Reaping Its Benefits: A Heterogeneous Multi-Tier Edge Computing Architecture. *Future Internet*. 2025. Vol. 17 (1). P. 1–23.

98. Cao H., Wachowicz M. An Edge-Fog-Cloud Architecture of Streaming Analytics for Internet of Things Applications. *Sensors*. 2019. Vol. 19 (16). P. 1–32.

99. Заковоротний О.Ю., Орлова Т.О. Порівняльний аналіз хмарних та туманних середовищ Інтернету речей. *Системи управління, навігації та зв'язку*. 2023. № 3. С. 152–154.

100. Пулеко І.В., Єфіменко А.А. Архітектура та технології Інтернету речей: навч. посіб.: електронні дані. Житомир: Державний університет «Житомирська політехніка», 2022. 234 с.

101. Семенов С., Лактіонов І. Структурно-алгоритмічна формалізація програмного забезпечення комплексної трансформації просторово-часових даних агромоніторингу. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2025. Вип. 353 (3.2). С. 469–474.

102. Лактіонов І., Дяченко Г., Семенов С. Компаративний синтез туманної архітектури програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу // Програмне та апаратне забезпечення в інформаційних технологіях: матеріали Міжнар. наук.-практ. конф. молодих вчених та студентів, 6 травня 2025 р.: Луцьк: ЛНТУ, 2025. С. 106–109.

103. Franco da Silva A.C., Hirmer P. Models for Internet of Things Environments – A Survey. *Information*. 2020. Vol. 11 (10). P. 1–19.

104. Костюк Ю.В. Інформаційно-інтелектуальна система оцінки та

прогнозування якості харчової продукції: дис. ... доктор філософії: 122 Комп'ютерні науки. Київ, 2023. 270 с.

105. Muneeb M., Ko K.-M., Park Y.-H. A Fog Computing Architecture with Multi-Layer for Computing-Intensive IoT Applications. *Applied Sciences*. 2021. Vol. 11 (24). P. 1–11.

106. Ungurean I., Gaitan N.C. Software Architecture of a Fog Computing Node for Industrial Internet of Things. *Sensors*. 2021. Vol. 21 (11). P. 1–15.

107. Lucidchart: UML Sequence Diagram Tutorial. URL: <https://www.lucidchart.com/pages/uml-sequence-diagram> (дата звернення 08.02.2026).

108. Lucidchart: What is a Data Flow Diagram. URL: <https://www.lucidchart.com/pages/data-flow-diagram> (дата звернення 08.02.2026).

109. Lucidchart: UML Activity Diagram Tutorial. URL: <https://www.lucidchart.com/pages/uml-activity-diagram> (дата звернення 08.02.2026).

110. Kovtun V., Izonin I., Gregus M. Mathematical models of the information interaction process in 5G-IoT ecosystem: Different functional scenarios. *ICT Express*. 2023. Vol. 9 (2). P. 264–269.

111. Lanotte R., Merro M. A semantic theory of the Internet of Things. *Information and Computation*. 2018. Vol. 259 (1). P. 72–101.

112. Krishnamurthi R., Kumar A., Gopinathan D., Nayyar A., Qureshi B. An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques. *Sensors*. 2020. Vol. 20 (21). P. 1–23.

113. Medium: Data Imputation Techniques: Filling the Gaps in Your Dataset. URL: <https://shamsfiroz.medium.com/data-imputation-techniques-filling-the-gaps-in-your-dataset-125e792fc541> (дата звернення 21.02.2026).

114. Medium: Time series filtering algorithms: a brief overview. URL: <https://medium.com/deelvin-machine-learning/time-series-filtering-algorithms-a-brief-overview-af2d3112cd03> (дата звернення 21.02.2026).

115. Medium: All methods for balancing imbalanced data. URL: <https://medium.com/@dlikhitha99/all-methods-for-balancing-imbalanced-data-decfe>

1f4048d (дата звернення 21.02.2026).

116. Medium: A Simple Guide to Feature Scaling: Making Numbers Easier for Computers to Understand. URL: <https://medium.com/@ashwinkumar577/a-simple-guide-to-feature-scaling-making-numbers-easier-for-computers-to-understand-58ab30f44c35> (дата звернення 21.02.2026).

117. Laktionov I., Semenov S., Diachenko G., Rutkowski L. An end-to-end ML and XAI-based framework for crop yield prediction under agroclimatic variability. *Applied Soft Computing Journal*, 2026, Vol. 200. P. 1–25.

118. Diachenko G., Laktionov I., Moroz D., Derzhevetska M., Semenov S. A computerized method for predicting the risk of powdery mildew in wheat based on software analysis of soil and climatic monitoring data // 1st International Workshop on Advanced Applied Information Technologies (AdvAIT–2024), 05 December 2024: Khmelnytskyi, Ukraine – Zilina, Slovakia: CEUR workshop proceedings, 2025. Vol. 3899. P. 172–185.

119. Python.org: Python 3.12.0. URL: <https://www.python.org/downloads/release/python-3120/> (дата звернення 01.04.2026).

120. NumPy.org: NumPy. URL: <https://numpy.org/> (дата звернення: 02.04.2026).

121. Pandas: pandas documentation. URL: <https://pandas.pydata.org/docs/> (дата звернення: 02.04.2026).

122. Matplotlib documentation: Matplotlib 3.10.6 documentation. Matplotlib – Visualization with Python. URL: <https://matplotlib.org/stable/index.html> (дата звернення: 02.04.2026).

123. Seaborn: Statistical data visualization – seaborn 0.13.2 documentation. URL: <https://seaborn.pydata.org/> (дата звернення: 02.04.2026).

124. Scikit-learn: Machine learning in Python – scikit-learn 1.7.1 documentation. URL: <https://scikit-learn.org/stable/> (дата звернення: 02.04.2026).

125. XGBoost Documentation: Xgboost 3.0.4 documentation. URL: <https://xgboost.readthedocs.io/en/stable/parameter.html> (дата звернення: 02.04.2026).

126. GitHub – thieu1995/mafese: Feature Selection using Metaheuristics Made

Easy: Open Source MAFESE Library in Python. URL: <https://github.com/thieu1995/mafese> (дата звернення: 12.04.2026).

127. MAFESE 1.0.0 documentation: Welcome to MAFESE's documentation. URL: <https://mafese.readthedocs.io/en/latest/pages/support.html> (дата звернення: 12.04.2026).

128. Thieu V., Nguyen N.H., Heidari A.A. Feature selection using metaheuristics made easy: Open source MAFESE library in Python. *Future Generation Computer Systems*. 2024. Vol. 160. P. 340–358.

129. Nguyen V.T., Nguyen N.H., Heidari A.A. Feature Selection using Metaheuristics Made Easy: Open Source MAFESE Library in Python. *Zenodo*. 2023. URL: <https://zenodo.org/records/15586748>

130. Thieu V., Mirjalili S. MEALPY: An open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture*. 2023. Vol. 139. P. 1–27.

131. ELI5 0.16.0 documentation: Welcome to ELI5's documentation. URL: <https://eli5.readthedocs.io/en/latest/overview.html> (дата звернення: 04.05.2026).

132. LIME documentation: Welcome to LIME's documentation. URL: <https://lime.readthedocs.io/en/latest/> (дата звернення: 04.05.2026).

133. Brinch C., Hogerheijde M.R. LIME – a flexible, non-LTE line excitation and radiation transfer method for millimeter and far-infrared wavelengths. *Astromomy & Astrophysics*. 2010. Vol. 523, A25. P. 1–12.

134. Joblib: Running Python functions as pipeline jobs – joblib 1.5.2 documentation. URL: <https://joblib.readthedocs.io/en/stable/> (date of access: 11.05.2026).

135. Python documentation: JSON encoder and decoder. URL: <https://docs.python.org/3/library/json.html> (дата звернення: 14.05.2026).

136. Python documentation: Higher-order functions and operations on callable object. URL: <https://docs.python.org/3/library/functools.html> (дата звернення: 15.05.2026).

137. IPython Documentation: IPython 9.5.0 documentation. URL:

<https://ipython.readthedocs.io/en/9.0.0/api/generated/IPython.display.html> (дата звернення: 15.05.2026).

138. Poetry: Python packaging and dependency management made easy. URL: <https://python-poetry.org/> (дата звернення: 16.05.2026).

139. Kaggle: Crop Yield Prediction Dataset. URL: <https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset> (дата звернення: 15.05.2026).

140. Kaggle: Crop Yield Prediction 99%. URL: <https://www.kaggle.com/code/mahmoudmagdyelnahal/crop-yield-prediction-99> (дата звернення: 15.05.2026).

141. Kaggle: Crop Yield [EDA + Viz]. URL: <https://www.kaggle.com/code/patelris/crop-yield-eda-viz> (дата звернення: 15.05.2026).

142. World Bank Group: Data 360. CCKP ERA5 Dataset. URL: https://data360.worldbank.org/en/dataset/WB_CCKP (дата звернення: 16.05.2026).

143. Laktionov I., Semenov S., Diachenko, G. Dataset for Crop Yield Prediction [Data set]. *Zenodo*. 2026. <https://doi.org/10.5281/zenodo.18761663>.

144. Ying X. An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*. 2019. Vol. 1168. P. 1–6.

145. Scikit-learn: StandardScaler. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> (дата звернення: 18.05.2026).

146. Seger C. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing: Degree project technology. Stockholm: KTH Royal Institute of Technology School of Electrical Engineering and Computer Science, 2018. 34 p. URL: <https://www.diva-portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf>.

