Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

магістра
(назва освітньо-кваліфікаційного рівня)

студента	*Візнюка Артема Валентиновича*
	(ПІБ)
академічної групи	*121м-22-3*
	(шифр)
спеціальності	*121 Інженерія програмного забезпечення*
	(код і назва спеціальності)
освітньої програми	*«Інженерія програмного забезпечення»*
	(назва освітньої програми)
на тему:	*Дослідження ефективності програмних реалізацій ML алгоритмів під час прогнозування вірогідності виникнення хвороб сільськогосподарських культур*

_____*А.В. Візнюк*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
розділів кваліфікаційної роботи				
спеціальний	*проф. Лактіонов І.С.*			

Рецензент				

Нормоконтролер	*проф. Лактіонов І.С.*			

Дніпро
2023

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
Завідувач кафедри
Програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 20 23 Року

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

студенту *121м-22-3* *Візнюку Артему Валентиновичу*
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи *Дослідження ефективності програмних реалізацій ML алгоритмів під час прогнозування вірогідності виникнення хвороб сільськогосподарських культур*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 09.10.2023 р. № 1227-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – інфокомунікаційні процеси збору та обробки даних щодо прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Предмет досліджень – алгоритми ML для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Мета НДР – оцінка ефективності ML алгоритмів для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight та визначення найбільш ефективного (згідно метрик RMSE та R^2) методу, який буде інтегровано в програмний застосунок для надання прогнозів щодо ймовірності захворювання спостережуваних культур в режимі реального часу.

Вихідні дані для проведення роботи – дані з метеостанції METOS by Pessl Instruments із використанням IoT платформи FieldClimate.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Новизна запропонованих рішень полягає у встановленні набору вхідних вимірюваних кліматичних даних та оцінці ефективності алгоритмів машинного навчання під час прогнозування вірогідності виникнення хвороби Fusarium Head Blight

кукурудзи, що дає змогу оптимізувати вибір підходу під час проектування конкретних типів систем із обліком метрик R^2 та RMSE.

Практична цінність полягає в розробці програмного застосунку на базі розробленого підходу, який допоможе аграріям прогнозувати виникнення хвороби кукурудзи Fusarium Head Blight, що дозволить своєчасно вжити необхідних заходів для захисту врожаю.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє використання моделі машинного навчання. В результаті роботи повинна бути розроблена програмна реалізація для прогнозування ймовірності виникнення хвороби кукурудзи Fusarium Head Blight.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт (початок –кінець)**
Аналіз теми та постановка задачі	12.09.2023-30.09.2023
Статистичний аналіз та попередня обробка розподілених у часі ґрунтокліматичних даних	01.10.2023-31.10.2023
Написання програмного коду для створення та оцінки методів машинного навчання для прогнозування виникнення захворювання кукурудзи Fusarium Head Blight	01.11.2023-08.12.2023

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки збільшенню врожаю та зменшенню втрат, за рахунок інтеграції ефективного алгоритму машинного навчання для прогнозування вірогідності виникнення хвороби кукурудзи, що дозволить фермерам приймати інформовані рішення щодо застосування заходів захисту та оптимізації агротехнічних процедур.

Соціальний ефект від реалізації результатів роботи очікується позитивним завдяки зменшенню поширення хвороби серед культур, що призведе до зменшення використання хімічних пестицидів та, відповідно, до зменшення впливу на здоров’я населення, яке може бути викликане використанням шкідливих хімікатів.

7 ДОДАТКОВІ ВИМОГИ

Завдання видав _____ *Лактіонов І.С.*
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ *Візнюк А.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *12.09.2023 р.*

Термін подання кваліфікаційної роботи до ЕК *18.12.2023*

РЕФЕРАТ

Пояснювальна записка: 78 стор., 32 рис., 6 таблиць, 1 додаток, 18 джерел.

Об'єкт дослідження: інфокомунікаційні процеси збору та обробки даних щодо прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Предмет дослідження: алгоритми ML для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Мета кваліфікаційної роботи: оцінка ефективності ML алгоритмів для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight та визначення найбільш ефективного (згідно метрик $RMSE$ та R^2) методу, який буде інтегровано в програмний застосунок для надання прогнозів щодо ймовірності захворювання спостережуваних культур в режимі реального часу.

Методи дослідження. Для вирішення поставлених задач використані методи лінійної регресії, нейронної мережі прямого поширення, Random Forest, критичного аналізу і логічного узагальнення відомих результатів наукових досліджень у галузі машинного навчання, статистичного аналізу даних.

Новизна отриманих результатів полягає у встановленні набору вхідних вимірюваних кліматичних даних та оцінці ефективності алгоритмів машинного навчання під час прогнозування вірогідності виникнення хвороби Fusarium Head Blight кукурудзи, що дає змогу оптимізувати вибір підходу під час проектування конкретних типів систем із обліком метрик R^2 та $RMSE$.

Практична цінність полягає в розробці програмного застосунку на базі розробленого підходу, який допоможе аграріям прогнозувати виникнення хвороби кукурудзи Fusarium Head Blight, що дозволить своєчасно вжити необхідних заходів для захисту врожаю.

Область застосування. Розроблене програмне рішення може бути інтегровано в програмний застосунок для надання прогнозів щодо ймовірності захворювання спостережуваних культур в режимі реального часу.

Значення роботи та висновки. Результати проведених досліджень сприятимуть зменшенню втрат врожаю, підвищенню продовольчої безпеки та експортному потенціалу.

Прогнози щодо розвитку досліджень. Оцінка ефективності запропонованого рішення для інших зернових культур та хвороб.

Ключові слова: с/г культури, прогнозування захворювання, машинне навчання, лінійна регресія, нейронна мережа, AdamW, Random Forest.

ABSTRACT

Explanatory note: 78 pages, 32 figures, 6 tables, 1 application, 18 sources.

Object of research: information and communication processes for collecting and processing data to predict the probability of occurrence of Fusarium Head Blight disease in corn.

Subject of research: machine learning algorithms for predicting the probability of occurrence of Fusarium Head Blight disease in corn.

Purpose of Master's thesis: to evaluate the effectiveness of ML algorithms for predicting the probability of occurrence of Fusarium Head Blight disease in corn and to determine the most effective (according to RMSE and R2 metrics) method that will be integrated into a software application to provide real-time predictions of the probability of disease in the observed crops.

Research methods. linear regression, feed-forward neural network, Random Forest, critical analysis and logical generalization of known results of scientific research in the field of machine learning, and statistical data analysis were used to solve the objectives of the work.

Originality of research is in establishing the set of input measured climatic data and evaluation of the effectiveness of machine learning algorithms in predicting the probability of Fusarium Head Blight disease in corn, which makes it possible to optimize the choice of approach when designing specific types of systems, taking into account the R^2 and RMSE metrics.

Practical value of the results consists of the development of a software application based on the developed approach, which will help farmers to predict the occurrence of Fusarium Head Blight disease of corn, which will allow them to take the necessary measures to protect the crop in a timely manner.

Scope of application. The developed software solution can be integrated into a software application to provide real-time forecasts of the probability of disease of the observed crops.

The value of the work and conclusions. The results of the research will help reduce crop losses, increase food security and export potential.

Research forecast and development. To evaluate the effectiveness of the proposed solution for other crops and diseases.

Keywords: crops, disease prediction, machine learning, linear regression, neural network, AdamW, Random Forest.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 9
ВСТУП 10
РОЗДІЛ 1. СУЧАСНИЙ СТАН НАУКОВО-ПРИКЛАДНИХ РОЗРОБОК У ГАЛУЗІ ПРОГНОЗУВАННЯ ВІРОГІДНОСТІ ХВОРОБ С/Г КУЛЬТУР 13
1.1. Аналіз вимог щодо комп'ютеризованого моніторингу ґрунтокліматичних параметрів 13
1.2. Аналіз відомих ML алгоритмів 16
1.3. Аналіз сучасних програмно-апаратних рішень прогнозування вірогідності виникнення хвороб с/г культур 24
1.4. Обґрунтування мети та задач дослідження 29
1.5. Висновки 30
РОЗДІЛ 2. МАТЕРІАЛИ, МЕТОДИ І ПІДХОДИ ДО ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ 32
2.1. Опис обмежень досліджуваної технології 32
2.2. Узагальнена структурно-алгоритмічна організація досліджуваної технології 33
2.3. Методика проведення досліджень 34
2.4. Висновки 48
РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ML АЛГОРИТМІВ ПІД ЧАС ПРОГНОЗУВАННЯ ВІРОГІДНОСТІ ВИНИКНЕННЯ ХВОРОБ С/Г КУЛЬТУР 49
3.1. Регресійні метрики для оцінки якості моделей ML 49
3.2. Результати дослідження лінійної регресії 50
3.3. Результати дослідження для Random Forest 52
3.4. Результати дослідження для нейронної мережі прямого поширення 56
3.5. Порівняльний аналіз отриманих результатів 60
3.6. Перспективи подальшого вдосконалення 63
ВИСНОВКИ 66
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 68
ДОДАТОК А. ПРОГРАМНИЙ КОД 70
ДОДАТОК Б. ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ 78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CNN – convolutional neural network

EMA – exponential moving average

FNN – feedforward neural network

HTTP – hypertext transfer protocol

IoT – internet of things

KNN – k-nearest neighbors

LSTM – long short-term memory

MIoT – massive internet of things

ML – machine learning

RNN – recurrent neural network

SVM – support vector machines

WSN – wireless sensor networks

НМ – нейронна мережа

ШІ – штучний інтелект

ВСТУП

Масивний Інтернет речей (MIoT) – це мережа, що характеризується великою кількістю взаємопов’язаних пристроїв. Як правило, ці пристрої, часто розташовані у віддалених місцях, збирають дані і передають інформацію на центральний сервер або хмару. Ці «речі», як правило, недорогі, малопотужні і мають обмежені обчислювальні можливості. Наочним прикладом є сільськогосподарські датчики, розміщені на полях, річках і в лісах.

Стрімке зростання MIoT значною мірою пов’язане з розвитком стільникового зв’язку. За прогнозами Juniper, кількість підключень до IoT зросте до 83 мільярдів до 2024 року з 35 мільярдів у 2020 році, причому приблизно половина з них припадатиме на категорію MIoT. Водночас IndustryARC прогнозує середньорічний темп зростання (CAGR) ринку масового IoT на рівні 7,1%, який досягне 121,4 мільярда доларів до 2026 року.

Зростання світового попиту на продовольство, який, за прогнозами, подвоїться до 2050 року, ще більше загострює проблему підвищення продуктивності сільського господарства. Незважаючи на такі виклики, як зниження рівня води, зміна клімату та зменшення площ орних земель, технології, що базуються на даних, є перспективним шляхом підвищення продуктивності фермерських господарств на 67% до 2050 року, згідно з даними Міжнародного науково-дослідного інституту продовольчої політики (International Food Policy Research Institute).

Польові випробування продемонстрували ефективність сенсорних технологій, таких як точне зрошення, яке може підвищити продуктивність фермерських господарств на 45%, зменшивши при цьому споживання води на 35%. Аналогічно, використання сенсорних даних з систем аерофотозйомки, таких як дрони, дозволяє фермерам картографувати поля, стежити за станом посівів і виявляти аномалії. Таким чином, потенціал MIoT знаходить гідне застосування в аграрному секторі, де потреба в освоєнні великих і віддалених територій є нагальною.

Однак, для того щоб з часом накопичені дані полегшили визначення оптимальних методів ведення сільського господарства, що призводить до підвищення врожайності, зменшення витрат ресурсів та мінімізації впливу на навколишнє середовище, потрібно їх певним чином обробляти.

Всупереч традиційним уявленням, ШІ відіграє ключову роль у сучасному сільському господарстві. Тонкощі фермерства, з його переплетеною павутиною передсезонних і сезонних рішень, виграють від здатності ШІ обробляти великі обсяги даних від сільськогосподарського обладнання, датчиків навколишнього середовища і віддалених джерел. Така зміна парадигми підвищує ефективність і стійкість у всьому ланцюжку виробництва продуктів харчування.

Підвищення врожайності та зменшення витрат ресурсів набуває особливої нагальності в контексті зменшення площ вирощування сільськогосподарських культур в України. Тому значно актуалізується проблема розробки та впровадження стратегій менеджменту сільськогосподарських підприємств на основі логіки «більші обсяги виробництва на менших площах». Такий результат може бути досягнутий шляхом збільшення показників збереження сільськогосподарських культур протягом повного циклу вирощування, шляхом прогнозування появи хвороб сільськогосподарських культур.

Таким чином, тема даної кваліфікаційної роботи є актуальною, оскільки в рамках дослідження визначається найбільш ефективний ML алгоритм для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Об’єкт дослідження – інфокомунікаційні процеси збору та обробки даних щодо прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Предмет дослідження – алгоритми ML для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Мета НДР – оцінка ефективності ML алгоритмів для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight та визначення

найбільш ефективного (згідно метрик $RMSE$ та R^2) методу, який буде інтегровано в програмний застосунок для надання прогнозів щодо ймовірності захворювання спостережуваних культур в режимі реального часу.

Наукова новизна запропонованих рішень полягає у встановленні набору вхідних вимірюваних кліматичних даних та оцінці ефективності алгоритмів машинного навчання під час прогнозування вірогідності виникнення хвороби Fusarium Head Blight кукурудзи, що дає змогу оптимізувати вибір підходу під час проектування конкретних типів систем із обліком метрик R^2 та $RMSE$.

Практична цінність полягає в розробці програмного застосунку на базі розробленого підходу, який допоможе аграріям прогнозувати виникнення хвороби кукурудзи Fusarium Head Blight, що дозволить своєчасно вжити необхідних заходів для захисту врожаю.

Структура роботи: пояснювальна записка: 78 стор., 32 рис., 6 таблиць, 1 додаток, 18 джерел.

РОЗДІЛ 1. СУЧАСНИЙ СТАН НАУКОВО-ПРИКЛАДНИХ РОЗРОБОК У ГАЛУЗІ ПРОГНОЗУВАННЯ ВІРОГІДНОСТІ ХВОРОБ С/Г КУЛЬТУР

1.1. Аналіз вимог щодо комп'ютеризованого моніторингу ґрунтокліматичних параметрів

За статистикою сільськогосподарського виробництва, проведеною всесвітньо визнаною організацією FAO [1], зернові культури, виробництво яких зросло втричі за останні 20 років, є найбільш популярними сільськогосподарськими культурами вирощуваними у світі в умовах відкритого ґрунту. На підставі статистичного аналізу даних, агрегованих FAO [2] було встановлено, що найбільш вирощуваними (за засіяними площами) зерновими культурами в регіонах Східної та Південної Європи, є пшениця, кукурудза та ячмінь. Тренд динаміки врожайності (т/га) та задіяних для вирощування площ (млн. га) за період з 2012 по 2021 р. для Південної та Східної Європи наведено на рис. 1.1 та 1.2.

Відповідно до проведеного статистичного аналізу задіяних на вирощування площ в Україні серед зернових культур перше місце посідає пшениця, друге кукурудза, а третє – ячмінь (див. рис. 1.3.). Серед цих зернових культур кукурудза дає найбільші обсяги врожайності, виражені в т/га згідно статистики, наведеній на рис. 1.4.

Наведені на рис. 1.1., 1.2 та 1.3. статистичні дані з FAO підтверджують тренд динаміки популярності вирощування зернових культур в регіонах Південної та Східної Європи, а також України, що доводить необхідність застосування науково-прикладних підходів для збереження врожаю та раціонального використання ресурсів і добрив під час повного циклу вирощування.