147. Kuchibhotla A.K., Brown L.D., Buja A., Cai J. All of Linear Regression. *ArXiv*. 2019. P. 1–39.

148. Cunningham P., Delany S.J. k-Nearest Neighbour Classifiers – A Tutorial.

ACM Computing Surveys. 2021. Vol. 54 (6). P. 1–25.

149. Singh Kushwah J., Kumar A., Patel S., Soni R., Gawande A., Gupta S. Comparative study of regressor and classifier with decision tree using modern tools. *Materials Today: Proceedings*. 2022. Vol. 56. P. 3571–3576.

150. Biau G., Scornet E. A Random Forest Guided Tour. *ArXiv*. 2015. P. 1–42.

151. Mahamat A.A., Boukar M.M., Leklou N., Celino A., Obianyo I.I., Bih N.L., Stanislas T.T., Savastanos H. Decision Tree Regression vs. Gradient Boosting Regressor Models for the Prediction of Hygroscopic Properties of Borassus Fruit Fiber. *Applied Sciences*. 2024. Vol. 14 (17). P. 1–22.

152. Thurnhofer-Hemsi K., López-Rubio E., Molina-Cabello M.A., Najarian K. Radial basis function kernel optimization for Support Vector Machine classifiers. *ArXiv*. 2020. P. 1–9.

153. Qin Y., Li C., Shi X., Wang W. MLP-Based Regression Prediction Model For Compound Bioactivity. *Front Bioeng Biotechnol*. 2022. Vol. 10. P. 1–10.

154. Gong Y., Zhang P. Weather Analysis and Prediction based on Bagging Regression // 2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), October 07–09: Hamburg, Germany, 2022. P. 121–128.

155. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13–17: San Francisco, USA, 2016. P. 785–794.

156. Chicco D., Warrens M.J., Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*. 2021. 7:e623. P. 1–24.

157. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95): Vol. 2, August 20–25: Montreal, Canada, 1995. P. 1137–1143.

158. TensorTonic: Mutual Information. URL: <https://www.tensortonic.com/ml-math/information-theory/mutual-information> (дата звернення: 25.06.2026).

159. Jaberipour M., Khorram E., Karimi B. Particle swarm algorithm for solving

systems of nonlinear equations. *Computers & Mathematics with Applications*. 2011. Vol. 62 (2). P. 566–576.

160. Scikit-learn: Mutual_info_regression. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html (дата звернення: 23.06.2026).

161. Scikit-learn: Random Forest Regressor. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (дата звернення: 23.06.2026).

162. Emergent Mind: Mean Decrease Impurity (MDI) in Tree Models. URL: <https://www.emergentmind.com/topics/mean-decrease-impurity-mdi> (дата звернення: 23.06.2026).

163. ELI5 0.15.0 documentation: Permutation Importance. URL: https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html (дата звернення: 03.07.2026).

164. Mansour, Y., Schain, M. Learning with Maximum-Entropy Distributions. *Machine Learning*. 2001. Vol. 45. P. 123–145.

165. Fisher A., Rudin C., Dominici F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *ArXiv*. 2019. P. 1–81.

166. Štrumbelj E., Kononenko I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*. 2014. Vol. 41. P. 647–665.

167. Kumar S., Kumar M. Developing an XAI-Based Crop Recommendation Framework Using Soil Nutrient Profiles and Historical Crop Yields. *IEEE Transactions on Consumer Electronics*. 2025. Vol. 71 (2). P. 6950–6959.

168. Ribeiro M.T., Singh S., Guestrin C. Why Should I Trust You?: Explaining the Predictions of Any Classifier. *ArXiv*. 2016. P. 1–10.

169. Flask: Pallets Projects. URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення: 07.07.2026).

170. Лактіонов І., Семенов С. Результати досліджень механізмів системної

інтеграції та кросрівневої взаємодії програмно-алгоритмічних компонент прогнозування врожайності агрокультур. *Наукові праці ДонНТУ. Серія: «Інформатика, кібернетика та обчислювальна техніка»*. 2026. № 1 (42)' 2026. С. 82–91.

171. Семенов С.Ю., Лактіонов І.С. Архітектура програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур на основі технології туманних обчислень // Молодь: наука та інновації: матеріали XIII Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених: Том 2, 12–14 листопада 2025 р.: Дніпро: НТУ «Дніпровська політехніка», 2025. С. 124–125.

172. Grzesik P., Mrozek D. Combining Machine Learning and Edge Computing: Opportunities, Challenges, Platforms, Frameworks, and Use Cases. *Electronics*. 2024. Vol. 13 (3). P. 1–26.

173. SQLAlchemy: The Python SQL Toolkit and Object Relational Mapper. URL: <https://www.sqlalchemy.org/> (дата звернення 11.07.2026).

174. Семенов С.Ю., Лактіонов І.С., Дяченко Г.Г. Результати розробки вебзастосунку прогнозування врожайності агрокультур із інтеграцією алгоритмів пояснювального штучного інтелекту // Проблеми використання інформаційних технологій в освіті, науці та промисловості: XX Міжнар. конф., 05 грудня 2025 р.: Дніпро: НТУ «ДП», 2025. №10. С. 56–59.

175. Jinja: Pallets Projects. URL: <https://jinja.palletsprojects.com/en/stable/> (дата звернення: 11.07.2026).

176. Лактіонов І., Семенов С. Вебзастосунок інтелектуалізованого аналізу врожайності сільськогосподарських культур у мінливих агрокліматичних умовах // IX Міжнародна науково-практична конференція «Кліматичні зміни та сільське господарство. Виклики для аграрної науки та освіти», 24 березня 2026 р.: Київ: Н М Ц В Ф П О, 2026. С. 9–11.

177. Codecademy: MVC Architecture Explained: Model, View, Controller. URL: <https://www.codecademy.com/article/mvc-architecture-model-view-controller> (дата звернення: 12.07.2026).

ДОДАТОК А

Перелік публікацій за темою дисертації

Наукові праці, в яких опубліковано основні наукові результати:

1. Семенов С., Лактіонов І. Структурно-алгоритмічна формалізація програмного забезпечення комплексної трансформації просторово-часових даних агромоніторингу. *Вісник Хмельницького національного університету. Серія: Технічні науки*, 2025, Вип. 353 (3.2), С. 469–474. doi.org/10.31891/2307-5732-2025-353-64.

2. Лактіонов І., Семенов С. Компаративний аналіз програмно-алгоритмічних рішень інтелектуалізованого моніторингу в рослинництві відкритого ґрунту: сучасний стан і перспективи розвитку. *Вимірювальна та обчислювальна техніка в технологічних процесах*, 2025, Вип. 82 (2), С. 372–379. doi.org/10.31891/2219-9365-2025-82-54.

3. Лактіонов І., Семенов С. Результати досліджень механізмів системної інтеграції та кросрівневої взаємодії програмно-алгоритмічних компонент прогнозування врожайності агрокультур. *Наукові праці ДонНТУ. Серія: «Інформатика, кібернетика та обчислювальна техніка»*, 2026, № 1 (42) 2026, С. 82–91. doi.org/10.31474/1996-1588-2026-1-42-82-91.

4. Laktionov I., Semenov S., Diachenko G., Rutkowski L. An end-to-end ML and XAI-based framework for crop yield prediction under agroclimatic variability. *Applied Soft Computing Journal*, 2026, Vol. 200. P. 1–25. doi.org/10.1016/j.asoc.2026.115394.

Опубліковані праці апробаційного характеру:

5. Diachenko G., Laktionov I., Moroz D., Derzhevetska M., Semenov S. A computerized method for predicting the risk of powdery mildew in wheat based on software analysis of soil and climatic monitoring data. In: *1st International Workshop on Advanced Applied Information Technologies (AdvAIT–2024)*, Khmelnytskyi, Ukraine – Zilina, Slovakia, December 05, 2024, in CEUR workshop proceedings, 2025, Vol. 3899. P. 172–185. URL: ceur-ws.org/Vol-3899/paper16.pdf.

6. Лактіонов І., Дяченко Г., Семенов С. Компаративний синтез туманної архітектури програмно-алгоритмічного забезпечення інтелектуалізованої трансформації даних агромоніторингу // *Програмне та апаратне забезпечення в*

інформаційних технологіях: матеріали Міжнар. наук.-практ. конф. молодих вчених та студентів (6 травня 2025 р, Луцьк). Луцьк: ЛНТУ, 2025. С. 106–109. URL: https://drive.google.com/drive/folders/1BaojxLgSvEj-o0o_90OCXukJx5NBdzbo.

7. Лактіонов І., Семенов С., Дяченко Г., Євстратьєв М. Результати критичного аналізу та систематизації програмно-алгоритмічних рішень інформаційно-орієнтованого прогнозування врожайності сільськогосподарських культур // *«Наукова весна» 2025: матеріали XV Міжнародної науково-технічної конференції аспірантів та молодих вчених* (26–28 березня 2025 р., Дніпро). Дніпро: НТУ «ДП», 2025. С. 196–197. URL: https://rmv.nmu.org.ua/ua/arkhiv-zbirok-konferentsiy/naukova-vesna-2025/Scientific_Spring_2025.pdf.

8. Семенов С.Ю., Лактіонов І.С. Архітектура програмно-алгоритмічного забезпечення прогнозування врожайності агрокультур на основі технології туманних обчислень // *Молодь: наука та інновації: матеріали XIII Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених: Том 2* (Дніпро, 12–14 листопада 2025 р.). Дніпро: НТУ «Дніпровська політехніка», 2025. С. 124–125. URL: <https://rmv.nmu.org.ua/ua/arkhiv-zbirok-konferentsiy/molod-nauka-ta-innovatsii-2025/molod-2025-vol2.pdf>.

9. Семенов С.Ю., Лактіонов І.С., Дяченко Г.Г. Результати розробки вебзастосунку прогнозування врожайності агрокультур із інтеграцією алгоритмів пояснювального штучного інтелекту // *Проблеми використання інформаційних технологій в освіті, науці та промисловості: XX міжнар. конф.* (5 грудня 2025 р., м. Дніпро): зб. наук. пр. Дніпро: НТУ «ДП», 2025. №10. С. 56–59. URL: https://pzks.nmu.org.ua/ua/science/2025_fin.pdf.

10. Лактіонов І., Семенов С. Вебзастосунок інтелектуалізованого аналізу врожайності сільськогосподарських культур у мінливих агрокліматичних умовах // *IX Міжнародна науково-практична конференція «Кліматичні зміни та сільське господарство. Виклики для аграрної науки та освіти»* (Київ, 24 березня 2026 р.). Київ: НМЦ ВФПО, 2026. С. 9–11. URL: <https://cloud.nmc-vfpo.gov.ua/s/ggaS6Fe72sbs2A9?dir=/&editing=false&openfile=true>.

ДОДАТОК Б

Документи, які підтверджують використання та впровадження одержаних
результатів досліджень

НАЦІОНАЛЬНА АКАДЕМІЯ
АГРАРНИХ НАУК УКРАЇНИ

ДОНЕЦЬКА ДЕРЖАВНА
СІЛЬСЬКОГОСПОДАРСЬКА
ДОСЛІДНА СТАНЦІЯ



NATIONAL ACADEMY
OF AGRARIAN SCIENCES OF UKRAINE

DONETSK STATE AGRICULTURAL
SCIENCE STATION

85307, Донецька обл., м. Покровськ,
вул. Захисників України, 1

Тел.: (050) - 965-91-39

85307, Donetsk region., Pokrovsk,
St. Zakhisnykiv Ukrainy, 1

Phone.: (050) - 965-91-39

E-mail: cnzdiapw@ukr.net

АКТ

використання результатів дисертаційного дослідження на тему «Програмно-алгоритмічне забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури», які одержано **Семеновим Сергієм Юрійовичем**

Цей документ засвідчує результати використання науково-прикладних результатів, здобутих аспірантом **Семеновим С.Ю.** в його дисертаційній роботі на тему «Програмно-алгоритмічне забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури» на здобуття ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення». Результати його досліджень пройшли апробацію й були використані у виробничій і науково-дослідній діяльності Донецької державної сільськогосподарської дослідної станції Національної академії аграрних наук України (ДДСДС НААН). Зокрема, інтелектуалізовані моделі прогнозування врожайності агрокультур на основі алгоритмів машинного навчання та пояснювального штучного інтелекту, а також вебінтерфейс розробленої системної організації комп'ютерної технології прогнозування врожайності агрокультур, було використано як засоби підтримки прийняття рішень під час планування і виконання агропроцедур, пов'язаних із вирощуванням і селекцією сільськогосподарських культур. Зазначені вище результати дисертаційного дослідження аспіранта **Семенова С.Ю.**, які були використані в ДДСДС НААН України, характеризуються практичною цінністю та мають перспективи подальшого масштабування в технологічній і науково-дослідній діяльності підприємств і установ аграрного спрямування, що спеціалізуються на вирощуванні та селекції сільськогосподарських культур.

В. о. директора ДДСДС НААН України,

д. с.-г. н., проф.

20.05.2026



Олександр ВІНЮКОВ

ЗАТВЕРДЖУЮ

В.о. ректора



НТУ «Дніпровська політехніка»

Артем ПАВЛИЧЕНКО

«10» 06 2026 р.

АКТ

впровадження результатів дисертаційної роботи **СЕМЕНОВА Сергія Юрійовича** на тему: **«Програмно-алгоритмічне забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури»**

Цей акт є документальним підтвердженням того, що науково-прикладні результати, отримані здобувачем СЕМЕНОВИМ Сергієм Юрійовичем у дисертаційному дослідженні на тему «Програмно-алгоритмічне забезпечення інтелектуалізованої трансформації вимірювальних даних агромоніторингу на основі туманної архітектури», впроваджено в освітній процес Національного технічного університету «Дніпровська політехніка». Їх використано під час підготовки здобувачів третього (освітньо-наукового) рівня вищої освіти «Доктор філософії» за спеціальностями галузі F Інформаційні технології, а саме під час викладання фахових дисциплін «Моделі та методи створення програмних систем паралельної та розподіленої обробки даних» і «Методи та засоби побудови систем Industry 4.0». У межах цього впровадження використано такі теоретичні та прикладні результати дисертації: структурно-функціональні компоненти децентралізованої архітектури програмно-алгоритмічного забезпечення із застосуванням технології туманних обрахунків; моделі повноциклової інтелектуалізованої трансформації просторово-часових даних агромоніторингу на основі алгоритмів машинного навчання та пояснювального штучного інтелекту; підходи до кросрівневої взаємодії програмно-алгоритмічних компонент прогнозовної аналітики вимірювальних даних.

Завідувач кафедри ПЗКС

Михайло АЛЕКСЕЄВ

Гарант ОНП «Інженерія
програмного забезпечення»

Геннадій ШВАЧИЧ

ДОДАТОК В

Програмне забезпечення прогнозування врожайності сільськогосподарських культур (.ірупb скрипт)

B.1. Import Libraries

```

from functools import reduce
import json
import joblib

import eli5
from eli5.sklearn import PermutationImportance
from eli5 import explain_prediction_sklearn
from eli5 import show_prediction

import lime
import lime.lime_tabular

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.metrics import r2_score, root_mean_squared_error, mean_squared_error,
mean_absolute_error, mean_absolute_percentage_error
from xgboost import XGBRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import BaggingRegressor

from mafese import (
    FilterSelector,
    SequentialSelector,
    RecursiveSelector,
    MhaSelector,
)

from sklearn.feature_selection import mutual_info_regression

import IPython.display as ipd
import IPython.core.display as icd

# Monkey patch
icd.display = ipd.display
icd.HTML = ipd.HTML

```

B.2. Gathering and Cleaning Data

```

# Crops Data: Yield
df_yield = pd.read_csv("data/yield.csv")
df_yield.shape

df_yield.head()

df_yield.tail()

df_yield.drop(columns=[
    "Domain Code",
    "Domain",
    "Area Code",

```

```

        "Element Code",
        "Item Code",
        "Year Code",
        "Element",
        "Unit"
    ],
    inplace=True
)

df_yield.rename(columns={"Value": "value_hg_ha", "Area": "area", "Year": "year",
                        "Item": "item"}, inplace=True)

df_yield.head()

df_yield.describe()

df_yield.info()

# Climate Data: Rainfall
df_rain = pd.read_csv("data/rainfall.csv")
df_rain.shape
df_rain.head()

df_rain.tail()

df_rain.rename(columns={"Area": "area", "Year": "year"}, inplace=True)
df_rain["average_rain_fall_mm_per_year"] =
pd.to_numeric(df_rain["average_rain_fall_mm_per_year"], errors="coerce")

df_rain.dropna(how="any", inplace=True)

df_rain.head()

df_rain.describe()

df_rain.info()

# Crops Data: Pesticides
df_pesticides = pd.read_csv("data/pesticides.csv")
df_pesticides.shape

df_pesticides.head()

df_pesticides.tail()

df_pesticides.rename(index=str, columns={"Value": "pesticides_tonnes", "Area":
"area", "Year": "year"}, inplace=True)
df_pesticides.drop(columns=['Element', 'Domain', 'Unit', 'Item'], inplace=True)

df_pesticides.head()

df_pesticides.describe()

df_pesticides.info()

# Climate Data: Temperature
df_avg_temperature = pd.read_csv("data/temp.csv")
df_avg_temperature.shape

df_avg_temperature.head()

```

```

df_avg_temperature.tail()

df_avg_temperature.dropna(inplace=True)
df_avg_temperature.rename(columns={"country": "area"}, inplace=True)

df_avg_temperature.head()

df_avg_temperature.describe()

df_avg_temperature.info()

# Merged dataset
df_yield_merged = reduce(
    lambda left, right: pd.merge(left, right, on=["year", "area"]),
    [df_rain, df_pesticides, df_avg_temperature, df_yield]
)

df_yield_merged.drop_duplicates(inplace=True)

# Save the DataFrame to a CSV file
# df_yield_merged.to_csv('df_yield_merged.csv', index=False)

df_yield_merged.head()

df_yield_merged.describe()

df_yield_merged.groupby('item').count()

print("Dataframe shape: ", df_yield_merged.shape)

print("\nDataframe columns and types: ")
df_yield_merged.info()

df_yield_merged.describe(include='object')