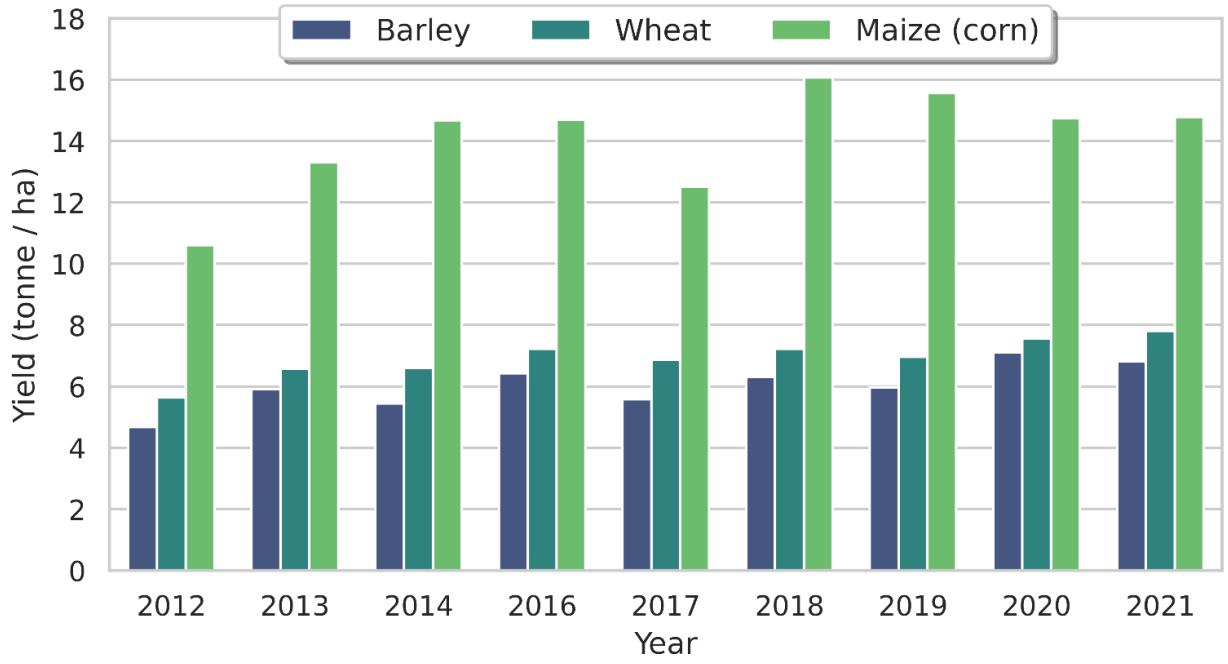

Рис. 1.1. Тренд динаміки врожайності ячменю, пшениці та кукурудзи в Східній та Південній Європі

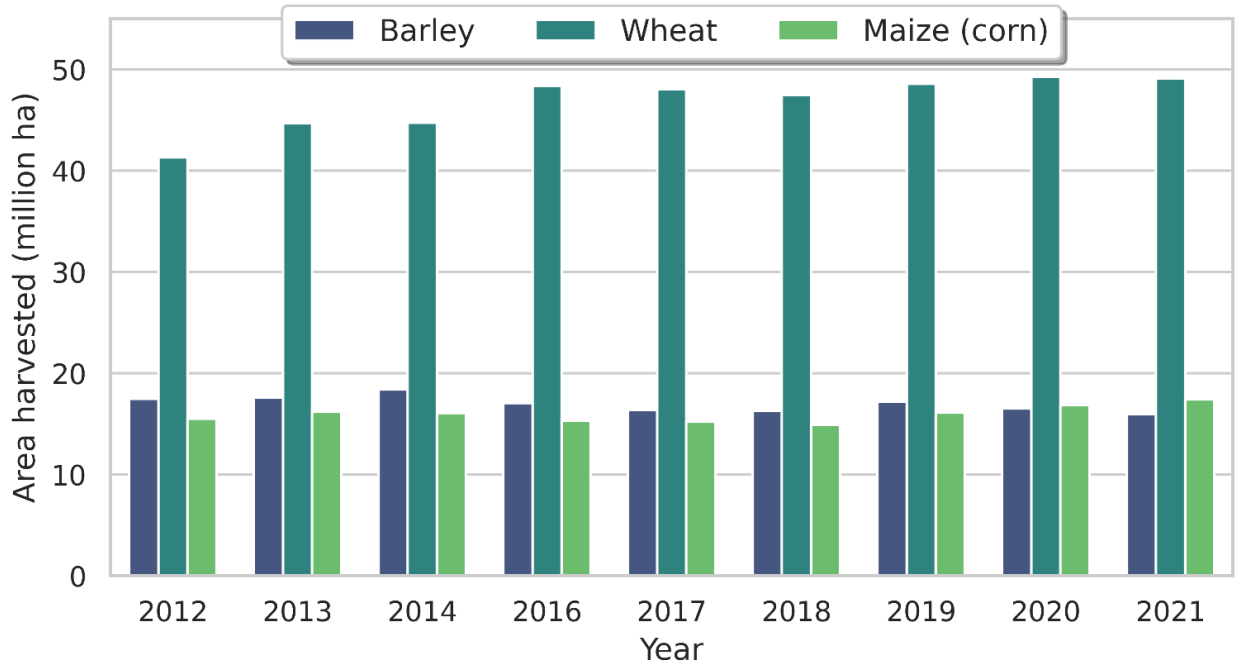

Рис. 1.2. Задіяні сільськогосподарські площі (млн. га) з 2012 по 2021 р. для ячменю, пшениці та кукурудзи в Східній та Південній Європі

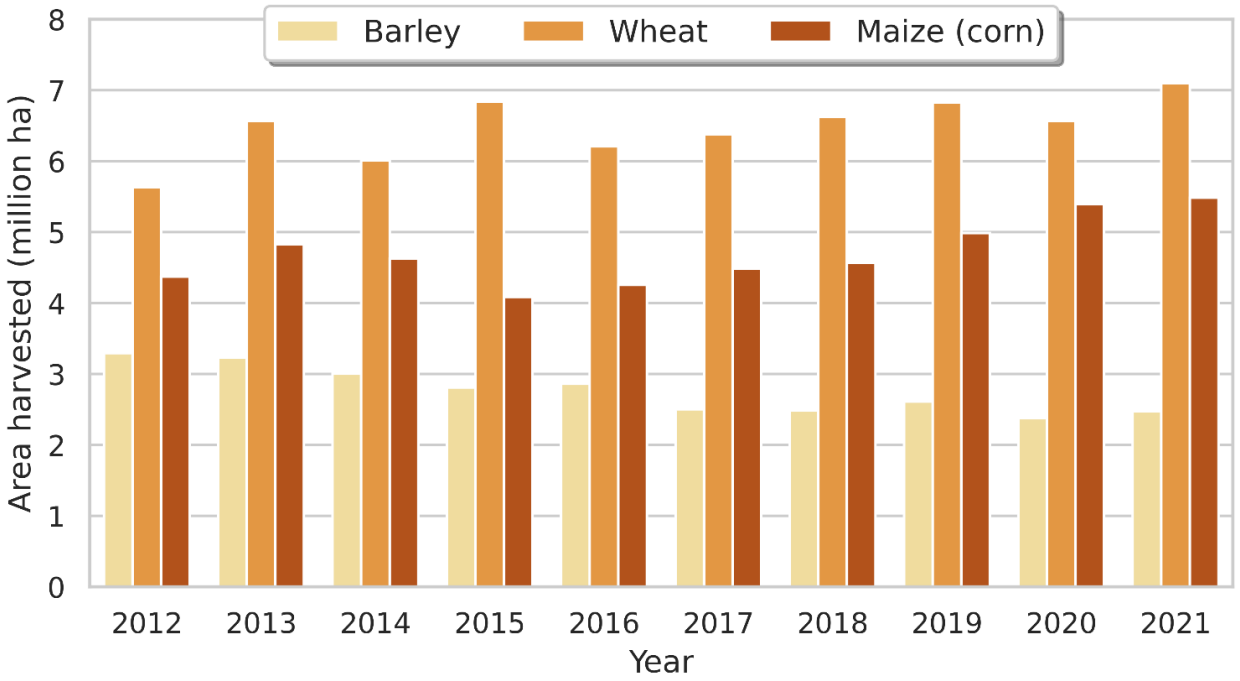

Рис. 1.3. Графік задіяних площ на ячмінь, пшеницю та кукурудзу для України з 2012 по 2021 рік.

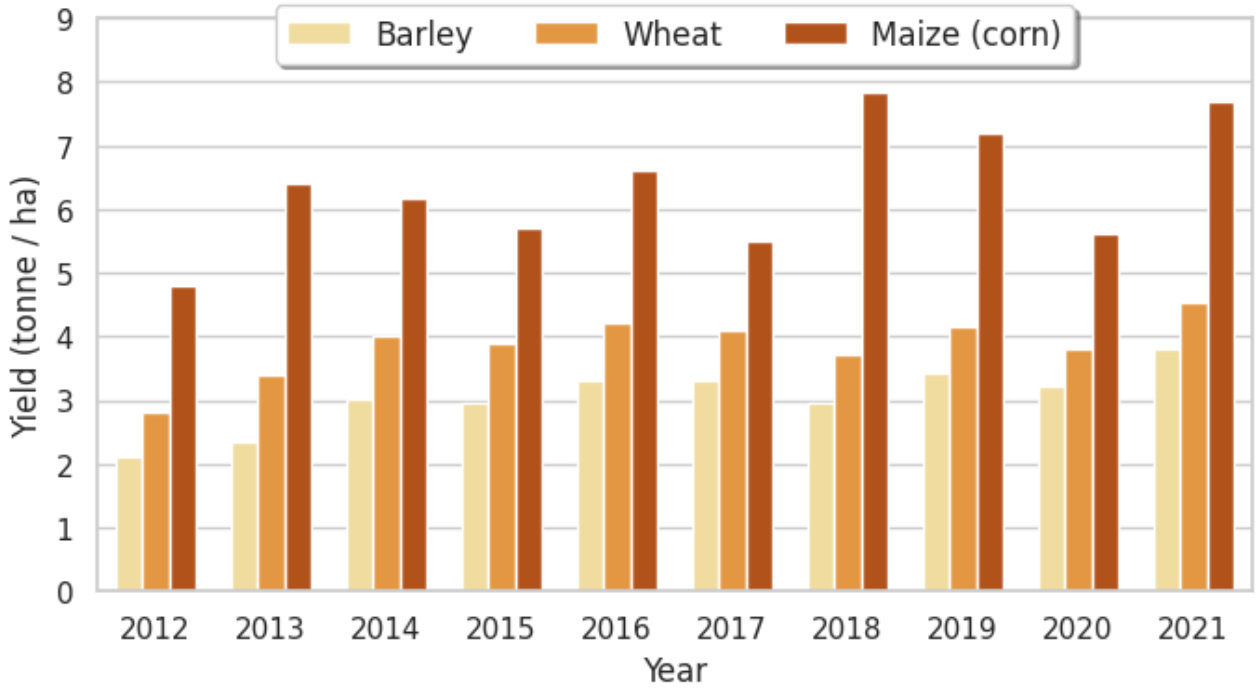

Рис. 1.4. Врожайність ячменю, пшениці та кукурудзи в Україні з 2012 по 2021 рік

В ході аналізу останніх досліджень та публікацій [3-7] на тему визначення ймовірності захворювання с/г культур встановлено, що найбільше на виникнення захворювання впливають зміни кліматичних показників з часом, а саме: температури повітря, відносної вологості повітря, вологості ґрунту, кількості опадів, швидкості вітру, вологості листків, часу вологості листків, часу сонячного сяйва тощо. На основі встановленої сукупності вимірюваних параметрів, а також сучасних вимог до агротехнічних процедур протягом повного циклу вирощування зернових культур можна зробити висновок про необхідність реалізації задачі агрегування вхідних даних методами комп'ютеризованого моніторингу в режимі онлайн.

1.2. Аналіз відомих ML алгоритмів

Задача прогнозування ймовірності виникнення захворювання с/г культур є задачею прогнозування часових рядів, де на вихідне значення цільової змінної y_t (ймовірності захворювання) в момент часу t впливають поточні та попередні значення кліматичних показників, а також попередні значення y_k, де $k < t$.

Регресійні моделі можуть невдало спрацювати на задачах прогнозування часових рядів, якщо враховувати лише значення кліматичних показників в поточний момент часу t. Тому вхідні дані необхідно обробити таким чином, щоб в кожному рядку даних були не лише значення поточних кліматичних показників, а й значення в k минулих моментах часу, а також значення цільової змінної (ймовірності захворювання) в k минулих моментах часу. В такому разі дану проблему можна переформулювати в задачу машинного навчання з вчителем, де кожний рядок даних буде незалежним від інших і може бути використаний окремо для прогнозу. Недолік такого підходу – потрібно визначати значення гіперпараметра k.

Розглянемо спочатку такі регресійні моделі, як: лінійна регресія, нейронна мережа прямого поширення та випадковий ліс:

1. Лінійна регресія [8]

Під час використання лінійної регресії припускається лінійна взаємозалежність між вхідними значеннями та вихідним. Лінійна регресія визначається за формулою 1.1:

$$\hat{y} = \sum_{i=1}^{n} w_i x_i + b, \tag{1.1}$$

де $\hat{y}$ – прогнозоване моделлю значення;

x_i – вхідна характеристика;

w_i – ваговий коефіцієнт при вхідній характеристиці x_i;

b – вільний коефіцієнт.

На рис. 1.5 наведено приклад лінії апроксимації моделі лінійної регресії:

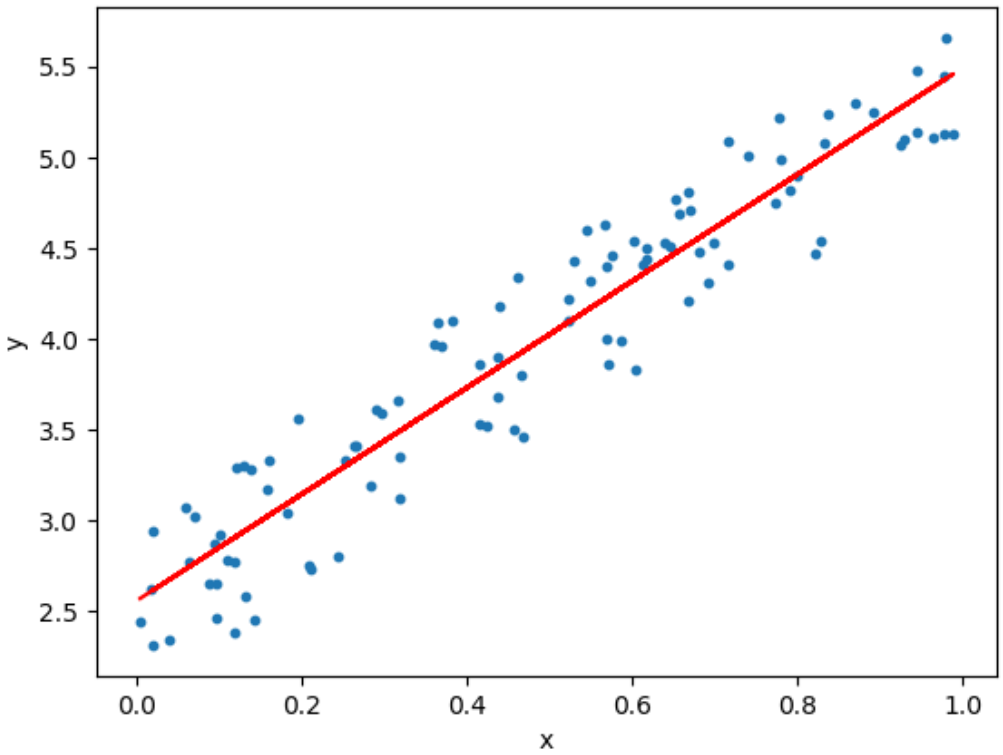

Рис. 1.5. Апроксимація даних лінійною регресією

Функція втрат J, яку потрібно мінімізувати для знаходження лінії апроксимації – квадратична похибка (див. формула 1.2.):

$$J = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2\,, \tag{1.2}$$

де m – кількість екземплярів в навчальних даних;

y_i – реальне значення цільової змінної для i-того екземпляру;

$\hat{y}_i$ – встановлене моделлю значення.

Знайти оптимальні коефіцієнти, що мінімізують функцію втрат J можна двома шляхами: аналітичним та чисельним, який передбачає використання методу градієнтного спуску.

Нижче наведено формулу 1.3 аналітичного рішення для знаходження локального мінімуму функції втрат J:

$$\vec{w} = (X^T X)^{-1} X^T Y, \tag{1.3}$$

де $\vec{w}$ – вектор коефіцієнтів лінійної регресії довжини k+1 або координати точки мінімуму функції J (число k – кількість вхідних значень, а перше значення вектору $\vec{w}$ – значення вільного коефіцієнта регресії);

Y – вектор-стовпчик довжини m, що є розміром навчальної вибірки;

X – матриця розмірністю m на k+1 (перший стовпчик цієї матриці – одиниці, інші елементи рядків – вхідні значення в навчальній вибірці).