```

B.3. Data Exploration

```

df_yield_merged.groupby(['area'], sort=True)['value_hg_ha'].sum()

df_yield_merged.groupby(['item', 'area'], sort=True)['value_hg_ha'].sum().nlargest(10)

df_yield_merged[df_yield_merged['area'] == 'Ukraine'].groupby(['item', 'area'],
sort=True)['value_hg_ha'].sum().nlargest(10)

correlation_data=df_yield_merged.select_dtypes(include=[np.number]).corr()

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['font.size'] = 12
plt.rcParams['pdf.fonttype'] = 42 # 3 for Type 3 fonts, 42 for TrueType
plt.rcParams['ps.fonttype'] = 42

mask = np.triu(np.ones_like(correlation_data, dtype=bool))

f, ax = plt.subplots(figsize=(8.27, 6)) # A4 width, 6in height (adjust as needed)
sns.set_style("white")
# Generate a custom diverging colormap
cmap = sns.palette = "mako"

sns.heatmap(correlation_data, cmap = cmap, vmin = -1, vmax = 1, center = 0,
annot=True, fmt='.3f',

```

```

        square=True, linewidths=.5, mask = mask, annot_kws={"size": 12, "family":
"Times New Roman"}, cbar_kws={'shrink': 0.5}, ax=ax)

cbar = ax.collections[0].colorbar
cbar.ax.tick_params(labelsize=12)

# Change the font family of the colorbar label
cbar.ax.yaxis.label.set_fontfamily('Times New Roman') # Or 'monospace', 'sans-
serif', etc.

# Change the font family of the colorbar tick labels
for t in cbar.ax.yaxis.get_ticklabels():
    t.set_fontfamily('Times New Roman')

plt.tight_layout()

# RASTER (if heatmap/complex figure)
# plt.savefig("img/de_heatmap.png", dpi=600, bbox_inches="tight")

plt.show()

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['font.size'] = 12
plt.rcParams['pdf.fonttype'] = 42 # 3 for Type 3 fonts, 42 for TrueType
plt.rcParams['ps.fonttype'] = 42

axes = df_yield_merged.hist(figsize=(8.27, 9))

# Flatten axes array in case it's 2D
for ax in np.array(axes).flatten():
    # Make all spines thinner
    for spine in ax.spines.values():
        spine.set_linewidth(1)

plt.tight_layout()
# VECTOR (preferred for line art)
# plt.savefig("img/de_hists_all.pdf", format="pdf", bbox_inches="tight")
plt.show()

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['font.size'] = 12
plt.rcParams['pdf.fonttype'] = 42 # 3 for Type 3 fonts, 42 for TrueType
plt.rcParams['ps.fonttype'] = 42

axes = df_yield_merged[df_yield_merged['area'] == 'Ukraine'].hist(figsize=(8.27, 9))

# Flatten axes array in case it's 2D
for ax in np.array(axes).flatten():
    # Make all spines thinner
    for spine in ax.spines.values():
        spine.set_linewidth(1)

plt.tight_layout()
# VECTOR (preferred for line art)
# plt.savefig("img/de_hists_Ukraine.pdf", format="pdf", bbox_inches="tight")
plt.show()

sns.pairplot(data=df_yield_merged, hue='item', kind='scatter', height=2, aspect=1)

# RASTER (if heatmap/complex figure)
# plt.savefig("img/de_scatter_all.png", dpi=600, bbox_inches="tight")

```

```

plt.show()

sns.pairplot(data=df_yield_merged[df_yield_merged['area'] ==
'Ukraine'], hue='item', kind='scatter', height=2, aspect=1)

# RASTER (if heatmap/complex figure)
# plt.savefig("img/de_scatter_Ukraine.png", dpi=600, bbox_inches="tight")
plt.show()

# Get unique items
items = df_yield_merged['item'].unique()

plt.figure(figsize=(8.2,6))

for item in items:
    df_item = df_yield_merged[df_yield_merged['item'] == item]
    yearly_mean = df_item.groupby('year')['value_hg_ha'].mean()
    plt.plot(yearly_mean.index, yearly_mean.values, label=item)

plt.xlabel('Year', fontsize=12, family='Times New Roman')
plt.ylabel('Yield (hg/ha)', fontsize=12, family='Times New Roman')
plt.title('Yearly Average Yield per Item per Area', fontsize=12, family='Times New
Roman')
plt.legend(title='Item', fontsize=11, title_fontsize=12, loc='lower center',
bbox_to_anchor=(0.5, -0.35), ncol=5, shadow=False)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/de_yearly_yield_items_all.pdf", format="pdf",
bbox_inches="tight")
plt.show()

# Get unique items
items = df_yield_merged[df_yield_merged['area'] == 'Ukraine']['item'].unique()

plt.figure(figsize=(8.2,6))

for item in items:
    df_item = df_yield_merged[(df_yield_merged['item'] == item) &
(df_yield_merged['area'] == 'Ukraine')]
    yearly_mean = df_item.groupby('year')['value_hg_ha'].mean()
    plt.plot(yearly_mean.index, yearly_mean.values, label=item)

plt.xlabel('Year', fontsize=12, family='Times New Roman')
plt.ylabel('Yield (hg/ha)', fontsize=12, family='Times New Roman')
plt.title('Yearly Yield per Item', fontsize=12, family='Times New Roman')
plt.legend(title='Item', fontsize=11, title_fontsize=12, loc='lower center',
bbox_to_anchor=(0.5, -0.35), ncol=3, shadow=False)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/de_yearly_yield_items_Ukraine.pdf", format="pdf",
bbox_inches="tight")
plt.show()

fig, ax1 = plt.subplots(figsize=(8.27, 3.8))#11.69)) -A4

df2 = df_yield_merged[df_yield_merged['area'] == 'Ukraine']

ax1.set_ylabel('avg_temp')
ax1.set_zorder(4)
ax1.patch.set_visible(False)
ax1.grid(False)

```

```

ax1.set_ylim([5, 12])
ax1.tick_params(
    axis='y',
    which='both',
    left=False,
    length=0)

line1, = ax1.plot(df2['year'], df2['avg_temp'], color='#d500f9', label="avg_temp")

ax2 = ax1.twinx()
ax2.set_zorder(2)
ax2.grid(False)
ax2.tick_params(
    axis='y',
    which='both',
    right=False,
    labelfontfamily='Times New Roman',
    labelsz=12)

ax2.set_ylabel('pesticides_tonnes')

line2, = ax2.plot(df2['year'], df2['pesticides_tonnes'],
    color='#0091ea', label="pesticides_tonnes")

ax3 = ax1.twinx()
ax3.set_zorder(1)
ax3.grid(True, color='#e0e0e0', linewidth=0.5)
ax3.tick_params(
    axis='y',
    which='both',
    right=False,
    labelfontfamily='Times New Roman',
    labelsz=12)

ax3.spines['right'].set_position(('axes', 1.14))
ax3.set_ylabel('average_rain_fall_mm_per_year',)

line3, = ax3.plot(df2['year'], df2['average_rain_fall_mm_per_year'],
    color='#4caf50', label="average_rain_fall_mm_per_year")

lines = [line1, line2, line3]
labels = [line.get_label() for line in lines]

# Create the legend
fig.legend(lines, labels, loc='lower center', bbox_to_anchor=(0.5, -0.05), ncol=5,
    shadow=False, fontsize=9)

# Adjust layout
plt.tight_layout()
# plt.savefig("img/de_yearly_data_Ukraine.pdf", format="pdf", bbox_inches="tight")
plt.show()

# Assume your dataframe is df_yield_merged
areas = df_yield_merged['area'].unique()

# Split areas into chunks of 10
def chunk_list(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

for i, area_chunk in enumerate(chunk_list(areas, 15), start=1):
    fig, ax = plt.subplots(figsize=(8.2, 4))

```

```

# Compute average rainfall for this chunk
avg_rain = df_yield_merged[df_yield_merged['area'].isin(area_chunk)] \
    .groupby('area')['average_rain_fall_mm_per_year'].mean()

# Bar plot
ax.bar(avg_rain.index, avg_rain.values, color='skyblue')

ax.set_title(f'Average Rainfall per Area (Chunk {i})', fontsize=14, family='Times
New Roman')
ax.set_xlabel('Area', fontsize=12, family='Times New Roman')
ax.set_ylabel('Average Rainfall (mm/year)', fontsize=12, family='Times New
Roman')
ax.tick_params(axis='x', rotation=90, labels=10)
ax.tick_params(axis='y', labels=12)
plt.grid(True)
plt.tight_layout()
#plt.savefig(f"img/de_rainfall_per_area_part_{i}.pdf", format="pdf",
bbox_inches="tight")
plt.show()

areas = df_yield_merged['area'].unique()

# Helper to split areas into chunks of 10
def chunk_list(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i+n]

for i, area_chunk in enumerate(chunk_list(areas, 15), start=1):
    fig, ax = plt.subplots(figsize=(8.2, 4))

    # Aggregate by area
    avg_vals = (
        df_yield_merged[df_yield_merged['area'].isin(area_chunk)]
        .groupby("area")[["pesticides_tonnes", "value_hg_ha"]]
        .mean()
    )

    # Set up bar positions
    x = np.arange(len(avg_vals.index))      # positions for areas
    width = 0.35                            # bar width

    # Bars
    ax.bar(x - width/2, avg_vals["pesticides_tonnes"], width,
        label="Pesticides (tonnes)", color="skyblue")
    ax.bar(x + width/2, avg_vals["value_hg_ha"], width,
        label="Yield (hg/ha)", color="sandybrown")

    # Labels and formatting
    ax.set_title(f"Pesticides vs Yield per Area (Chunk {i})",
        fontsize=14, family="Times New Roman")
    ax.set_xlabel("Area", fontsize=12, family="Times New Roman")
    ax.set_ylabel("Value", fontsize=12, family="Times New Roman")
    ax.set_xticks(x)
    ax.set_xticklabels(avg_vals.index, rotation=90, ha="right",
        fontsize=10, family="Times New Roman")
    ax.tick_params(axis="y", labels=12)
    ax.legend(fontsize=10)

```

```

plt.grid(True)
plt.tight_layout()
# plt.savefig(f"img/de_rest_vs_yield_per_area_part_{i}.pdf", format="pdf",
bbox_inches="tight")
plt.show()

sns.barplot(data=df_yield_merged, x = df_yield_merged['item'], y =
df_yield_merged['pesticides_tonnes'])
plt.xticks(rotation=90)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/de_pest_vs_items.pdf", format="pdf", bbox_inches="tight")
plt.show()

f, ax = plt.subplots(figsize=(8.2,4))

sns.scatterplot(x = 'item', y = 'avg_temp', data = df_yield_merged, size=10,
color='g', ax=ax)
plt.xticks(rotation=90)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/de_avg_temp_vs_items.png", dpi=600, bbox_inches="tight")
plt.show()

```

B.4. Data Preprocessing

```

scaler=StandardScaler()

df_train, df_test = train_test_split(
    df_yield_merged,
    test_size=0.1,
    stratify=df_yield_merged["item"]
)

target_column = "value_hg_ha"
numeric_columns = ["year", "average_rain_fall_mm_per_year", "pesticides_tonnes",
"avg_temp"]

df_train[numeric_columns] = scaler.fit_transform(df_train[numeric_columns])
df_train = pd.get_dummies(df_train, columns=["area", "item"], prefix=["country",
"item"])
X_train, y_train = df_train.drop(columns=[target_column]), df_train[target_column]

df_test[numeric_columns] = scaler.transform(df_test[numeric_columns])
df_test = pd.get_dummies(df_test, columns=["area", "item"], prefix=["country",
"item"])
X_test, y_test = df_test.drop(columns=[target_column]), df_test[target_column]

# check if the number of features matches
assert df_train.shape[-1] == df_test.shape[-1]

# Save the DataFrame to a CSV file
# X_train.to_csv('X_train.csv', index=False)

# scaler_stats = {
#     "mean": scaler.mean_.tolist(),
#     "scale": scaler.scale_.tolist(),
#     "var": scaler.var_.tolist(),
#     "n_samples_seen": int(scaler.n_samples_seen_)
# }

```

```
# }

# with open("scaler_stats.json", "w") as f:
#     json.dump(scaler_stats, f, indent=4)
```

```
df_test.head()
```

B.5. Model Building and Evaluation

```
# Linear Regression
linear_regression = LinearRegression()

linear_regression.fit(X_train, y_train)

y_pred = linear_regression.predict(X_test)
print('\nLinear Regression TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
        linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_linear_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# Random forest
random_forest = RandomForestRegressor()

random_forest.fit(X_train, y_train)

y_pred = random_forest.predict(X_test)
print('\nRandom Forest TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
        linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_randforest.png", dpi=600, bbox_inches="tight")
plt.show()

# Random forest: Feature importance
```

```

rf_feature_imp= {'imp':random_forest.feature_importances_, 'names':X_train.columns}

fig, ax = plt.subplots(figsize=(8.27,16.7))
df_imp=pd.DataFrame.from_dict(rf_feature_imp)
df_imp.sort_values(ascending=False,by=["imp"],inplace=True)
df_imp=df_imp.dropna()
sns.barplot(x="imp",y="names",data=df_imp,orient="h",ax=ax)
ax.set_title("Feature importances using MDI")
ax.set_xlabel("Mean decrease in impurity")
plt.tight_layout()
# plt.savefig("img/model_RF_feature_imp_plot.pdf", format="pdf",
bbox_inches="tight")
plt.show()

#7 most important factors that affect crops
fig, ax = plt.subplots(figsize=(8.27,4))
df_imp=pd.DataFrame.from_dict(rf_feature_imp)
df_imp.sort_values(ascending=False,by=["imp"],inplace=True)
df_imp=df_imp.dropna()
df_imp=df_imp.nlargest(7, 'imp')
sns.barplot(x="imp",y="names",data=df_imp,orient="h",ax=ax)
ax.set_title("Feature importances using MDI")
ax.set_xlabel("Mean decrease in impurity")
plt.tight_layout()
# plt.savefig("img/model_RF_feature_imp_plot_t7.pdf", format="pdf",
bbox_inches="tight")
plt.show()

# Multi-layer Perceptron Regressor
mlp = MLPRegressor(
    hidden_layer_sizes=[32, 16],
    activation="relu",
    solver="adam",
    max_iter=1000,
    random_state=42,
    early_stopping=True,
    validation_fraction=0.1,
    verbose=True
)

mlp.fit(X_train, y_train)

y_pred = mlp.predict(X_test)

print('\nMultilayer Perceptron TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred,edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_mlp_reg.png", dpi=600, bbox_inches="tight")

```

```

plt.show()

# Gradient Boosting Regressor
gradient_boosting = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
max_depth=3, random_state=42)

gradient_boosting.fit(X_train, y_train)

y_pred = gradient_boosting.predict(X_test)

print('\nGradient Boosting TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_grad_boost_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# Epsilon-Support Vector Regression
svr = SVR(kernel='rbf', C=0.5, epsilon=0.1)

svr.fit(X_train, y_train)

y_pred = svr.predict(X_test)

print('\nSupport Vector Regressor TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_svr_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# XGBoost
xgboost_reg = XGBRegressor(random_state=42)

xgboost_reg.fit(X_train, y_train)

y_pred = xgboost_reg.predict(X_test)

```

```

print('\nXGBRegressor TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred,edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_xgboost.png", dpi=600, bbox_inches="tight")
plt.show()

# Decision Tree Regressor
dtree_reg = DecisionTreeRegressor(random_state=42)

dtree_reg.fit(X_train, y_train)

y_pred = dtree_reg.predict(X_test)

print('\nDecisionTreeRegressor TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred,edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_decisiontree_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# KNN Regressor
knn_reg = KNeighborsRegressor(n_neighbors=5)

knn_reg.fit(X_train, y_train)

y_pred = knn_reg.predict(X_test)

print('\nKNeighborsRegressor TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred,edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)

```

```

ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_knn_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# Bagging Regressor
bag_reg = BaggingRegressor(n_estimators=150, random_state=42)

bag_reg.fit(X_train, y_train)

y_pred = bag_reg.predict(X_test)

# Define filename
#filename = "bagging_regressor_model.joblib"

# Save the model
#joblib.dump(bag_reg, filename)

print('\nBaggingRegressor TESTING SET RESULTS:')
print('RMSE: {}'.format(root_mean_squared_error(y_test, y_pred)))
print('R2 : {}'.format(r2_score(y_test, y_pred)))

fig, ax = plt.subplots(figsize=(4.2,4))

ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))

ax.set_xlabel('Actual (hg/ha)', fontsize=12)
ax.set_ylabel('Predicted (hg/ha)', fontsize=12)
ax.set_title("Actual vs Predicted Yield (hg/ha)", fontsize=12)
ax.grid(color='grey', linestyle='dashed')
ax.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red',
linewidth = 3)
plt.grid(True)
plt.tight_layout()
# plt.savefig("img/model_bag_reg.png", dpi=600, bbox_inches="tight")
plt.show()

# K-fold Cross Validation for all models
entries = []

models = [
    ('Linear Regression', LinearRegression()),
    ('Random Forest', RandomForestRegressor(random_state=42)),
    ('MLP', MLPRegressor(hidden_layer_sizes=[32, 16],
        activation="relu",
        solver="adam",
        max_iter=1000,
        random_state=42,
        early_stopping=True,
        validation_fraction=0.1,
        verbose=False
    )),
    ('Gradient Boost', GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
max_depth=3, random_state=42)),
    ('SVR', SVR(kernel='rbf', C=0.5, epsilon=0.1)),
    ('XGBoost', XGBRegressor(random_state=42)),

```

```

('Decision Tree', DecisionTreeRegressor(random_state=42)),
('KNN', KNeighborsRegressor(n_neighbors=5)),
('Bagging Regressor', BaggingRegressor(n_estimators=150, random_state=42))
]

for model_name, model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    ACC = model.score(X_test, y_test)
    MAE = mean_absolute_error(y_test, y_pred)
    MAPE = mean_absolute_percentage_error(y_test, y_pred)
    MSE = mean_squared_error(y_test, y_pred)
    RMSE = root_mean_squared_error(y_test, y_pred)
    R2_score = r2_score(y_test, y_pred)
    entries.append((model_name, ACC, MAE, MAPE, MSE, RMSE, R2_score))

    print(f"\n{model_name}")
    num_folds = 5
    kf = KFold(n_splits=num_folds, shuffle=True)
    scores = cross_val_score(model, X_train, y_train, cv=kf)

    for fold, score in enumerate(scores):
        print(f"Fold {fold+1}: {score}")

    mean_score = np.mean(scores)
    print(f"Mean Score: {mean_score}")

df = pd.DataFrame(entries, columns=['Model', 'Accuracy', 'MAE', 'MAPE', 'MSE',
    'RMSE', 'R2_score'])
df_styled_best = df.style.highlight_max(subset=['Accuracy', 'R2_score'],
    color='green').highlight_min(subset=['MSE', 'MAE', 'MAPE', 'RMSE'],
    color='green').highlight_max(subset=['MSE', 'MAE', 'MAPE', 'RMSE'],
    color='red').highlight_min(subset=['Accuracy', 'R2_score'], color='red')

display(df_styled_best)

# Feature Importance
mutual_information_selector = FilterSelector(
    problem="regression", method="MI", n_features=30
)

forward_feature_selector = SequentialSelector(
    problem="regression", n_features=10, estimator="knn"
)

recursive_feature_selector = RecursiveSelector(
    problem="regression", estimator="rf", n_features=10
)

pso_feature_selector = MhaSelector(
    problem="regression",
    estimator="knn",
    optimizer="OriginalPSO",
)

feature_selectors = [
    (mutual_information_selector, "Mutual information"),
    (forward_feature_selector, "Forward"),
    (recursive_feature_selector, "Recursive"),
    #(pso_feature_selector, "Particle swarm optimization")