Аналітичне рішення дозволяє точно порахувати координати точки мінімуму, однак, воно містить такі операції, як матричне множення та знаходження оберненої матриці. Складність обчислень, описаних у формулі 1.3 – $O(k^2 m + k^3)$, де k – кількість вхідних характеристик, а m – розмір навчальної вибірки. Для великої кількості вхідних значень та навчальних даних таке рішення може вимагати занадто багато обчислень та часу, тому в якості альтернативи може бути використане рішення через градієнтний спуск.

За алгоритмом навчання градієнтного спуску коефіцієнти лінійної регресії змінюються на кожній ітерації наступним чином (див. формула 1.4):

$$b \leftarrow b - \eta \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i),$$
$$w_k \leftarrow w_k - \eta \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_i^k, \tag{1.4}$$

де η – коефіцієнт навчання;

w_k – значення вагового коефіцієнта для k-того вхідного значення;

x_i^k – значення k-того вхідного значення для i-того екземпляра.

2. Нейронна мережа прямого поширення (FNN) [8]

На відміну від лінійної регресії нейронна мережа прямого поширення здатна знаходити нелінійні залежності між вхідними значеннями та вихідним. На рис. 1.6. представлена типова структура нейронної мережі для регресійних задач:

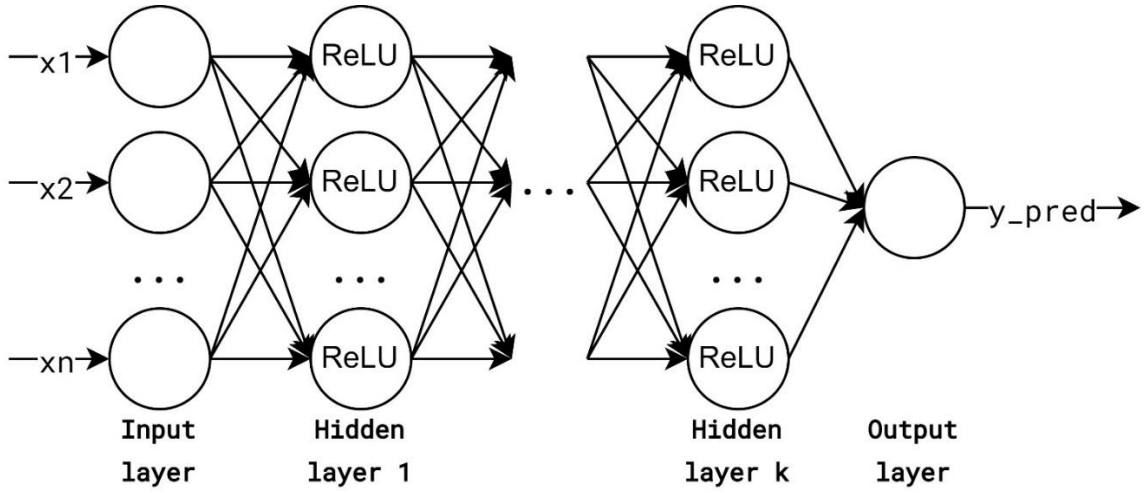

Рис. 1.6. Структура нейронної мережі

Вхідні значення для j-того нейрону в l-тому прихованому шару нейронної мережі обчислюються за наступною формулою (див. формула 1.5):

$$z_l^j = w_l^j \cdot a_{l-1} + b_l^j, \tag{1.5}$$

де z_l^j – вхідне значення до j-того нейрону l-того шару;

w_l^j – вектор відповідних вагових коефіцієнтів для j-того нейрону l-того шару;

a_{l-1} – вектор вихідних значень з попереднього l-1 шару (якщо l=1, то a_{l-1} – вектор вхідних значень до нейронної мережі);

b_l^j – вільний коефіцієнт j-того нейрона.

Для того, щоб нейрона мережа могла знаходити нелінійні залежності до вхідних значень в нейронах у внутрішніх шарах застосовують нелінійні активаційні функції. Для регресійних задач найчастіше використовують функцію ReLU, або $max(x, 0)$, де x – аргумент функції (див. рис. 1.7.):

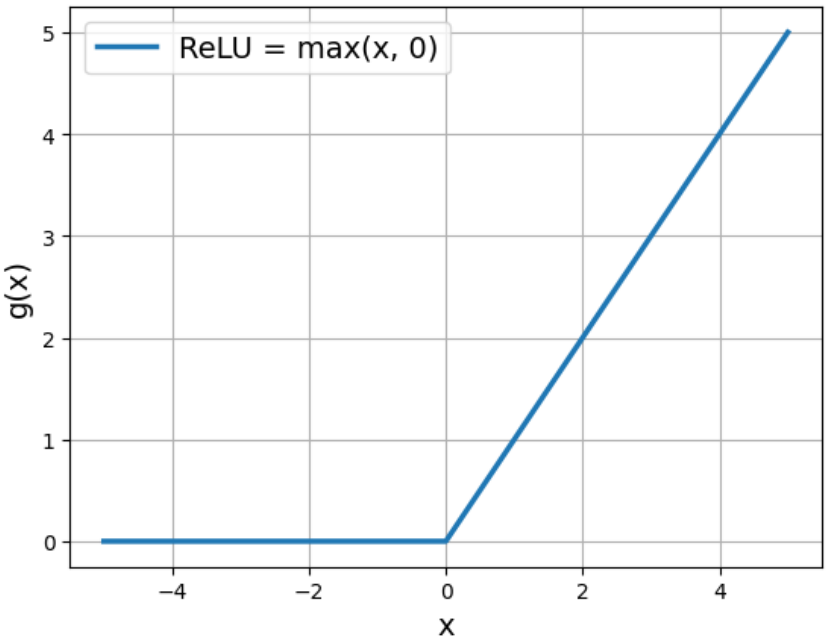

Рис. 1.7. Графік активаційної функції ReLU

Функція втрат, яку потрібні мінімізувати під час навчання аналогічна тій, що наведена у формулі 1.2 (середня квадратична похибка). Коефіцієнти нейронної мережі оновлюються за алгоритмом градієнтного спуску за формулою 1.6:

$$b_j^l \leftarrow b_j^l - \eta \frac{dJ}{db_j^l},$$
$$w_{ij}^l \leftarrow w_{ij}^l - \eta \frac{dJ}{dw_{ij}^l}, \tag{1.6}$$

де η – коефіцієнт навчання;

l – індекс шару;

i – індекс нейрону в шарі l;

j – індекс нейрона в попередньому l-1 шарі;

$\frac{dJ}{db_j^l}$ та $\frac{dJ}{dw_{ij}^l}$ – частинні похідні функції втрат J відносно коефіцієнтів нейронної мережі, обчислені за методом зворотного розповсюдження помилки (Backpropagation).

Нижче наведено формулу 1.7 для обчислення частинних похідних $\frac{dJ}{db_j^l}$ та $\frac{dJ}{dw_{ij}^l}$ відносно коефіцієнтів вихідного шару L нейронної мережі (символ «$\circ$» позначає покомпонентний добуток матриць):

$$\delta^L = \frac{dJ}{da^L} \circ \frac{da^L}{dz^L},$$
$$\frac{dJ}{db_j^L} = \delta_j^L, \tag{1.7}$$
$$\frac{dJ}{dw_{ij}^L} = \delta_j^L a_i^{L-1},$$

де $\frac{da^L}{dz^L}$ – вектор частинних похідних активаційних функцій відносно вхідних значень z^L до останнього шару L;

$\frac{dJ}{da^L}$ – вектор частинних похідних функції втрат J відносно активацій a^L;

δ^L – вектор сигналів похибок вихідного шару;

δ_j^L – сигнал похибки для j-того нейрону шару L;

a_i^{L-1} – вихід з i-того нейрону шару L-1.

Обчислення частинних похідних функції втрат J відносно вагових коефіцієнтів не вихідного шару (індекси шарів $l < L$) за алгоритмом Backpropagation наведено у формулі 1.8:

$$\delta^l = (\delta^{l+1} W^l) \circ \frac{da^l}{dz^l},$$
$$\frac{dJ}{db_j^l} = \delta_j^l, \tag{1.8}$$
$$\frac{dJ}{dw_{ij}^l} = \delta_j^l a_i^{l-1},$$

де W^l – матриця вагових коефіцієнтів (кількість рядків – кількості нейронів в шарі l, кількість стовпців – кількості нейронів в попередньому шарі l-1);

$\frac{da^l}{dz^l}$ – вектор похідних активаційних функцій шару l;

δ^l та δ^{l+1} – вектори сигналів похибок для поточного та наступного шару нейронної мережі відповідно.

3. Випадковий ліс (Random Forest, RF) [8, 9]

Ансамблевий метод машинного навчання, який удосконалює модель «Дерево прийняття рішень». Один із головних недоліків глибоких дерев прийняття рішень – перенавчання. Метод «Випадковий ліс» вирішує цю проблему за рахунок використання багатьох дерев прийняття рішень, кожне з яких створене з окремої частини навчальної вибірки, для отримання усередненого результату. Для регресійної задачі вихідне значення з алгоритму – середнє значення результатів, отриманих з n дерев прийняття рішень. На рис. 1.8. наведено структуру моделі «Випадковий ліс». Одними з найбільш важливих гіперпараметрів для налаштування даної моделі є кількість дерев прийняття рішень та максимальна глибина кожного дерева. Збільшення кількості дерев та

зменшення їх максимальної глибини допомагає вирішити проблему перенавчання.

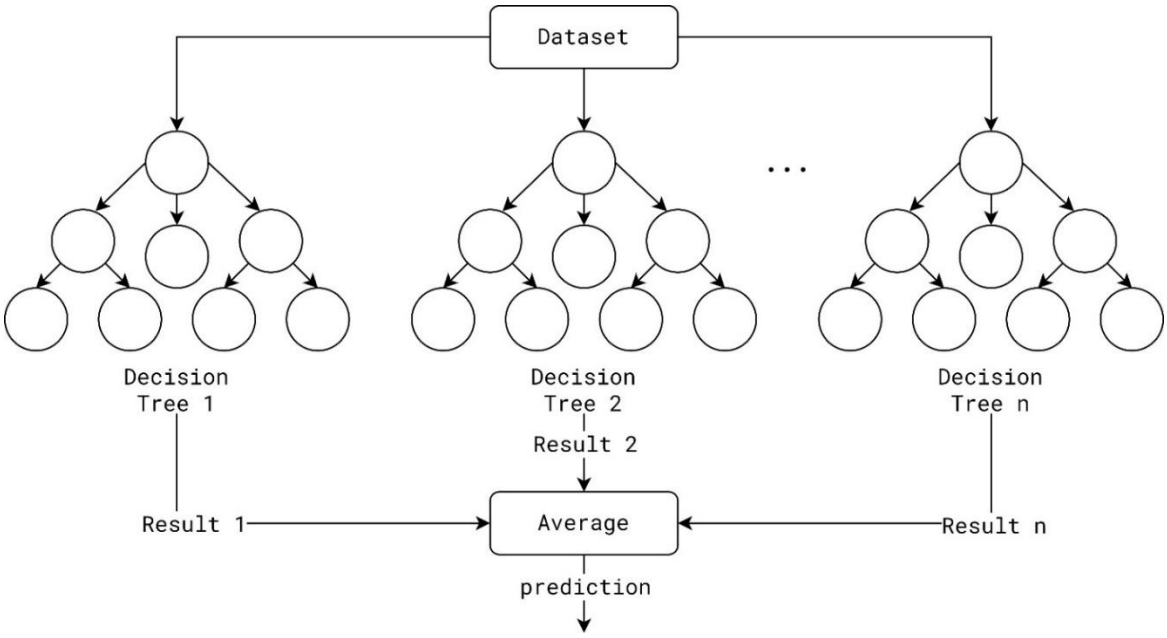

Рис. 1.8. Алгоритм «Випадковий ліс»

4. Рекурентна нейронна мережа (RNN) [8]

На відміну від звичайних нейронних мереж прямого поширення FNN, рекурентні нейронні мережі мають внутрішню «пам’ять»: активаційні значення в нейронах у попередній момент часу t-1 використовуються в якості вхідних значень у поточних момент часу t. Архітектура такої нейронної мережі в розгорнутому вигляді наведена на рис. 1.9:

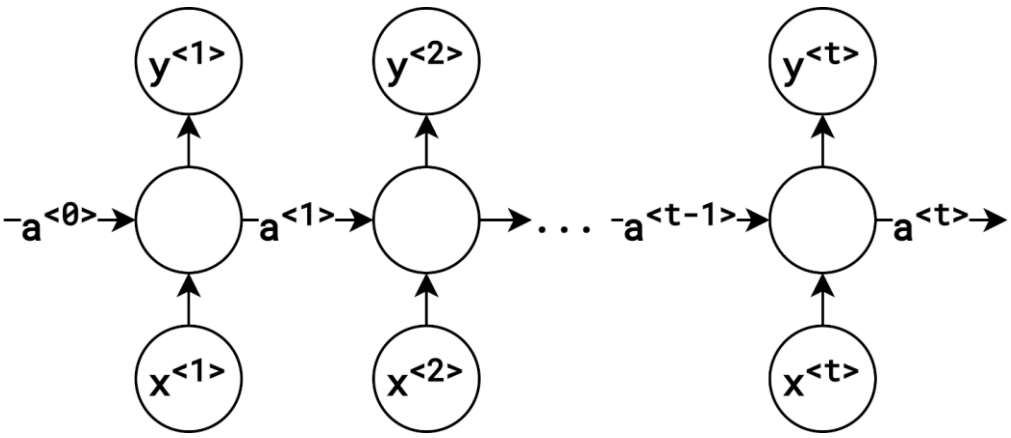

Рис. 1.9. Структура RNN в розгорнутому вигляді

Активації $a^{<t>}$ нейронів внутрішнього шару та вихід нейронної мережі $y^{<t>}$ в поточний момент часу t виражені формулами 1.9 та 1.10 відповідно:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a), \quad (1.9)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y), \quad (1.10)$$

де W_{aa} – коефіцієнти для попередніх активацій $a^{<t-1>}$;

W_{ax} – коефіцієнти для поточних вхідних значень $x^{<t>}$;

W_{ya} – коефіцієнти нейронів вихідного шару;

b_a та b_y – вільні коефіцієнти для нейронів внутрішнього та вихідного шару відповідно;

g_1 та g_2 – активаційні функції для внутрішнього та вихідного нейрона відповідно.

Навчається даний тип нейронних мереж за допомогою алгоритму зворотного поширення помилки з часом (Backpropagation through time, або BPTT).

1.3. Аналіз сучасних програмно-апаратних рішень прогнозування вірогідності виникнення хвороб с/г культур

Аналіз підходів до прогнозування захворювання с/г культур.

Автори Gianni Fenu та Francesca Maridina Malloci використовували метод опорних векторів (SVM) та нейронну мережу прямого поширення (FNN) для класифікації ризику захворювання картоплі (низький, середній та високий) на півдні Сардинії. За допомогою методу опорних векторів вдалося досягти точності класифікації в 98%, а за допомогою нейронної мережі 96%. В якості вхідних даних до моделей використано середню добову вологість повітря (%), середню добову температуру повітря (°C), а також одиниці Blight Units, обчислені за

допомогою математичної моделі SimCast. Погодні дані були зібрані на півдні Сардинії за період з 2016 по 2018 рік [3].

Дослідники Helong Yu, Jiawen Liu, Chengcheng Chen, Ali Asghar Heidari, Qian Zhang, Huling Chen запропонували підхід для класифікації захворювання кукурудзи на основі зображень з використанням алгоритмів K-means та CNN. Запропонована архітектура CNN у поєднанні з K-means для попередньої обробки зображень кукурудзи (k=32) показала точність 93% на тестових даних, що перевершувало результати, отримані за допомогою архітектур VGG-16, VGG-19, ResNet18 та Inception v3 [10].

Дослідники Qingxin Xiao, Weilu Li, Yuanzhong Kai, Peng Chen, Jun Zhang та Bing Wang порівнювали моделі ML, такі як LSTM, SVM, Random Forest та KNN для визначення наявності захворювання / шкідників на бавовні. Вхідні погодні дані: максимальна та мінімальна температура (°C), відносна вологість вранці та ввечері (%), опади (мм), швидкість вітру (км/год), час сонячного світла (год) та випаровування (мм). Модель LSTM показала найкращі результати на тестових даних зі значенням метрики AUC (area under the curve) = 0,97. Кількість часових проміжків (timestamps) для прогнозу = 4 [11].

Дослідники Retuja Rajendra Patil, Sumit Kumar та Ruchi Rani порівнювали моделі машинного навчання, такі як CNN, RNN, FNN, SVM та KNN (k-найближчих сусідів) для класифікації захворювання рослин. Взято до уваги наступні кліматичні параметри: відносна вологість повітря, температура повітря та вологість ґрунту. Найкращі результати показала нейронна мережа прямого поширення FNN: точність (accuracy) = 90,79%, повнота (recall) = 0,915 та влучність (precision) = 0,9909 [5].

На основі розглянутих наукових робіт з обліком фізичних технологій на яких буде реалізовано задачу прогнозування вірогідності появи хвороб с/г культур було встановлено, що найбільш оптимальним є підхід аналізу часових рядів результатів спостережень кліматичних даних. Серед розглянутих підходів прогнозування захворювання с/г культур на основі погодних даних було виявлено, що ML алгоритми показують значну ефективність під час обробки

ґрунтокліматичних даних з метою прогнозування вірогідності захворювання с/г культур [3, 5, 11].

Отже, виникає необхідність обґрунтування оптимального алгоритму (згідно регресійних метрик R^2 та $RMSE$) обробки часових рядів результатів спостережень ґрунтокліматичних параметрів (час вологості листя, температура повітря, вологість повітря та опади) з метою прогнозування ймовірності захворювання кукурудзи в Дніпропетровському регіоні.

Приклади сучасних систем прогнозування захворювання с/г культур на основі польового моніторингу:

1. RICE-GUARD. Проект ЄС від програми FP7 розробив недорогу бездротову мережу сенсорів (WSN) для підвищення репрезентативності метеоданих, які використовуються в системах прогнозування рисового захворювання. Погодні дані є основним фактором для розвитку захворювання і їхній точності часто загрожує розташування метеостанцій поза зонами рисового вирощування. WSN від RICE-GUARD базується на досягненнях технології Інтернету речей (IoT), що дозволяє впровадження бездротових мереж та радіочастотних комунікацій для збору метеоданих у реальному часі в різних частинах поля [7].

2. DSS LANDS (DSS – Decision Support System, LANDS – Laore Architecture Network Developed in Sardinia). Розроблена для фермерів в Сардинії дослідниками Gianni Fenu та Francesca Maridina Malloci для допомоги у прийнятті рішень щодо вирощування наступних с/г культур: цитрус, артишок, пшениця, кукурудза, картопля, оливки, персик та помідор. Дана система виконує наступні задачі: збирає та обробляє погодні дані з метеостанцій, аналізує та інтерпретує інформацію, використовує результати аналізу для рекомендації найкращих рішень щодо вирощування с/г культур. Архітектура даної системи складається з наступних компонент (див. рис. 1.10.) [3]:

- Метеостанції для збору погодних даних: температура (°C), відносна вологість (%), швидкість вітру (км/год), напрямок вітру, кількість опадів (мм) та сонячна радіація (W/m^2)

- Веб-сервер для попередньої обробки та зберігання зібраних з метеостанцій кліматичних параметрів в базі даних, математичного аналізу та інтерпретації даних для рекомендації найкращих рішень щодо вирощування с/г культур

- Кросплатформений застосунок, використовуваний фермерами, для візуалізації результатів аналізу з серверу у вигляді графіків.

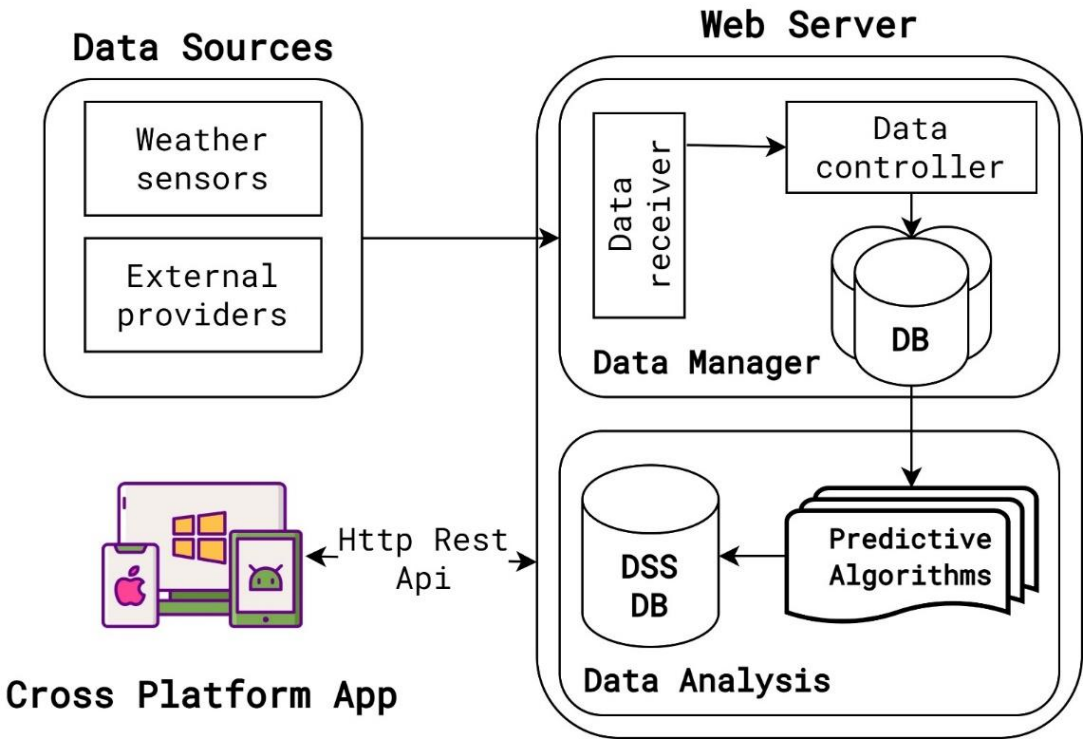

Рис. 1.10. Архітектура системи DSS LANDS

3. FarmBeats [12]. Комплексне рішення на основі IoT та машинного навчання, розроблене Microsoft. Воно використовує поєднання недорогих датчиків, дронів та алгоритмів машинного навчання для моніторингу різних аспектів сільського господарства, включаючи стан ґрунту та посівів. Система збирає дані з датчиків, розміщених у полі, і використовує моделі машинного навчання для прогнозування ймовірності захворювань сільськогосподарських

культур. Microsoft провела успішні пілотні проекти, продемонструвавши покращення прогнозування врожайності та рівня виявлення хвороб.

Деякі ключові особливості, висвітлені в статті [12], включають:

- використання недорогих датчиків, що дозволяє широко використовувати їх на сільськогосподарських полях. Ці датчики збирають дані про стан ґрунту, температуру, вологість та інші важливі параметри.

- використання дронів, що оснащені датчиками, для збору знімків полів з висоти. Ці дані розширюють можливості платформи для ефективного моніторингу великих територій.

- периферійного обчислення (Edge computing) для локальної обробки даних, що зменшує потребу в постійному підключенні. Це має вирішальне значення для віддалених сільськогосподарських районів з обмеженим доступом до інтернету.

- використання алгоритмів ML для аналізу зібраних даних. Ці моделі навчені розпізнавати закономірності та кореляції, що дозволяє прогнозувати різні сільськогосподарські параметри, включаючи хвороби сільськогосподарських культур.

Автори [12] підкреслюють важливість наявності системи прийняття рішень на основі даних у сучасній сільськогосподарській практиці та демонструють отриманими результатами потенційний вплив технологій на підвищення ефективності та сталості сільського господарства. Представляючи успіхи FarmBeats, автори також визнають виклики, з якими зіткнулися під час впровадження, такі як проблеми з надійним інтернет підключенням та потреба у зручному для користувача інтерфейсі.

4. Платформа рішень Watson від IBM [13] інтегрує технології Інтернету речей та штучного інтелекту, щоб надавати практичну інформацію для точного землеробства. Вона аналізує дані з різних джерел, таких як метеостанції, супутникові знімки та датчики ґрунтокліматичних параметрів. Використовуючи машинне навчання, платформа прогнозує ймовірність захворювань сільськогосподарських культур, допомагаючи фермерам приймати обґрунтовані

рішення. Рішення продемонструвало ефективність у прогнозуванні хвороб та ранньому втручанні, що сприяло підвищенню врожайності.

5. CropX [14] – це програмно-апаратне рішення, яке використовує ґрунтові датчики та алгоритми машинного навчання для оптимізації зрошення та прогнозування хвороб сільськогосподарських культур. Система безперервно відстежує стан ґрунту, збираючи дані про рівень вологості та інші важливі параметри. Моделі машинного навчання аналізують ці дані для прогнозування ймовірності виникнення хвороб. CropX отримала визнання завдяки зручному інтерфейсу та можливостям моніторингу в режимі реального часу, що підвищує ефективність управління хворобами сільськогосподарських культур.

В ході аналізу розглянутих програмних систем для моніторингу кліматичних параметрів та прогнозування виникнення с/г хвороб визначені основні архітектурні компоненти таких систем: мережа бездротових датчиків WSN та/або метеостанції в якості джерел отримання кліматичних даних, веб-сервер (попередня обробка, аналіз та машинне навчання), база даних та клієнтський застосунок. В рамках цієї магістерської роботи будуть розглянуті такі складові, як: попередня обробка даних, статистичний аналіз даних та машинне навчання.

1.4. Обґрунтування мети та задач дослідження

Метою дослідження є оцінка ефективності ML алгоритмів для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight та визначення найбільш ефективного (згідно метрик $RMSE$ та R^2) методу, який буде інтегровано в програмний застосунок для надання прогнозів щодо ймовірності захворювання спостережуваних культур в режимі реального часу.

Об’єкт дослідження: інфокомунікаційні процеси збору та обробки даних щодо прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Предмет дослідження: алгоритми ML для прогнозування вірогідності виникнення хвороби кукурудзи Fusarium Head Blight.

Задачі дослідження:

1. Провести аналіз та логічне узагальнення відомих підходів до розробки програмно-апаратного забезпечення систем прогнозування вірогідності виникнення захворювання с/г культур.
2. Обґрунтувати методи та засоби дослідження щодо оцінки ефективності ML алгоритмів для прогнозування вірогідності виникнення захворювання с/г культур.
3. Розглянути лінійну регресію, нейронну мережу FNN та випадковий ліс (Random Forest) для прогнозування ймовірності захворювання хвороби кукурудзи Fusarium Head Blight.
4. Визначити найбільш ефективну регресійну модель згідно метрик $RMSE$ та R^2, а також гіперпараметри цієї моделі при яких отримано найкращі значення на навчальних та тестових даних.
5. Програмно реалізувати алгоритм для прогнозування ймовірності захворювання кукурудзи із регресійною моделлю, яка показала найкращі результати.
6. Виконати кількісну та якісну оцінку отриманих результатів та сформулювати подальші перспективні напрямки досліджень у зазначеній предметній галузі.

1.5. Висновки

1. Зернові культури є найбільш вирощуваними культурами у світі. Пшениця, кукурудза та ячмінь є найбільш вирощуваними зерновими культурами в Східній та Південній Європі, а також в Україні згідно статистика FAO.
2. Зміни навколишнього середовища відіграють найбільшу роль у розвитку захворювання рослин. Тому, для прогнозу ймовірності захворювання

варто враховувати такі кліматичні показники, як температура повітря, відносна вологість повітря, кількість опадів, час вологості листків тощо.

3. Для застосування регресійних моделей, таких як лінійна регресія, нейронна мережа прямого поширення та випадковий ліс для задачі прогнозування часових рядів дані потрібно попередньо обробити так, щоб у вхідних значеннях враховувати не лише поточні вхідні показники у момент часу t, а й попередні разом з минулими значення цільової змінної.

4. Більшість програмних рішень для створення системи прогнозування захворювання певних типів с/г культур та аналізу зібраних даних включають такі складові, як: мережа бездротових датчиків WSN, веб-сервер з алгоритмами ML, база даних та клієнтські застосунки з графічним інтерфейсом.

5. Авторами праць [3, 5, 11] встановлено, що найкращі результати для прогнозування захворювання с/г показують нейронні мережі FNN та LSTM, що в свою чергу потребує додаткового розвитку в контексті прогнозування ймовірності захворювання кукурудзи в Дніпровському регіоні під впливом таких кліматичних факторів, як: відносна вологість повітря, кількість опадів, температура повітря та час вологості листків.

РОЗДІЛ 2. МАТЕРІАЛИ, МЕТОДИ І ПІДХОДИ ДО ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

2.1. Опис обмежень досліджуваної технології

- Тип сільськогосподарських культур: кукурудза.
- Розглянуті хвороби кукурудзи для прогнозування ймовірності захворювання: Fusarium Head Blight.
- Вимірюванні ґрунтокліматичні параметри: температура повітря (°C), відносна вологість повітря (%), кількість опадів (мм) та час зволоження листя (хв.).
- Агрокліматична зона отримання експериментальних даних: північний степ України (посушлива і тепла зона; гідротермічний коефіцієнт становить від 0.7 до 1.0; типова річна сума температур знаходиться в діапазоні від 2900 °C до 3300 °C). Дані були завантажені з метеостанції METOS by Pessl Instruments із використанням IoT платформи FieldClimate, доступ до якої був наданий компанією Metos Ukraine LLC.

Загальні наукові методи дослідження:

- Лінійна регресія;
- «Випадковий ліс»;
- Нейронна мережа прямого поширення.

Дослідження проведено у хмарному середовищі Google Colab на мові програмування Python 3.10.12. Причиною вибору даної мови програмування для проведення дослідження є наявність величезної кількості зручних та добре задокументованих бібліотек для машинного навчання, статистичного аналізу даних, візуалізації даних, виконання операцій лінійної алгебри тощо. Додатковим критерієм при виборі мови програмування Python стала її поширеність використання для вирішення вище описаного кола задач.

Для досягнення поставленої мети використані наступні бібліотеки мови Python:

- numpy (для виконання операцій лінійної алгебри);
- pandas (для проведення статистичного аналізу даних);
- matplotlib (для побудови графічних зображень та діаграм);
- keras (для створення та оцінки ефективності моделей ML);
- sklearn (для створення моделей лінійної регресії та Random Forest);
- fast-ml (для поділу даних на навчання, валідацію та тестування).

2.2. Узагальнена структурно-алгоритмічна організація досліджуваної технології

Як вже було досліджено в розділі 1, сучасні програмні системи прогнозування ймовірності захворювання с/г культур на основі погодних даних складаються з таких архітектурних складових, як бездротова мережа датчиків WSN, веб-сервер, база даних та клієнтський застосунок. Приклад архітектури наведено на рис. 2.1.

На рис. 2.1 показано, що взаємодія з сервером відбувається через захищений протокол HTTPS. При цьому веб-сервер надає програмний інтерфейс API, до якого можуть підключатися інші компоненти системи за тим же протоколом.

До цього інтерфейсу підключені клієнтський застосунок і мережа датчиків WSN, яка надсилає POST-запити з кліматичними даними для внесення нових показників у базу даних. Після прийому та збереження цих даних, вони використовуються для створення моделей ML.

Для прогнозування ймовірності виникнення захворювань с/г культур, на сервері використовується найбільш оптимальна модель згідно визначених регресійних метрик. Клієнтський застосунок аналізує зібрані дані та використовує прогнози, що отримані з цієї моделі.