```

```

]

input_columns = X_train.columns

for feature_selector, feature_selector_name in feature_selectors:
    feature_selector.fit(X_train, y_train)
    print(f"\n===== Feature selection method: {feature_selector_name}
===== \n")
    selected_features =
input_columns[feature_selector.selected_feature_indexes].tolist()
    print(f"Selected features: {selected_features}")
    #X_filtered = feature_selector.transform(X_train)

XX_train = X_train.to_numpy()
yy_train = y_train.to_numpy()

pso_feature_selector.fit(XX_train, yy_train)
print(f"\n===== Feature selection method: Particle swarm optimization
===== \n")
selected_features =
input_columns[pso_feature_selector.selected_feature_indexes].tolist()
print(f"Selected features: {selected_features}")
#X_filtered = feature_selector.transform(X_train)

#Running the mutual information classification and storing the results in a pandas
series
mi=musal_info_regression(X_train, y_train)
series=pd.Series(mi,index=X_train.columns).sort_values(ascending=False)
series_top_s = series.nlargest(n=10)
#Show the top 10 most important features
fig,ax = plt.subplots(figsize=(8.27,5))
plt.style.use('ggplot')

series_top_s.plot(kind='bar')
ax.set_facecolor('white')
ax.grid(False)

ax.tick_params(axis='x', labelsz=11)
ax.tick_params(axis='y', labelsz=11)

for tick in ax.get_xticklabels():
    tick.set_fontname('Times New Roman')
for tick in ax.get_yticklabels():
    tick.set_fontname('Times New Roman')

plt.tight_layout()
# plt.savefig("img/fs_musal_info_regression_plot.pdf", format="pdf",
bbox_inches="tight")
plt.show()

```

B.6. Explainable AI

Eli5

```
fig = plt.figure()
```

```
perm = PermutationImportance(random_forest, random_state=42).fit(X_test, y_test)
```

```
eli5.show_weights(perm, feature_names=X_test.columns.to_numpy())
```

```

eli5.show_prediction(random_forest, X_test[X_test['country_Ukraine'] ==
True].values[1], horizontal_layout=False, show_feature_values=True, feature_names =
X_train.columns.to_numpy())

# LIME
explainer = lime.lime_tabular.LimeTabularExplainer(
    training_data=X_train.values,
    feature_names=list(X_train.columns),
    mode='regression'
)

exp = explainer.explain_instance(
    data_row=X_test[X_test['country_Ukraine'] == True].values[1], # Ukraine, Maize
    predict_fn=random_forest.predict,
    num_features=115,
)

# Display explanation (in notebook)
print(exp.as_list())

exp.show_in_notebook(show_table=True)
#fig = exp.as_pyplot_figure()
#plt.show()

#exp.save_to_file('img/lime_xtest1_uk_maize.html')

explanations = exp.as_list()
filtered = [(feat, val) for feat, val in explanations if not feat.endswith(" 0.00")
            or feat.startswith("country_Ukra") or feat.startswith("item_Maiz")]
print(filtered)

```

B.7. Requirements pyproject.toml

```

[project]
name = "projectssy"
version = "0.1.0"
description = ""
authors = [
    {name = "USERNAME_HERE", email = "USER_EMAIL_HERE"}
]
readme = "README.md"
requires-python = ">=3.12"
dependencies = [
    "numpy (>=2.3.2,<3.0.0)",
    "pandas (>=2.3.2,<3.0.0)",
    "matplotlib (>=3.10.5,<4.0.0)",
    "scikit-learn (>=1.7.1,<2.0.0)",
    "eli5 (>=0.16.0,<0.17.0)",
    "lime (>=0.2.0.1,<0.3.0.0)",
    "nbconvert (>=7.16.6,<8.0.0)",
    "seaborn (>=0.13.2,<0.14.0)",
    "pandoc (>=2.4,<3.0)",
    "xgboost (>=3.0.4,<4.0.0)",
    "mafese (>=1.0.0,<2.0.0)",
    "ipython (>=9.4.0,<10.0.0)"
]

[build-system]
requires = ["poetry-core>=2.0.0,<3.0.0"]
build-backend = "poetry.core.masonry.api"

```

ДОДАТОК Г

Програмне забезпечення серверної та клієнтської частин вебзастосунку
прогнозування врожайності сільськогосподарських культур

Г.1. Серверна частина

Файл forms.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField, IntegerField, FloatField
from wtforms.validators import DataRequired

# WTForm for creating a blog post
class CreateCYPForm(FlaskForm):
    area = StringField("Country", validators=[DataRequired()])
    year = StringField("Year", validators=[DataRequired()])
    average_rain_fall_mm_per_year = IntegerField(
        "Average rain fall mm per year", validators=[DataRequired()]
    )
    pesticides_tonnes = FloatField("Pesticides tonnes", validators=[DataRequired()])
    avg_temp = FloatField("Average temperature", validators=[DataRequired()])
    item = StringField("Crop name", validators=[DataRequired()])
    submit = SubmitField("Email me with prediction")

class CreateTrendQueryForm(FlaskForm):
    area = StringField("Country", validators=[DataRequired()])
    item = StringField("Crop name", validators=[DataRequired()])
    submit = SubmitField("Plot data")
```

Файл main.py

```
from flask import (
    Flask,
    abort,
    render_template,
    request,
    url_for,
    redirect,
    flash,
    send_from_directory,
    jsonify,
    copy_current_request_context,
)
import threading
from werkzeug.security import generate_password_hash, check_password_hash
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column
from sqlalchemy import Integer, String, Float, and_, select
from flask_login import (
    UserMixin,
    login_user,
    LoginManager,
    login_required,
    current_user,
    logout_user,
)
from functools import wraps
from flask_bootstrap import Bootstrap5

from .forms import CreateCYPForm, CreateTrendQueryForm

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import json
import joblib
```

```

import smtplib
import plotly.graph_objs as go
import plotly.offline as pyo
from email.message import EmailMessage
from dotenv import load_dotenv
import os
import eli5
load_dotenv()

SMTP_EMAIL = os.environ["SMTP_EMAIL"]
SMTP_PASSWORD = os.environ["SMTP_PASSWORD"]
SECURE_API_KEY = os.environ["SECURE_API_KEY"]
FILES_FOLDER = os.environ["FILES_FOLDER"]

# INITIALIZATION -----
# Load your base dataframe (one row)
DF_STRUCT = pd.read_csv(f"{FILES_FOLDER}/df_struct.csv")
NUMERIC_COLUMNS = [
    "year",
    "average_rain_fall_mm_per_year",
    "pesticides_tonnes",
    "avg_temp",
]
# Load stats from JSON
with open(f"{FILES_FOLDER}/scaler_stats.json", "r") as f:
    stats = json.load(f)

# Recreate StandardScaler
scaler = StandardScaler()

# Manually set attributes
scaler.mean_ = np.array(stats["mean"])
scaler.scale_ = np.array(stats["scale"])
scaler.var_ = np.array(stats["var"])
scaler.n_samples_seen_ = stats["n_samples_seen"]

# Load ML Model
loaded_model = joblib.load(f"{FILES_FOLDER}/rf_regressor_model.joblib")
# -----

login_manager = LoginManager()

# poetry add $(cat requirements.txt)
# poetry run flask run --debug

app = Flask(__name__)
app.config["SECRET_KEY"] = os.environ["SECRET_KEY"]
Bootstrap5(app)
login_manager.init_app(app)

# CREATE DATABASE
class Base(DeclarativeBase):
    pass

app.config["SQLALCHEMY_DATABASE_URI"] = os.environ["SQLALCHEMY_DATABASE_URI"]
app.config["DOWNLOAD_FOLDER"] = "static"
db = SQLAlchemy(model_class=Base)
db.init_app(app)