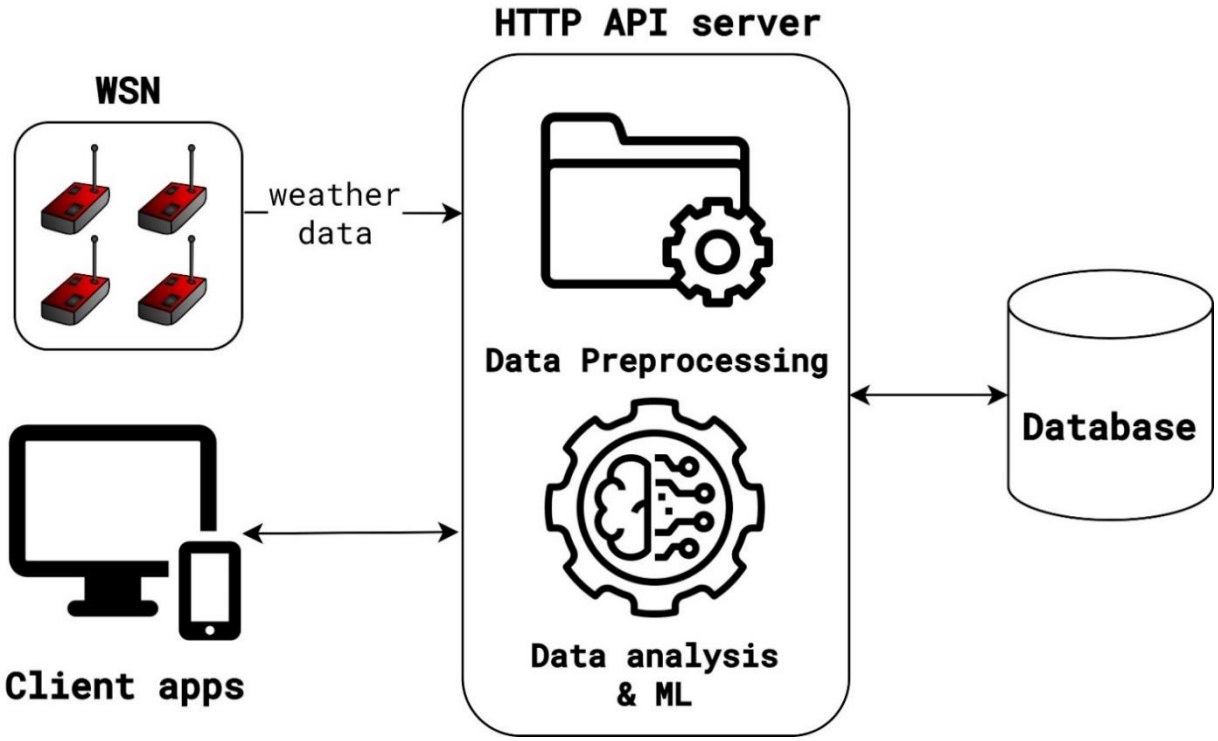

Рис. 2.1. Типові складові архітектури програмної системи прогнозування ймовірності захворювання с/г культур на основі погодних даних

Важливо відзначити, що при взаємодії з сервером клієнт отримує дані не у вигляді згенерованої HTML-сторінки, а в форматі JSON. Такий спосіб передачі даних забезпечує можливість використання цього інтерфейсу не лише веб-додатками, але і мобільними пристроями.

В рамках цієї магістерської роботи детально розглянуто саме підхід для виконання прогнозу, що включає такі етапи, як: статистичний аналіз даних, попередня обробка даних, навчання та оцінка моделей ML з метою вибору найбільш оптимального підходу для прогнозування ймовірності виникнення фузаріозу кукурудзи.

2.3. Методика проведення досліджень

Кліматичні дані та ймовірність захворювання Fusarium Head Blight кукурудзи зібрані погодинно за період з вересня 2022 року по вересень 2023 року. Зібрані дані включали в себе наступні характеристики:

- Datetime. Дата та час збору кліматичних параметрів з датчиків. Кожне наступне значення на годину більше за попереднє;
- Air temperature. Температура повітря (°C);
- Precipitation. Кількість опадів (мм);
- Humidity. Вологість повітря (%);
- Leaf wetness duration. Час зволоження листя (хв);
- Infection probability. Ймовірність ризику захворювання Fusarium Head Blight (%) у поточний момент часу.

Оскільки характеристика «Datetime» не містить корисної інформації для прогнозування ймовірності захворювання кукурудзи, то її не буде враховано в якості вхідного значення у досліджуваних моделях ML.

В табл. 2.1 наведені статистичні показники, такі як: середнє значення, стандартне відхилення, медіана, мінімальне значення та максимальне значення для зібраних кліматичних параметрів та ймовірності захворювання (див. табл. 2.1):

Таблиця 2.1

Статистичні показники зібраних даних

	Температура повітря (°C)	Вологість повітря (%)	Опади (мм.)	Час вологості листків (хв.)	Ймовірність захворювання (%)
Кількість	8658				
Середнє	10,62	72,8	0,07	8,7	4,84
Стандартне відхилення	9,74	17,32	0,34	21,13	17,61
Медіана	9,65	76	0	0	0
Мін.	-10,9	19	0	0	0
Макс.	37	100	14	60	100

Виходячи зі статистики, наведеної в табл. 2, кліматичні показники та знаходяться в межах норми для розглянутої агрокліматичної зони (немає аномально низьких або великих значень, наприклад, негативної або більшої за 100% відносної вологості повітря або ймовірності захворювання).

Для аналізу взаємозв'язку кліматичних показників створено кореляційну матрицю (див. рис. 2.2):

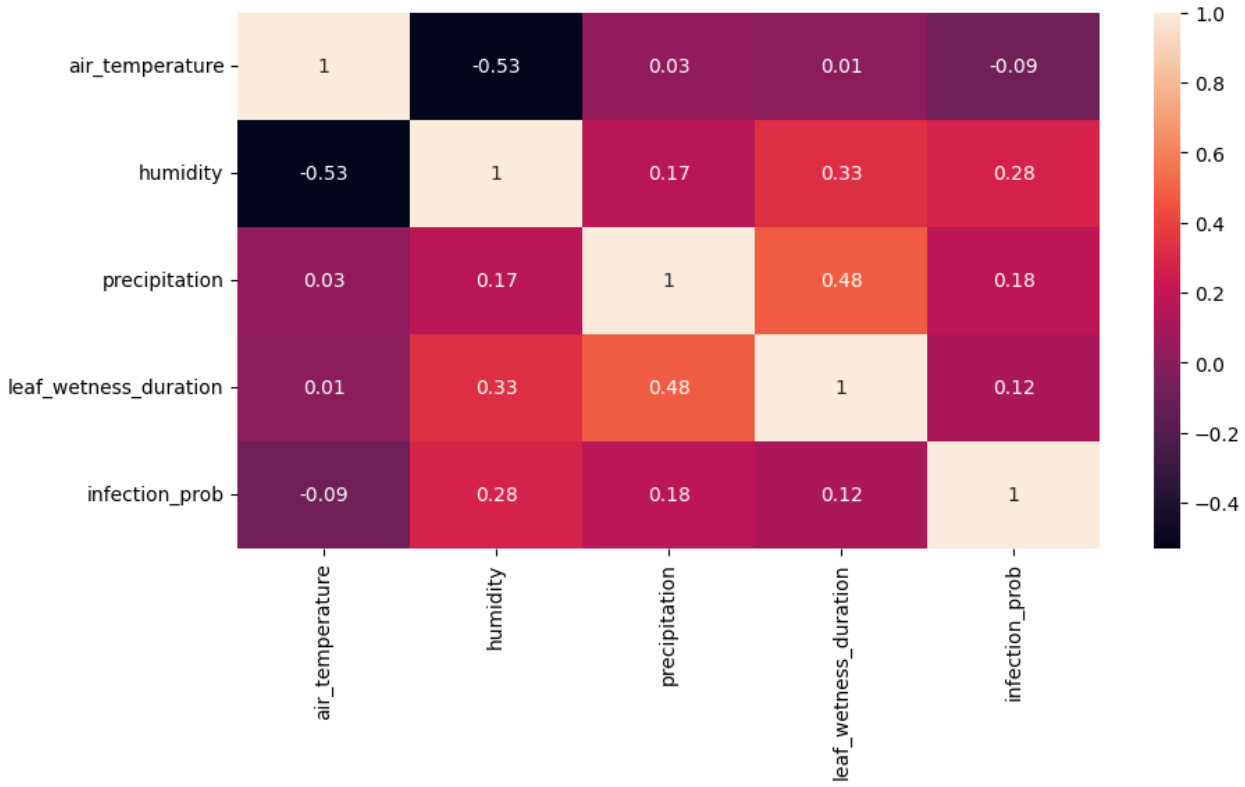

Рис. 2.2. Матриця кореляцій

Значення коефіцієнта кореляції від -1 до 1 відображає ступінь взаємозв'язку між двома змінними. Значуща залежність визначається як абсолютне значення коефіцієнта кореляції більше 0,5. Негативний коефіцієнт кореляції означає обернену залежність між характеристиками. Один з найбільших коефіцієнтів кореляції (0,48) спостерігається між кількістю опадів та тривалістю вологості листя на рис. 2.2. Від'ємна кореляція -0,53 між відносною вологістю повітря та температурою свідчить про тенденцію зменшення температури при збільшенні вологості. Найбільший вплив на ймовірність

захворювання має відносна вологість повітря з коефіцієнтом кореляції 0,28. Невисокі значення кореляцій кліматичних показників із цільовою змінною (ймовірністю виникнення фузаріозу кукурудзи) не перекреслюють вплив цих факторів, оскільки матриця кореляції враховує лише значення у поточний момент часу.

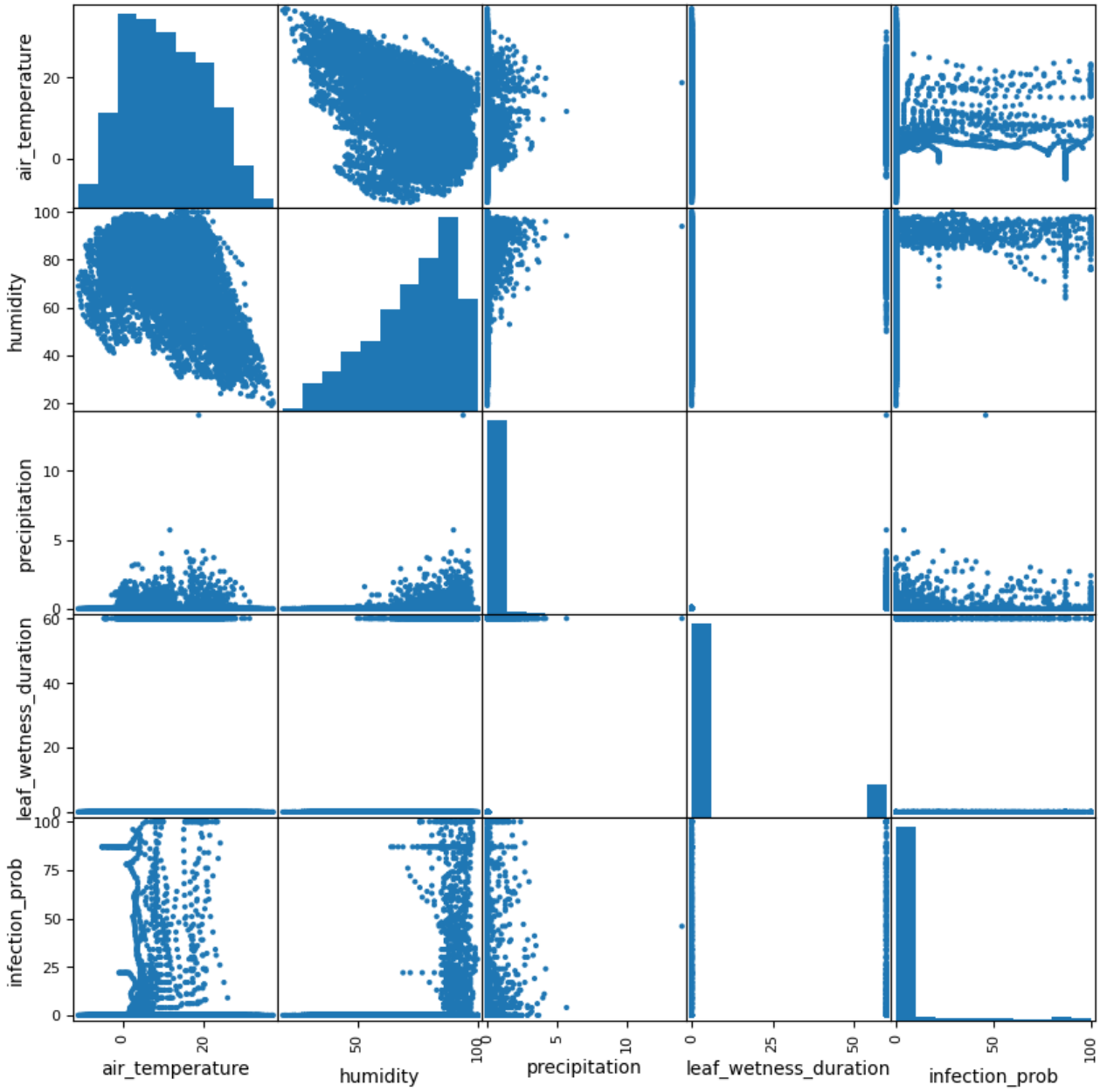

Рис. 2.3. Матриця розсіювання кліматичних даних та ймовірності захворювання

На рис. 2.3 наведена матриця розсіювання з гістограмами розподілу кліматичних показників та ймовірностей захворювання, а також точкових

графіків, що показують залежність змін одного кліматичного показника від іншого. З наведених графіків видно, що більшість додатних та високих значень ймовірностей ризику захворювання відповідала від'ємним та низьким температурам від 0 до 20 °C. Також майже всі значення додатних ймовірностей захворювання відповідали відносній вологості повітря (від 85% і більше).

На рис. 2.4, 2.5, 2.6 та 2.7 наведені графіки зміни зібраних кліматичних показників (температури повітря, часу зволоження листя, відносної вологості повітря, та кількості опадів відповідно) за кожну годину за період з вересня 2022 року по вересень 2023 року.

Температурні дані на рис. 2.4 демонструють властивості сезонності: значення температур, зареєстровані в вересні 2023 р. приблизно дорівнюють значенням в вересні 2022 р. (що і не дивно, оскільки дані зібрані з одного регіону). До того ж, гістограма розподілу температур в матриці розсіювання на рис. 2.3 нагадує нормальний розподіл. Ці властивості числових значень температур будуть враховані при попередній обробці даних.

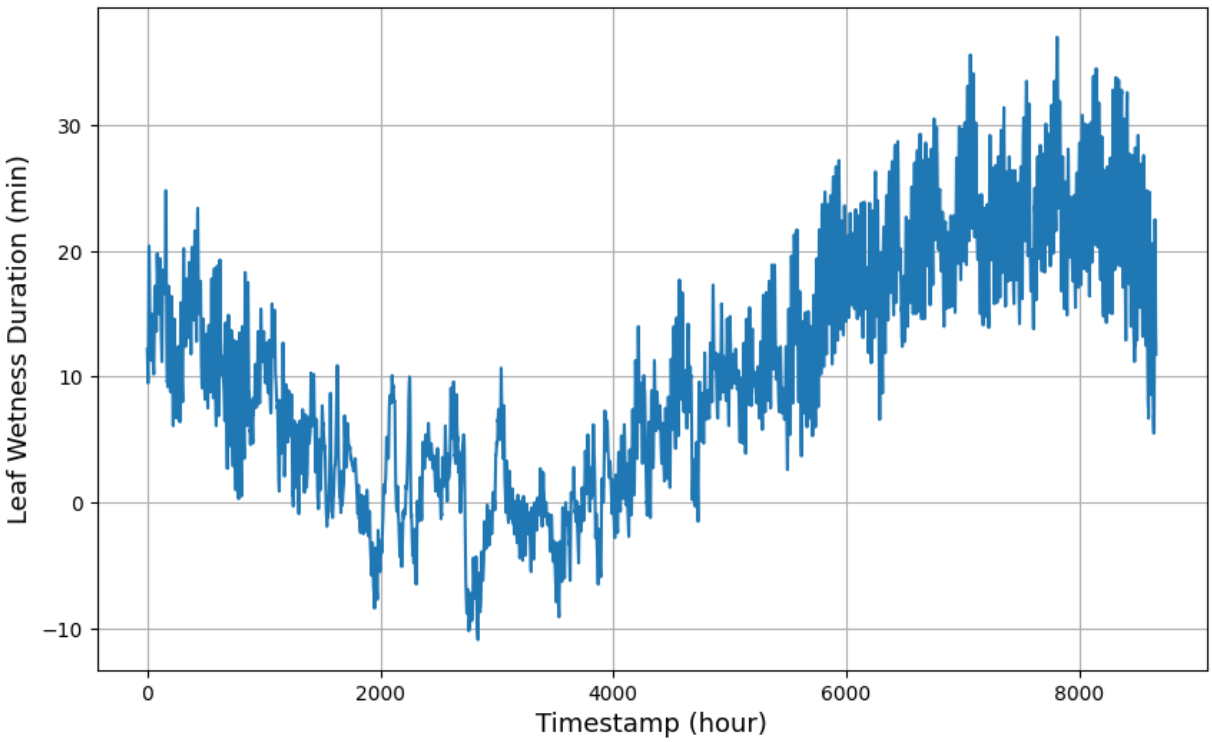

Рис. 2.4. Графік зміни температури повітря за кожну годину

Як можна побачити з точкового графіка для часу зволоження листя на рис. 2.5, для даного кліматичного показника в наведених даних були присутні лише два можливих значення – 0 або 60 хв. Це можна інтерпретувати як: чи було листя вологе або ні за останню годину. Для спрощення, значення 60 даного показника можна перетворити в одиниці перед тим, як подати на вхід моделі ML.

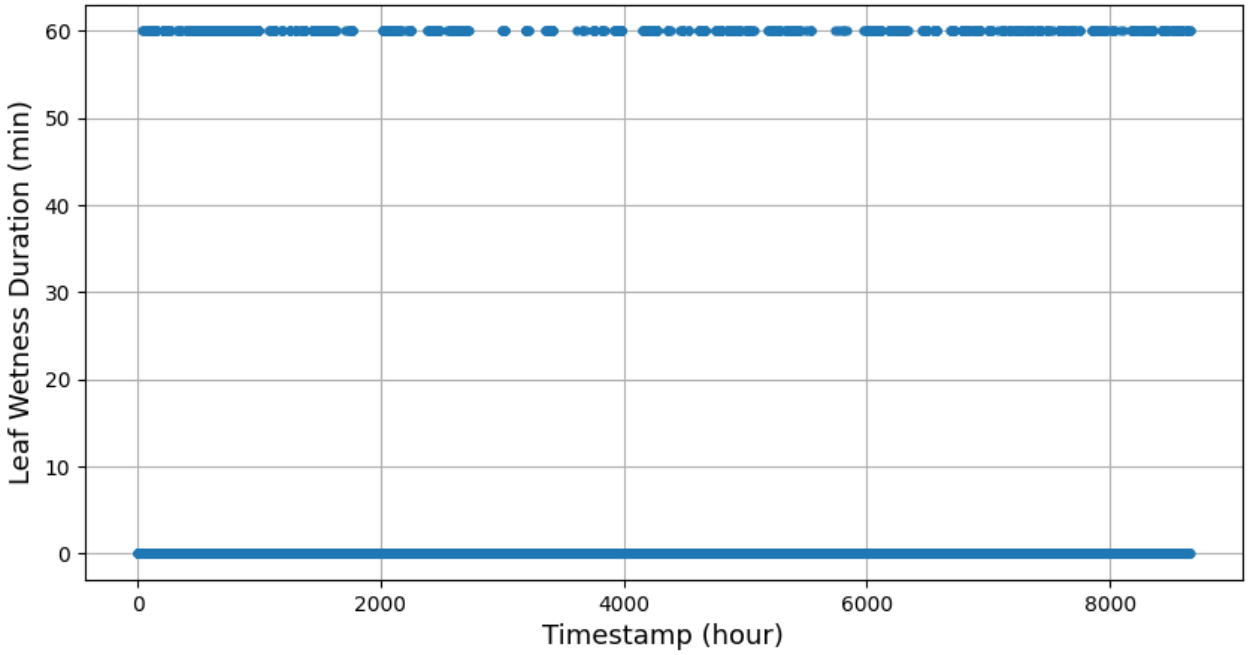

Рис. 2.5. Графік часу зволоження листя за кожну годину

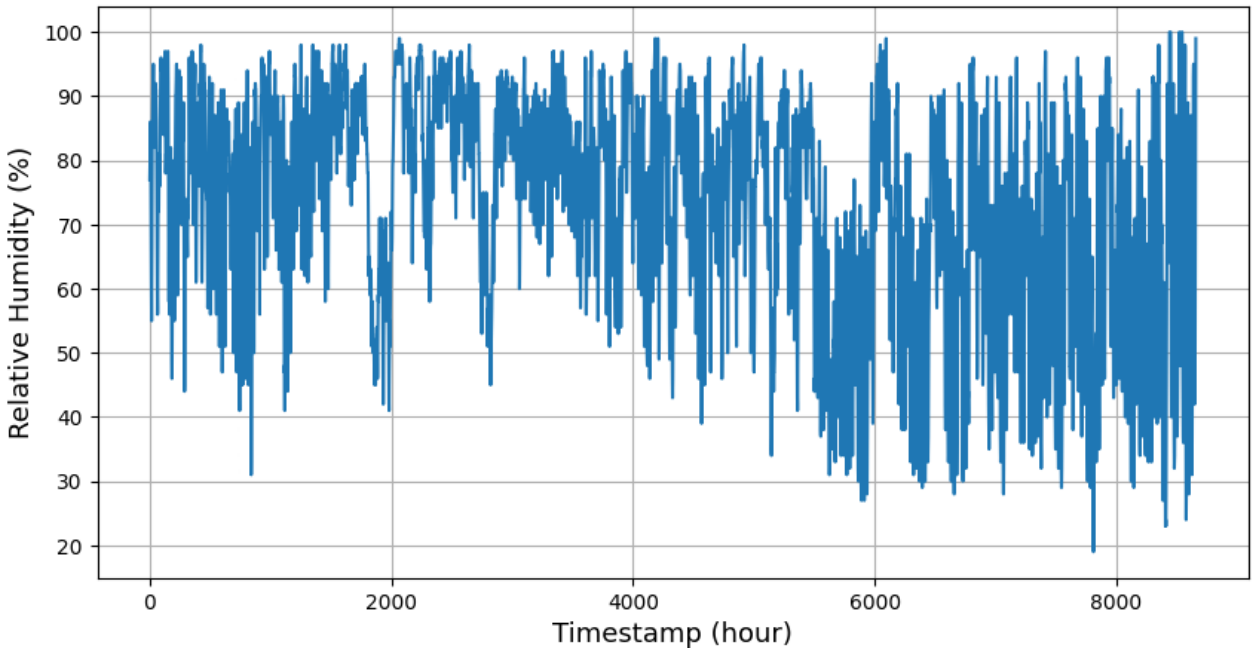

Рис. 2.6. Графік зміни відносної вологості повітря за кожну годину

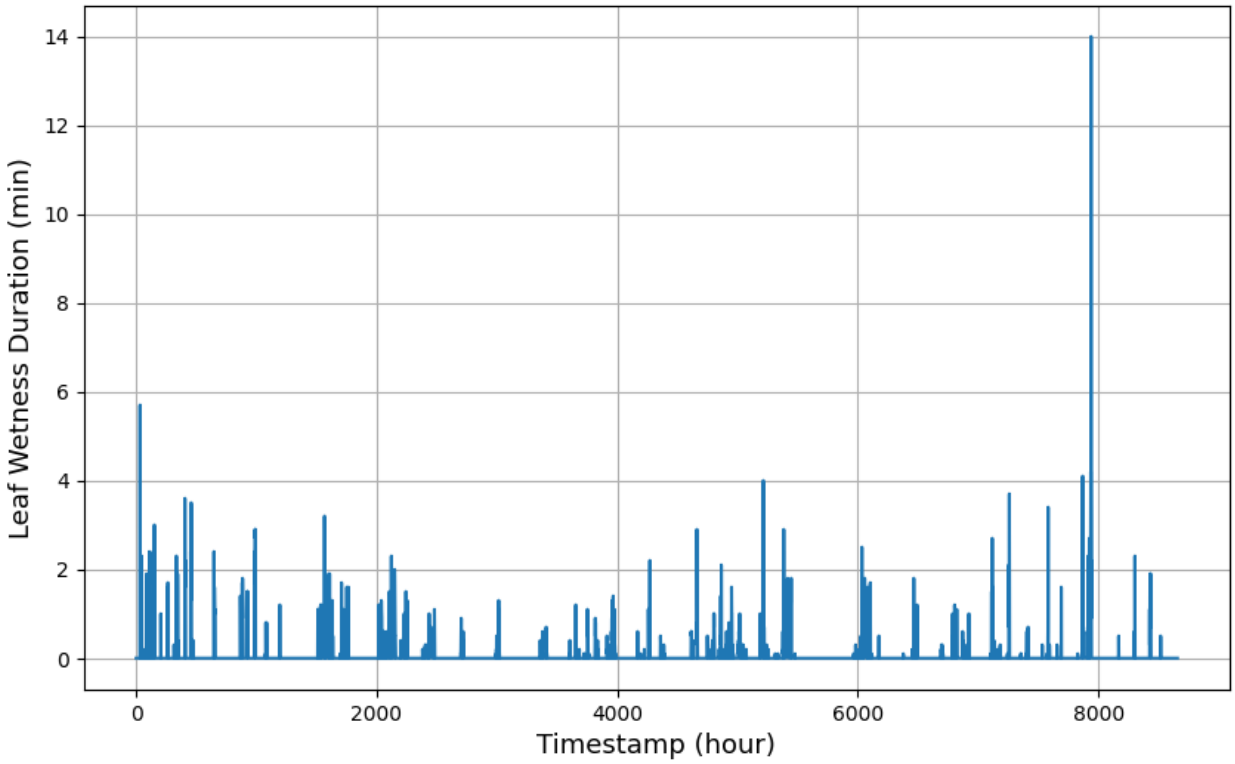

Рис. 2.7. Графік кількості опадів за кожну годину

Точковий графік зі збільшенням ймовірностей захворювання з плином часу наведено на рис. 2.8. Як можна побачити з наведених ймовірностей, лише невелика частина рядків (а саме 906 із 8658, як підраховано) містила додатні ймовірності захворювання. Для цих даних окремо пораховано середнє значення, стандартне відхилення, медіана, максимальне та мінімальне значення для кожного числового показника. Результати представлені в табл. 2.2.

Таблиця 2.2

Статистичні показники даних з додатною ймовірністю захворювання

	Температура повітря (°C)	Вологість повітря (%)	Опади (мм.)	Час вологості листків (хв.)	Ймовірність захворювання (%)
Кількість	906				

Продовження таблиці 2.2

Середнє	7,58	90,83	0,39	21,78	46,31
Стандартне відхилення	6,32	5,06	0,85	28,87	32,32
Медіана	6,2	92	0	0	41
Мін.	-5,1	64	0	0	1
Макс.	25,8	100	14	60	100

Зі значень в таблиці 2.2 можна виділити більш високі середні значення вологості повітря (90,8% проти 72,8% в табл. 2.1), а також більш низьке значення стандартного відхилення (лише 5% порівняно з 17,32% в табл. 2). Середнє значення часу вологості листків також більше (21,78 хв. проти 8,7 хв. в табл. 2.1). Крім того, помічені більш низькі значення температур (середнє значення 7,58°C при більш низькому стандартному відхилені 6,32°C).

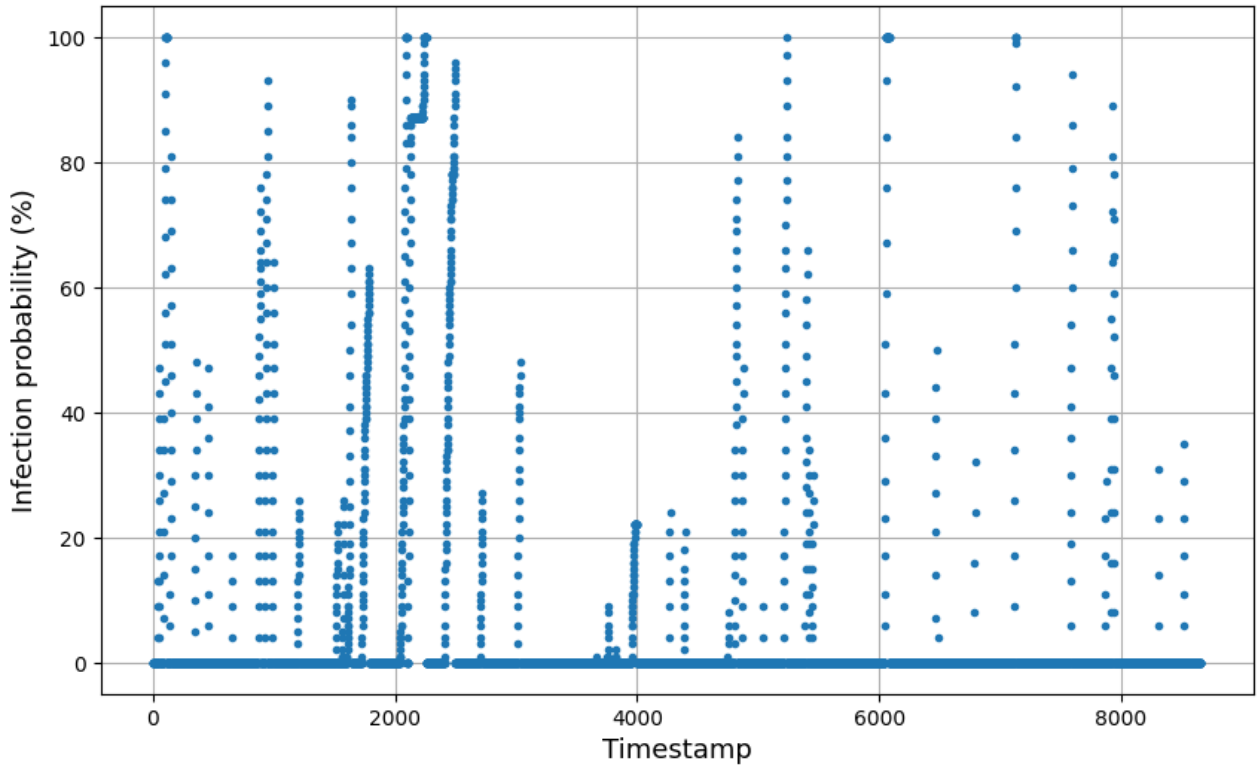

Рис. 2.8. Зміна ймовірності захворювання Fusarium Head Blight з часом

В зібраних даних підраховано 45 підпослідовностей де ймовірність захворювання поступово зростала. З цих 45-ти підпослідовностей зі зростанням ймовірності захворювання знайдено найдовший за часом проміжок, що тривав 152 години.

Для знаходження k найдовших послідовностей було створено функцію get_top_k_largest_sequences на мові Python. В якості параметрів функція приймає тип DataFrame з бібліотеки pandas, та ціле число k – кількість найдовших послідовностей. Функція повертає k найдовших послідовностей у порядку спадання їх довжини:

```
import pandas as pd

def get_top_k_largest_sequences(
        df: pd.DataFrame, k: int = 1
) -> list[list[pd.Series]]:
  if len(df) == 0:
    return []

  current_sequence = [df.iloc[0]]
  sequences = [current_sequence]
  for i in range(1, len(df)):
    row = df.iloc[i]

    current_index = df.index[i]
    prev_index = df.index[i - 1]
    index_difference = current_index - prev_index

    if index_difference == 1:
      current_sequence.append(row)
      continue

    current_sequence = [row]

    sequences.append(current_sequence)

  return sorted(sequences, key=len, reverse=True)[:k]
```

Інтерес представляють значення та поступова зміна кліматичних показників та відповідний ріст ймовірності виникнення захворювання Fusarium Head Blight на цьому проміжку часу. Зміна кліматичних показників на цьому

проміжку часу наведена на рис. 2.9, а відповідні ймовірності захворювання наведені на рис. 2.10. З цих графіків можна відмітити низькі значення температур повітря (від -5 до 10°C), а також високі значення вологості повітря, які досягали значень більше за 90%.