# CREATE TABLE IN DB with the UserMixin
class User(UserMixin, db.Model):

```

```

__tablename__ = "users"
id: Mapped[int] = mapped_column(Integer, primary_key=True)
email: Mapped[str] = mapped_column(String(100), unique=True)
password: Mapped[str] = mapped_column(String(100))
name: Mapped[str] = mapped_column(String(1000))

class YieldData(db.Model):
    __tablename__ = "yield_data"
    id: Mapped[int] = mapped_column(Integer, primary_key=True)
    area: Mapped[str] = mapped_column(String(100), nullable=False)
    year: Mapped[int] = mapped_column(Integer, nullable=False)
    average_rain_fall_mm_per_year: Mapped[float] = mapped_column(Float,
nullable=False)
    pesticides_tonnes: Mapped[float] = mapped_column(Float, nullable=False)
    avg_temp: Mapped[float] = mapped_column(Float, nullable=False)
    item: Mapped[str] = mapped_column(String(60), nullable=False)
    value_hg_ha: Mapped[float] = mapped_column(Float, nullable=False)

with app.app_context():
    db.create_all()

def recreate_df(new_data):
    df = DF_STRUCT
    # Replace values in the first row with form data
    for key, value in new_data.items():
        df.loc[0, key] = value
    # df.to_csv(f"{FILES_FOLDER}/recreated_df.csv", index=False)
    return df

def scale_df(df):
    df[NUMERIC_COLUMNS] = scaler.transform(df[NUMERIC_COLUMNS])
    # df.to_csv(f"{FILES_FOLDER}/scaled_df.csv", index=False)
    return df

def notify_via_email(form, pred_res, html_content):
    results_letter = f"""Hello {current_user.name},\n
        Thank you for using our Crop Yield Prediction tool.\n
        Here are the details you submitted:\n
        - Country/Area: {form.area.data} \n
        - Year: {form.year.data}\n
        - Average Rainfall (mm/year): {form.average_rain_fall_mm_per_year.data}\n
        - Pesticides Used (tonnes): {form.pesticides_tonnes.data}\n
        - Average Temperature: {form.avg_temp.data}\n
        - Crop: {form.item.data}\n

        Predicted Crop Yield: {pred_res} hg/ha.\n"""

    # Build email
    msg = EmailMessage()
    msg["From"] = SMTP_EMAIL
    msg["To"] = current_user.email
    msg["Subject"] = "Crop Yield Prediction and Eli5 Explanation Results"

    # Plain-text body
    msg.set_content(results_letter)
    # Add HTML alternative (inline body with ELI5 content)
    msg.add_alternative(
        f"""
<html>

```

```

<body>
    <p>{results_letter.replace("\n", "<br>")}</p>
    <h3>ELI5 Explanation:</h3>
    {html_content}
    <p>If you have any questions or would like further insights, feel free to reply
to this email.</p>
    <p>Best regards,</p>
    <p>Crop Yield Prediction Team</p>
</body>
</html>
"""
    subtype="html",
)

# Send
with smtplib.SMTP("smtp.gmail.com", port=587) as connection:
    connection.starttls()
    connection.login(user=SMTP_EMAIL, password=SMTP_PASSWORD)
    connection.send_message(msg)

def api_key_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        api_key = request.headers.get("X-API-Key")
        if api_key and api_key == SECURE_API_KEY:
            return f(*args, **kwargs)
        return abort(403)

    return decorated_function

@app.route("/")
def home():
    return render_template("index.html", logged_in=current_user.is_authenticated)

@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        email = request.form.get("email")
        result = db.session.execute(db.select(User).where(User.email == email))
        # Note, email in db is unique so will only have one result.
        user = result.scalar()
        if user:
            # User already exists
            flash("You've already signed up with that email, log in instead!")
            return redirect(url_for("login"))

        secured_password = generate_password_hash(
            request.form.get("password"), method="pbkdf2:sha256", salt_length=8
        )
        new_user = User(
            name=request.form.get("name"),
            password=secured_password,
            email=email,
        )

        db.session.add(new_user)
        db.session.commit()
        # Log in and authenticate user after adding details to database.

```

```

        login_user(new_user)

        return redirect(url_for("predictions",
logged_in=current_user.is_authenticated))
        return render_template("register.html", logged_in=current_user.is_authenticated)

@app.route("/login", methods=["GET", "POST"])
def login():
    error = None
    if request.method == "POST":
        email = request.form.get("email")
        password = request.form.get("password")

        # Find user by email entered.
        result = db.session.execute(db.select(User).where(User.email == email))
        user = result.scalar()

        # Email doesn't exist or password incorrect.
        if not user:
            flash("That email does not exist, please try again.")
            return redirect(url_for("login"))
        elif not check_password_hash(user.password, password):
            flash("Password incorrect, please try again.")
            return redirect(url_for("login"))
        else:
            login_user(user)
            return redirect(
                url_for("predictions", logged_in=current_user.is_authenticated)
            )

    return render_template("login.html", logged_in=current_user.is_authenticated)

# Only logged-in users can access the route
@app.route("/secrets")
@login_required
def secrets():
    print(current_user.name)
    # Passing the name from the current_user
    return render_template(
        "secrets.html",
        username=current_user.name,
        logged_in=current_user.is_authenticated,
    )

@app.route("/logout")
@login_required
def logout():
    logout_user()
    return redirect(url_for("home"))

@app.route("/trends", methods=["GET", "POST"])
@login_required
def trends():
    form = CreateTrendQueryForm()
    if form.validate_on_submit:
        result = db.session.execute(
            db.select(YieldData).where(

```

```

        and_(YieldData.area == form.area.data, YieldData.item == form.item.data)
    )
)
data = result.scalars().all()
# print(data)

years = [d.year for d in data]
avg_temp = [d.avg_temp for d in data]
average_rain_fall_mm_per_year = [d.average_rain_fall_mm_per_year for d in data]
pesticides_tonnes = [d.pesticides_tonnes for d in data]
value_hg_ha = [d.value_hg_ha for d in data]

# Build Plotly figure
fig = go.Figure()
fig.add_trace(
    go.Scatter(
        x=years, y=avg_temp, mode="lines+markers", name="Avg Temp, deg C"
    )
)
fig.add_trace(
    go.Scatter(
        x=years,
        y=average_rain_fall_mm_per_year,
        mode="lines+markers",
        name="Rainfall, mm",
    )
)
fig.add_trace(
    go.Scatter(
        x=years,
        y=pesticides_tonnes,
        mode="lines+markers",
        name="Pesticides, tonnes",
    )
)
fig.add_trace(
    go.Scatter(
        x=years, y=value_hg_ha, mode="lines+markers", name="Yield, hg/ha"
    )
)

fig.update_layout(
    title=f"Crop Data for {form.item.data} in {form.area.data}",
    xaxis_title="Year",
    yaxis_title="Values",
    template="plotly_white",
)

# Convert Plotly figure to HTML
plot_div = pyo.plot(fig, output_type="div", include_plotlyjs=False)
return render_template(
    "trends.html",
    logged_in=current_user.is_authenticated,
    form=form,
    plot_div=plot_div,
)

return render_template(
    "trends.html", logged_in=current_user.is_authenticated, form=form

```

```

)

@app.route("/predictions", methods=["GET", "POST"])
@login_required
def predictions():
    form = CreateCYPForm()
    if form.validate_on_submit():

        @copy_current_request_context
        def run_prediction_and_notify():
            new_submission = {
                f"country_{form.area.data}": True,
                "year": form.year.data,
                "average_rain_fall_mm_per_year": form.average_rain_fall_mm_per_year.data,
                "pesticides_tonnes": form.pesticides_tonnes.data,
                "avg_temp": form.avg_temp.data,
                f"item_{form.item.data}": True,
            }

            df = recreate_df(new_submission)
            scaled_df = scale_df(df)
            y_pred = loaded_model.predict(scaled_df)
            exp = eli5.explain_prediction(
                loaded_model,
                scaled_df.values[0],
                feature_names=scaled_df.columns.to_numpy(),
            )
            html = eli5.format_as_html(exp)
            notify_via_email(form, y_pred, html)

        # Run task in background
        thread = threading.Thread(target=run_prediction_and_notify)
        thread.start()
        # Respond to user immediately
        return redirect(url_for("secrets", logged_in=current_user.is_authenticated))

    return render_template(
        "predictions.html", form=form, logged_in=current_user.is_authenticated
    )

@app.route("/add_via_json", methods=["POST"])
@api_key_required
def add_new_data_via_json():
    if request.method == "POST":
        new_entry = request.json["cyp_data"]
        for entry in new_entry:
            new_cyp = YieldData(
                area=entry["area"],
                year=entry["year"],
                average_rain_fall_mm_per_year=entry["average_rain_fall_mm_per_year"],
                pesticides_tonnes=entry["pesticides_tonnes"],
                avg_temp=entry["avg_temp"],
                item=entry["item"],
                value_hg_ha=entry["value_hg_ha"],
            )
            db.session.add(new_cyp)
        db.session.commit()
    return jsonify(response={"success": "Successfully added cyp_data from json."})

```

```
# CREATE A USER LOADER CALLBACK
@login_manager.user_loader
def load_user(user_id):
    return db.get_or_404(User, user_id)
if __name__ == "__main__":
    app.run(debug=True)
```

Г.2. Клієнтська частина

Файл base.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Crop Yield Prediction</title>
    {% block styles %}
    <!-- Load Bootstrap-Flask CSS here -->
    {{ bootstrap.load_css() }}
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
    />
    <link
      rel="stylesheet"
      href="{{ url_for('static', filename='css/styles.css') }}"
    />
    {% endblock %}
  </head>

  <body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="/">Crop Yield Prediction</a>
      <div class="collapse navbar-collapse">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('home') }}">Home</a>
          </li>
          <!-- TODO: Hide the Login/Registration navigation for logged-in users -->
          {% if logged_in: %}
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('trends') }}">Trends</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('predictions') }}">Predictions</a>
          </li>
          {% endif %}
          {% if not logged_in: %}
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('login') }}">Login</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('register') }}">Register</a>
          </li>
          {% endif %}
          {% if logged_in: %}
          <li class="nav-item">
            <a class="nav-link" href="{{ url_for('logout') }}">Log Out</a>
          </li>
          {% endif %}
        </ul>
      </div>
    </nav>
```

```

        </li>
        {% endif %}
    </ul>
</div>
</nav>
{% block content %} {% endblock %}
</body>
</html>

```

Файл index.html

```
{% extends "base.html" %} {% block content %}
```

```
<div class="box">
```

```

    <h1>App Authentication</h1>
    {% if not logged_in: %}
    <!-- TODO: Hide the Login/Registration buttons for logged-in users -->
    <a href="{{ url_for('login') }}" class="btn btn-primary btn-block btn-large"
    >Login</a>
    >
    <a
    href="{{ url_for('register') }}"
    class="btn btn-secondary btn-block btn-large"
    >Register</a>
    >
    {% else %}
    <p> (you are already logged in) </p>
    {% endif %}
</div>

```

```
{% endblock %}
```

Файл login.html

```
{% extends "base.html" %} {% block content %}
```

```
<div class="box">
```

```

    <h1>Login</h1>
    <!-- TODO: Use Message Flashing to display feedback to the user with a <p> -->
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    <ul class="flashes">
    {% for message in messages %}
    <li><p>{{ message }}</p></li>
    {% endfor %}
    </ul>
    {% endif %}
    {% endwith %}

    <form action="{{ url_for('login') }}" method="post">
    <input type="text" name="email" placeholder="Email" required="required" />
    <input
    type="password"
    name="password"
    placeholder="Password"
    required="required"
    />
    <button type="submit" class="btn btn-primary btn-block btn-large">
    Let me in.
    </button>

```

```

    </form>

</div>

{% endblock %}

Файл predictions.html
{% extends "base.html" %}
{% block content %}
{% from "bootstrap5/form.html" import render_form %}

<main class="mb-4">
  <div class="container">
    <div class="row">
      <h1 class="text-center mb-4">Please Fill in the Details Below to Get Your Crop
Yield Prediction</h1>
      <div class="col-lg-8 col-md-10 mx-auto">
        {{ render_form(form, novalidate=True, button_map={"submit": "primary"}) }}
      </div>
    </div>
  </div>
</main>

{% endblock %}

Файл register.html
{% extends "base.html" %}
{% block content %}

<div class="box">
  <h1>Register</h1>
  <form action="{{ url_for('register') }}" method="post">
    <input type="text" name="name" placeholder="Name" required="required" />
    <input type="email" name="email" placeholder="Email" required="required" />
    <input type="password" name="password" placeholder="Password"
required="required" />
    <button type="submit" class="btn btn-primary btn-block btn-large">Sign me
up.</button>
  </form>
</div>

{% endblock %}

Файл secrets.html
{% extends "base.html" %} {% block content %}

<!--TODO: Display the user's name in the h1 title-->
<div class="container">
  <h1 class="title">Dear {{ username }}, we will email you soon with prediction
results!</h1>

</div>
{% endblock %}

Файл trends.html
{% extends "base.html" %}
{% block content %}
{% from "bootstrap5/form.html" import render_form %}
<!-- Load Plotly -->
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<main>

```

```

<div class="container">
  <h1>Crop Historical Data Visualization</h1>
  <div class="col-lg-3 col-md-3 mx-auto">
    {{ render_form(form, novalidate=True, button_map={"submit": "primary"}) }}
  </div>

  {% if plot_div %}
    <h1>Results:</h1>
    {{ plot_div|safe }}
  {% endif %}

</div>
</main>

{% endblock %}

```

Файл styles.css

```

@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px;
margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-align:
center; text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align: middle;
background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff,
#e6e6e6); background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6); background-
image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image:
-o-linear-gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top,
#ffffff, #e6e6e6); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-
color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid
#e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -
webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0,
0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0,
0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0,
0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-
color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-
border-radius: 5px; -moz-border-radius: 5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6;
background-position: 0 -15px; -webkit-transition: background-position 0.1s linear; -
moz-transition: background-position 0.1s linear; -ms-transition: background-position
0.1s linear; -o-transition: background-position 0.1s linear; transition: background-
position 0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color:
#ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-
gradient(top, #6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de,
#4a77d4); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de),
to(#4a77d4)); background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -o-linear-gradient(top, #6eb6de, #4a77d4); background-image:
linear-gradient(top, #6eb6de, #4a77d4); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0); border: 1px solid #3762bc; text-shadow: 1px
1px 1px rgba(0,0,0,0.4); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled,
.btn-primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display: block; margin-bottom: 20px; }

```

```

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-
sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }

html { width: 100%; height:100%; overflow:hidden; }

body {
  width: 100%;
  height:100%;
  font-family: 'Open Sans', sans-serif;
  background: #092756;
  background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%),-moz-linear-gradient(top, rgba(57,173,219,.25) 0%,
rgba(42,60,87,.4) 100%), -moz-linear-gradient(-45deg, #670d10 0%, #092756 100%);
  background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -webkit-linear-gradient(top, rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -webkit-linear-gradient(-45deg, #670d10 0%,#092756
100%);
  background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -o-linear-gradient(top, rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -o-linear-gradient(-45deg, #670d10 0%,#092756 100%);
  background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -ms-linear-gradient(top, rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -ms-linear-gradient(-45deg, #670d10 0%,#092756 100%);
  background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), linear-gradient(to bottom, rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), linear-gradient(135deg, #670d10 0%,#092756 100%);
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D',
endColorstr='#092756',GradientType=1 );
}

a {
  text-decoration: none;
  text-align: center;
}
p {
  color: #ee6f57;
  text-align: center;
}

.box {
  position: absolute;
  top: 50%;
  left: 50%;
  margin: -150px 0 0 -150px;
  width:300px;
  height:300px;
}
.box h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px;
text-align:center; }
.container {
  margin-top: 10rem;
  text-align: center;
}
.container h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-
spacing:1px; text-align:center; }
input {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(0,0,0,0.3);
  border: none;
}

```

```

outline: none;
padding: 10px;
font-size: 13px;
color: #fff;
text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
border: 1px solid rgba(0,0,0,0.3);
border-radius: 4px;
box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
-webkit-transition: box-shadow .5s ease;
-moz-transition: box-shadow .5s ease;
-o-transition: box-shadow .5s ease;
-ms-transition: box-shadow .5s ease;
transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }

.form-label {
color: whitesmoke;
}

```

Γ.3. Requirements pyproject.toml

```

[project]
name = "project-cyp-app"
version = "0.1.0"
description = ""
authors = [
    {name = "USERNAME_HERE", email = "USER_EMAIL_HERE"}
]
readme = "README.md"
requires-python = ">=3.12"
dependencies = [
    "flask-login (==0.6.3)",
    "werkzeug (==3.0.0)",
    "flask (==2.3.2)",
    "flask-sqlalchemy (==3.1.1)",
    "sqlalchemy (==2.0.25)",
    "bootstrap-flask (==2.2.0)",
    "flask-wtf (==1.2.1)",
    "wtforms (==3.0.1)",
    "dotenv (>=0.9.9,<0.10.0)",
    "numpy (>=2.3.2,<3.0.0)",
    "pandas (>=2.3.2,<3.0.0)",
    "scikit-learn (>=1.7.1,<2.0.0)",
    "plotly (>=6.3.0,<7.0.0)",
    "eli5 (==0.15.0)"
]

[build-system]
requires = ["poetry-core>=2.0.0,<3.0.0"]
build-backend = "poetry.core.masonry.api"

```

Г.4. Приклад електронного листа користувачу з результатами прогнозу



Crop Yield Prediction and Eli5 Explanation Results

crophyieldpredictionassistant@gmail.com <crophyieldpredictionassistant@gmail.com>

Sat, May 16, 2026 at 10:29 PM

Hello SemenovSY,

Thank you for using our Crop Yield Prediction tool.

Here are the details you submitted:

- Country/Area: Ukraine
- Year: 2018
- Average Rainfall (mm/year): 560
- Pesticides Used (tonnes): 86781.8
- Average Temperature: 10.5
- Crop: Maize

Predicted Crop Yield: [58320.89] hg/ha.

Eli5 Explanation:

Explained as: decision path

Features with largest coefficients.

Feature weights are calculated by following decision paths in trees of an ensemble (or a single tree for DecisionTreeClassifier). Each node of the tree has an output score, and contribution of a feature on the decision path is how much the score changes from parent to child. Weights of all features sum to the output score or proba of the estimator.

Caveats:

1. Feature weights just show if the feature contributed positively or negatively to the final score, and does not show how increasing or decreasing the feature value will change the prediction.
2. In some cases, feature weight can be close to zero for an important feature. For example, in a single tree that computes XOR function, the feature at the top of the tree will have zero weight because expected scores for both branches are equal, so decision at the top feature does not change the expected score. For an ensemble predicting XOR functions it might not be a problem, but it is not reliable if most trees happen to choose the same feature at the top.

y (score 58320.890) top features

Contribution?	Feature
+77131.239	<BIAS>
+23910.674	avg_temp
+17040.083	pesticides_tonnes
+14762.368	year
+13376.230	item_Maize
+3986.174	item_Soybeans
+432.436	country_Australia
+33.202	country_Belarus
+32.368	country_Romania
-6.073	country_Hungary
-6.765	country_Armenia
-7.494	country_Poland
-11.760	country_Lithuania
-52.917	country_Tajikistan
-439.757	country_Turkey
-462.130	country_Spain
Contribution?	Feature
-2560.232	item_Plantains and others
-2751.138	country_Canada
-3491.095	item_Yams
-8356.912	item_Cassava
-10940.528	item_Sweet potatoes
-13363.633	average_rain_fall_mm_per_year
-22027.716	item_Potatoes
-27905.735	country_Ukraine