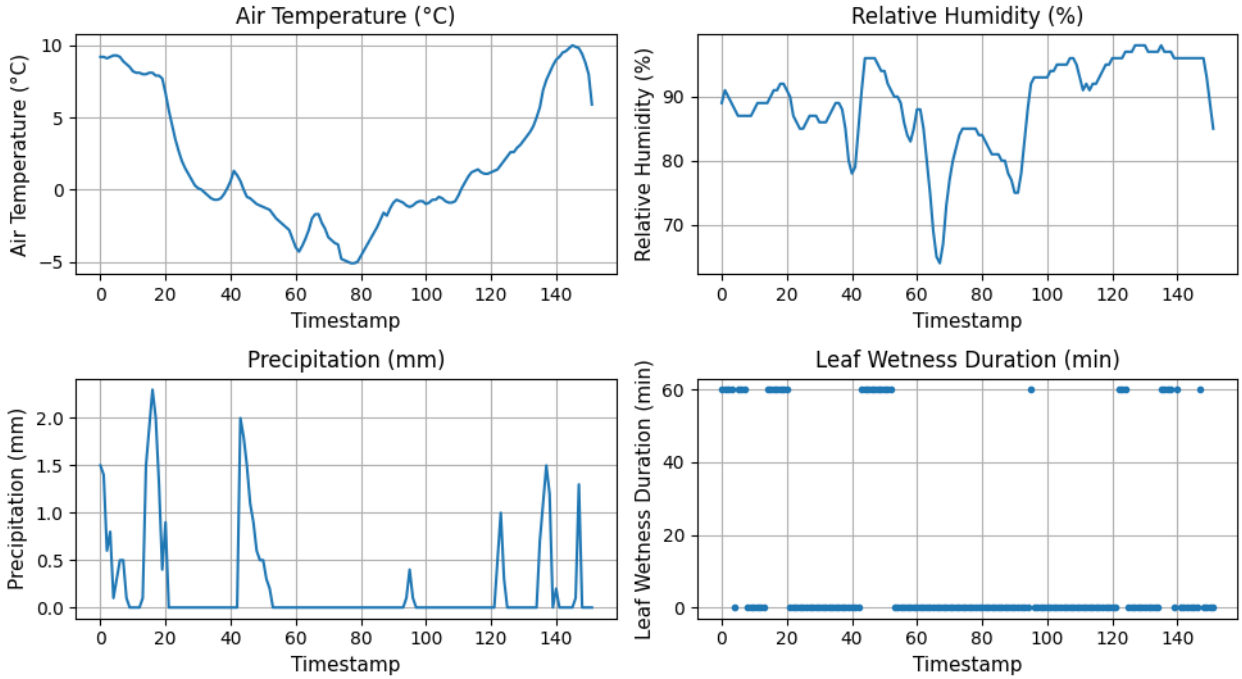

Рис. 2.9. Зміна кліматичних показників на найдовшому проміжку за часом (152 години), де ймовірність захворювання поступово зростала

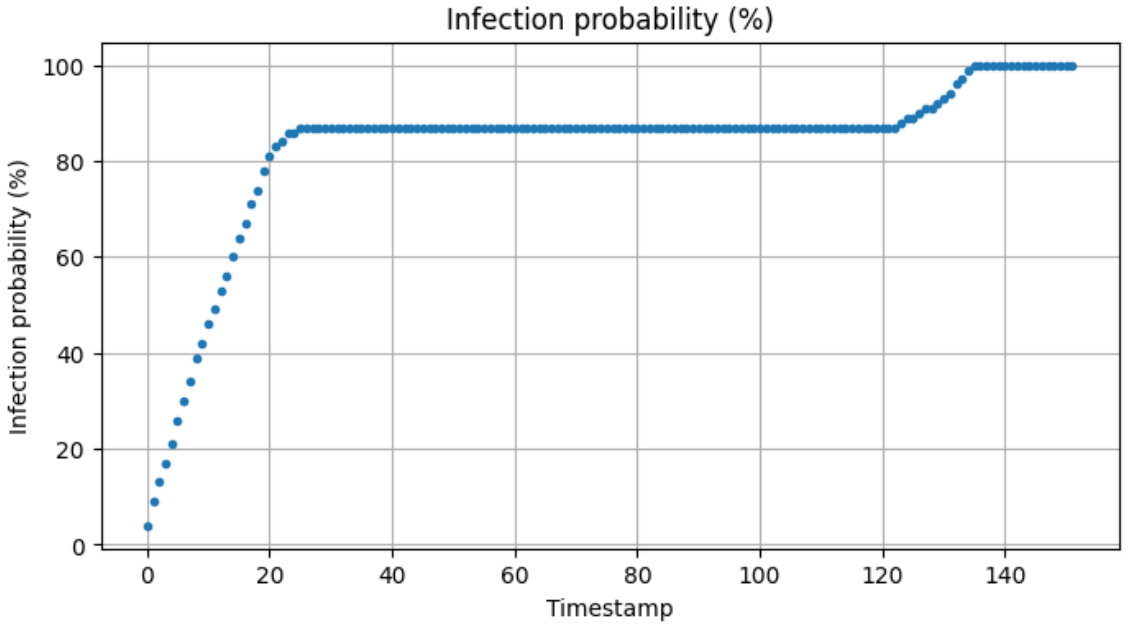

Рис. 2.10. Графік зростання ймовірності захворювання протягом 152 год

Для того, щоб врахувати історію змін кліматичних показників та ймовірностей захворювань введено додатковий гіперпараметр $timestamps$, який необхідно емпірично визначити для кожної досліджуваної моделі ML. Даний гіперпараметр визначає скільки попередніх кліматичних параметрів та ймовірностей захворювань додатково використовувати в якості вхідних значень. Наприклад, для значення $timestamps$=1, поточні кліматичні параметри разом з кліматичними показниками та ймовірністю захворювання годину тому використовуються в якості вхідних значень. Якщо ж встановити значення $timestamps$ =2, то додатково в якості вхідних значень використовуються кліматичні параметри та ймовірність захворювання 2 години тому.

Нижче наведено код функції add_previous_timestamps_values на мові Python для додавання попередніх значень кліматичних показників та ймовірностей захворювання. В якості параметрів функція приймає екземпляр класу DataFrame з бібліотеки pandas та значення гіперпараметра $timestamps$. Функція повертає новий DataFrame з минулими кліматичними показниками та ймовірностями захворювання за останні $timestamps$ годин.

```
import pandas as pd
from tqdm import tqdm

def add_previous_timestamps_values(
  df: pd.DataFrame, timestamps: int = 2
) -> pd.DataFrame:
  df_with_timestamps = pd.DataFrame()
  columns = df.columns.tolist()

  for index, _ in tqdm(
    df.iterrows(),
          total=len(df),
          desc="Processing Rows",
          ncols=100
  ):
    if index < timestamps:
      continue

    for column in columns:
      df_with_timestamps.at[index, column] = df.at[
                  index, column
```

```
                ]

        for t in range(1, timestamps + 1):
            df_with_timestamps.at[index, f"{column}_{t}"] = df.at[
                index - t, column
            ]

    return df_with_timestamps.drop(timestamps)
```

Окремі моделі машинного навчання, такі як нейронні мережі, навчаються за допомогою алгоритму градієнтного спуску, який досягає збіжності швидше, якщо вхідні дані однаково масштабовані [15]. Враховуючи розподіл даних наведений в табл. 2.1 та рис. 2.1, вхідні дані були трансформовані наступним чином:

- час зволоження листя (хв.) поділено на максимальне значення 60;
- відносна вологість повітря (%) поділено на максимальне значення 100;
- ймовірність захворювання (%) поділено на максимальне значення 100;
- кількість опадів (мм) та температура повітря (°C) стандартизовані за формулою (див. формулу 2.1) [15]:

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2.1}$$

де x' – стандартизоване значення числової характеристики x;

$\bar{x}$ – середнє значення на навчальній вибірці;

σ – стандартне відхилення на навчальній вибірці.

Дані для навчання, валідації та тестування були поділені у наступному співвідношенні: 70% (або 6058 рядків) на навчання, 15% (або 1298 рядків) на валідацію та 15% (решта 1299 рядків) на тестування. Оскільки моделі машинного навчання схильні до перенавчання, то виникає необхідність у використанні валідаційного набору даних для нейтральної оцінки роботи моделі та налаштування гіперпараметрів. Основна мета використання валідаційних даних

полягає в запобіганні перенавчанню, тоді як тестовий набір даних використовується для остаточної оцінки ефективності роботи моделі.

На рис. 2.11 представлена блок-схема методології навчання та вибору оптимальної моделі ML (згідно метрик R^2 та $RMSE$) для прогнозування захворювання Fusarium Head Blight кукурудзи.

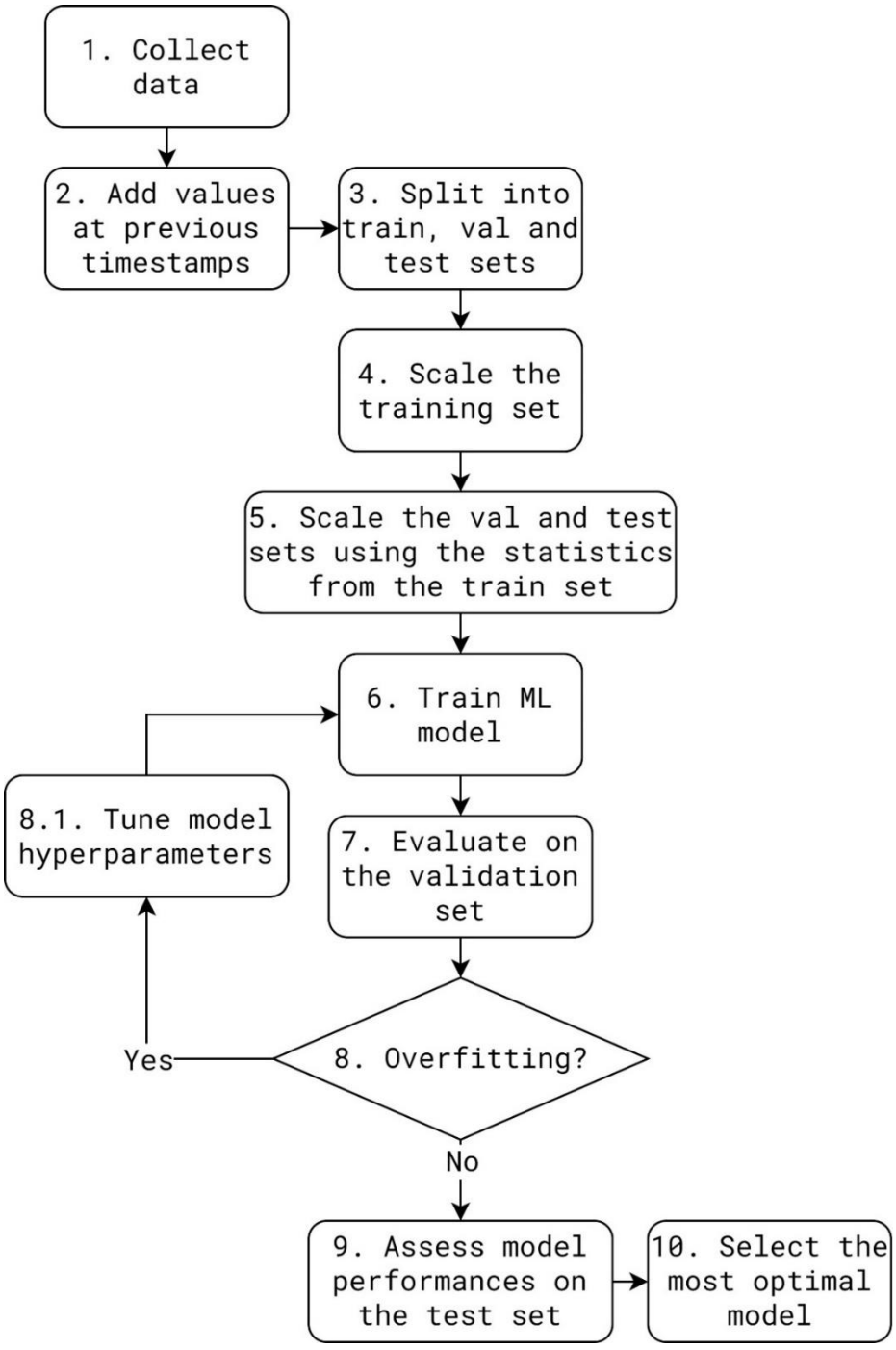

Рис. 2.11. Блок-схема використаної методології для отримання оптимальної моделі прогнозу захворювання Fusarium Head Blight кукурудзи

Пояснення кожного кроку алгоритму вибору оптимальної моделі ML за блок схемою 2.11 наведено нижче:

1. Зібрати дані (8658 записів з кліматичними даними та ймовірностями виникнення фузаріозу погодинно з метеостанції METOS by Pessl Instruments);

2. Додати значення за попередні проміжки часу (визначається гіперпараметром $timestamps$);

3. Розбити дані на навчальну, валідаційну та тестову вибірки (як вже було зазначено, поділ у співвідношення 70:15:15 відповідно);

4. Масштабувати навчальні дані: температуру повітря та кількість опадів за формулою 2.1, а відносну вологість, час зволоження листя та ймовірність захворювання за максимальним значенням – $x' = x \cdot \max(x)^{-1}$;

5. Масштабувати валідаційні та тестові дані з використанням статистики з навчальних даних: використати середні значення та стандартні відхилення кліматичних показників з навчальних даних при стандартизації температури та кількості опадів, а також максимальні значення для нормалізації відносної вологості повітря, часу зволоження листя та ймовірності захворювання;

6. Навчити модель ML використовуючи тренувальні дані. Обчислити метрики R^2 та $RMSE$ на навчальних даних;

7. Оцінити роботу моделі ML на валідаційних даних згідно метрик R^2 та $RMSE$;

8. Перевірити чи модель перенавчилась. Якщо значення метрик $RMSE$ та R^2 значно гірші на валідаційних даних (наприклад, коли R^2 на 0.1 менше або $RMSE$ на 4-5% відсотків більше ніж на навчальній вибірці), то:

   8.1. Повторно налаштувати гіперпараметри моделі ML;

   8.2. Перейти до кроку 6 алгоритму;

9. Оцінити якість моделі (згідно метрик R^2 та $RMSE$) на тестових даних;

10. Обрати найбільш оптимальну модель ML серед лінійної регресії, нейронної мережі прямого поширення та Random Forest згідно метрик R^2 та $RMSE$ для прогнозування фузаріозу кукурудзи.

2.4. Висновки

1. Зібрано 8658 записів з агрокліматичними даними за кожну годину за період з вересня 2022 року по вересень 2023 року для Дніпропетровського регіону за допомогою метеостанції METOS by Pessl Instruments із використанням IoT платформи FieldClimate, доступ до якої був наданий компанією Metos Ukraine LLC.

2. Серед вимірюваних кліматичних показників у наданих даних були присутні: температура повітря (°C), кількість опадів (мм), час зволоження листя (хв.) та відносна вологість повітря.

3. Для враховування історії змін кліматичних параметрів та ймовірностей виникнення захворювання Fusarium Head Blight введено додатковий гіперпараметр $timestamps$, який потребує додаткового налаштування. Даний гіперпараметр визначає скільки кліматичних показників, а також попередніх ймовірностей захворювання за останні $timestamps$ год. потрібно додатково використати в якості вхідних значень.

4. Для навчання виділено 6058 записів (або 70% від всіх даних), на валідацію моделей ML 1298 записів (або 15%), а на тестування – 1299 рядків або 15%.

5. Для пришвидшення тренування моделей ML, що використовують для навчання градієнтний спуск, вхідні дані були масштабовані: кількість опадів та температура повітря стандартизовані за співвідношенням $x' = (x - \bar{x})\sigma^{-1}$, відносна вологість повітря, час зволоження листя та ймовірність виникнення захворювання були нормалізовані до проміжку $[0; 1]$ згідно свого максимального значення.

6. Сформовано підхід для визначення найбільш оптимального методу ML прогнозування ймовірності виникнення захворювання кукурудзи Fusarium Head Blight в залежності від кліматичних даних.

РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ML АЛГОРИТМІВ ПІД ЧАС ПРОГНОЗУВАННЯ ВІРОГІДНОСТІ ВИНИКНЕННЯ ХВОРОБ С/Г КУЛЬТУР

3.1. Регресійні метрики для оцінки якості моделей ML

Для навчання моделей ML використані бібліотеки keras (для тренування нейронної мережі прямого поширення) та sklearn (для тренування лінійної регресії та алгоритму Random Forest) мови програмування Python. Для оцінки якості роботи даних моделей використані метрики R^2 та $RMSE$, наведені в формулах 3.1 та 3.2 відповідно:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}, \tag{3.1}$$

$$\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}}, \tag{3.2}$$

де m – розмір вибірки даних;

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

y_i та $\hat{y}_i$ – очікуване та отримане з моделі значення цільової змінної для i-того екземпляру в даних відповідно.

Коефіцієнт детермінації R^2 чисельно показує, яка частина варіації цільової змінної пояснена моделлю. В найкращому випадку, коли очікувані значення співпадають з прогнозованими, значення R^2 =1. Для моделі, яка прогнозує значення цільової змінної близькі до середнього значення $\bar{y}$, R^2 буде прямувати до нуля.

3.2. Результати дослідження лінійної регресії

В процесі навчання моделі лінійної регресії (згідно формули 1.3 першого розділу) визначено, що найкращі результати для даної моделі отримані для значення гіперпараметра $timestamps$ =3, тобто, разом з поточними кліматичними показниками додатково враховуються попередні кліматичних показники за останні 3 години разом з ймовірностями захворювання. Коефіцієнти створеної моделі лінійної регресії наведені у формулі 3.3:

$$\begin{aligned}\hat{y} = {} & 3{,}15 \cdot temp_0 + 0{,}14 \cdot temp_1 - 1{,}8 \cdot temp_2 - 1{,}43 \cdot temp_3 + \\ & 20{,}73 \cdot rh_0 - 19{,}52 \cdot rh_1 + 5{,}57 \cdot rh_2 - 4{,}99 \cdot rh_3 + \\ & 0{,}5 \cdot pre_0 - 0{,}02 \cdot pre_1 + 0{,}02 \cdot pre_2 - 0{,}13 \cdot pre_3 + \\ & 0{,}4 \cdot lwd_0 - 0{,}13 \cdot lwd_1 - 0.64 \cdot lwd_2 + 0{,}61 \cdot lwd_3 + \\ & 98{,}02 \cdot inf\_p_1 - 0{,}73 \cdot inf\_p_2 - 1{,}92 \cdot inf\_p_3 - 1{,}14,\end{aligned} \tag{3.3}$$

де $\hat{y}$ – ймовірність виникнення захворювання Fusarium Head Blight;

$temp_t$ – температура повітря t годин тому;

rh_t – відносна вологість повітря t годин тому;

pre_t – кількість опадів t годин тому;

lwd_t – час зволоження листя t годин тому;

inf_p_t – ймовірність захворювання t годин тому.

Значення індексу t=0 означає поточний момент часу.

Як можна побачити з формули 3.1 лінійної регресії найбільш впливовими чинниками згідно значень коефіцієнтів при розрахунку поточної ймовірності захворювання є ймовірність захворювання годину тому (відповідний коефіцієнт 98,02), поточна відносна вологість повітря (з коефіцієнтом 20,73) та вологості повітря за останні 3 години (коефіцієнти -19,52, 5,57 та -4,99 відповідно), а також поточна температура повітря (з коефіцієнтом 3,15).

Результати, отримані лінійною регресією для навчальних, валідаційних та тестових даних наведені в табл. 3.1.

Таблиця 3.1

Результати лінійної регресії

	Навчальні дані	Валідаційні дані	Тестові дані
R^2	0,93	0,96	0,96
$RMSE$	4,56	4,4	3,61

На рис. 3.1 наведено точковий графік справжніх та спрогнозованих лінійною регресією ймовірностей виникнення захворювання Fusarium Head Blight. Вісь x відповідає за спрогнозоване моделлю ML значення, а вісь y – за справжнє значення ймовірності. Чим прогнозовані значення ймовірностей ближчі до справжніх, тим ближче вони до червоної пунктирної лінії на рис. 3.1.

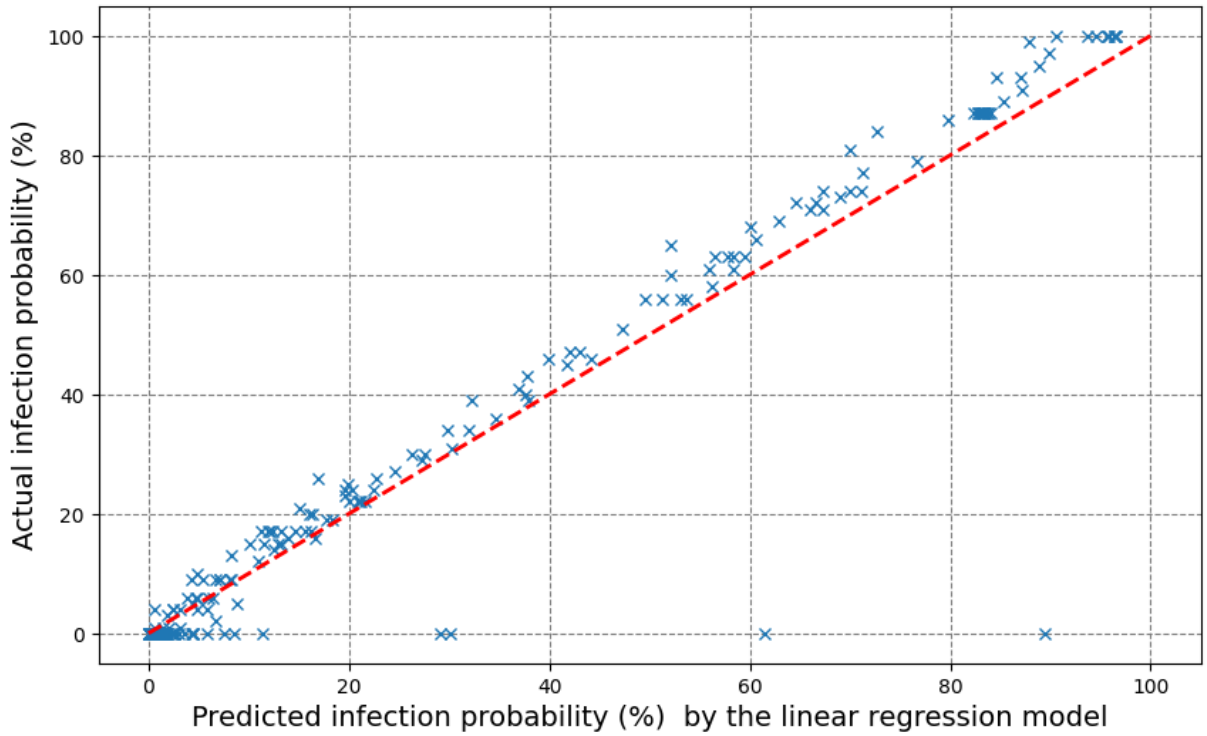

Рис. 3.1. Графік актуальних та спрогнозованих ймовірностей виникнення захворювання Fusarium Head Blight лінійною регресією для тестових даних

На рис. 3.2 нижче представлені графіки актуальних та спрогнозованих ймовірностей захворювання Fusarium Head Blight на тестових даних з плином часу. Головна відмінність від попереднього – в якості вісі x виступають погодинні позначки часу (в даному випадку 1299 годин, оскільки це розмірність

тестових даних). В якості вісі *y* – ймовірність захворювання в конкретний момент часу.

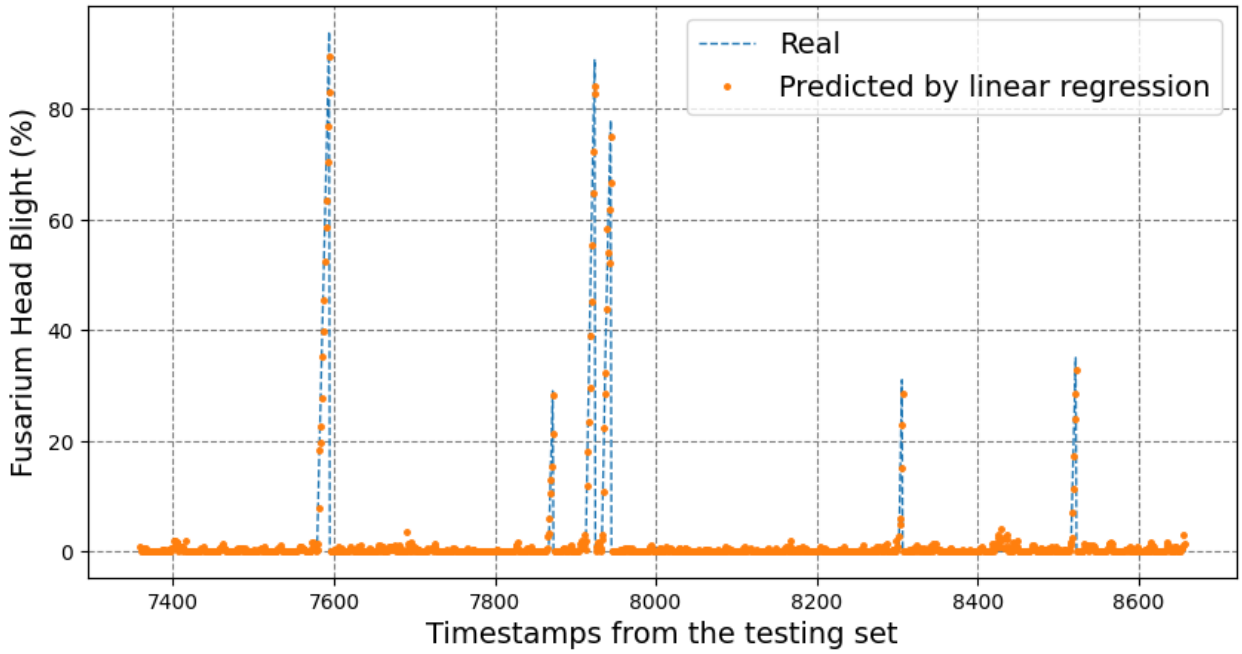

Рис. 3.2. Графік порівняння справжніх та спрогнозованих лінійною регресією ймовірностей захворювання Fusarium Head Blight протягом часу для тестових даних

3.3. Результати дослідження для Random Forest

Алгоритм Random Forest використовує результати з багатьох дерев прийняття рішень для запобіганню проблеми перенавчання. Розглянемо правила одного з дерев моделі Random Forest, створеної за допомогою бібліотеки sklearn (див. рис. 3.3).

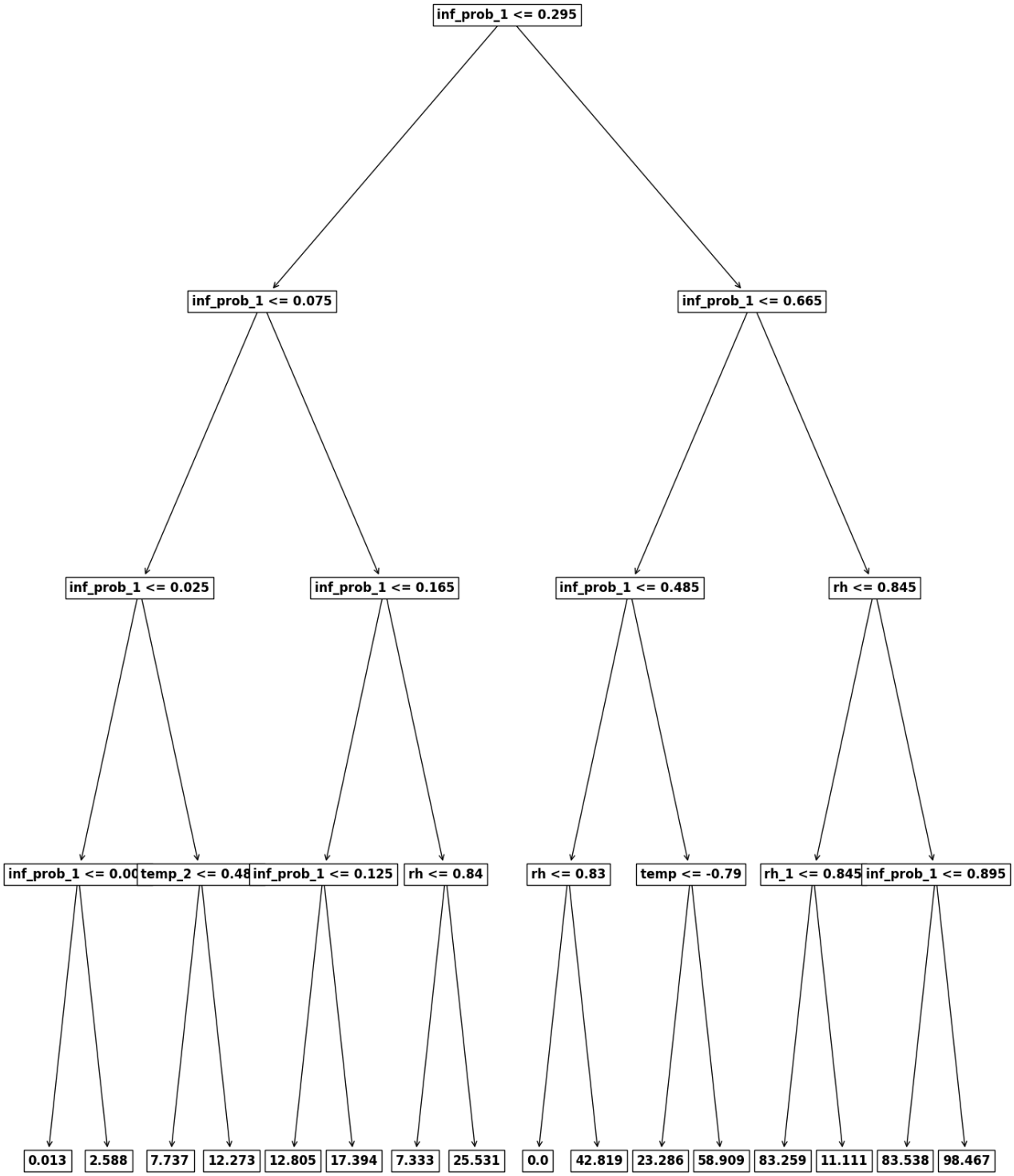

Рис. 3.3. Дерево прийняття рішень для визначення ймовірності виникнення захворювання

Як можна побачити зі створених правил, при прийнятті рішень в даному дереві переважали характеристики, такі як: inf_prob_1 (ймовірність захворювання за минулу годину), rh (поточна відносна вологість повітря), rh_1 (відносна вологість повітря годину тому) та $temp$ (поточна температура повітря). Це збігається з результатами, як були отримані лінійною регресією щодо значущості вхідних характеристик при прогнозування ймовірності виникнення захворювання.

Експериментальним шляхом встановлено, що найкращі результати для моделі «Випадковий ліс» отримано при значеннях гіперпараметрів $timestamps$ =3, $n\_estimators$ =10 (кількість дерев прийняття рішень) та max_depth=5 (максимальна глибина кожного дерева).

Результати, отримані для моделі «Випадковий ліс» на навчальних, валідаційних та тестових даних наведені в табл. 3.2.

Таблиця 3.2

Результати моделі «Випадковий ліс»

	Навчальні дані	Валідаційні дані	Тестові дані
R^2	0,984	0,941	0,965
$RMSE$	2,2	4,03	3,44

Точковий графік справжніх та спрогнозований моделлю «Випадковий ліс» ймовірностей виникнення захворювання Fusarium Head Blight наведено на рис. 3.4.

Для кращої наочності отриманих моделлю Random Forest результатів надано графік на рис. 3.5 з порівнянням справжніх та спрогнозованих моделлю ймовірностей захворювання Fusarium Head Blight з плином часу. Якщо порівнювати з графіком на рис. 3.2, де наведені результати лінійної регресії, то можна побачити, що модель Random Forest більш точно спрогнозувала нульові ймовірності захворювання.

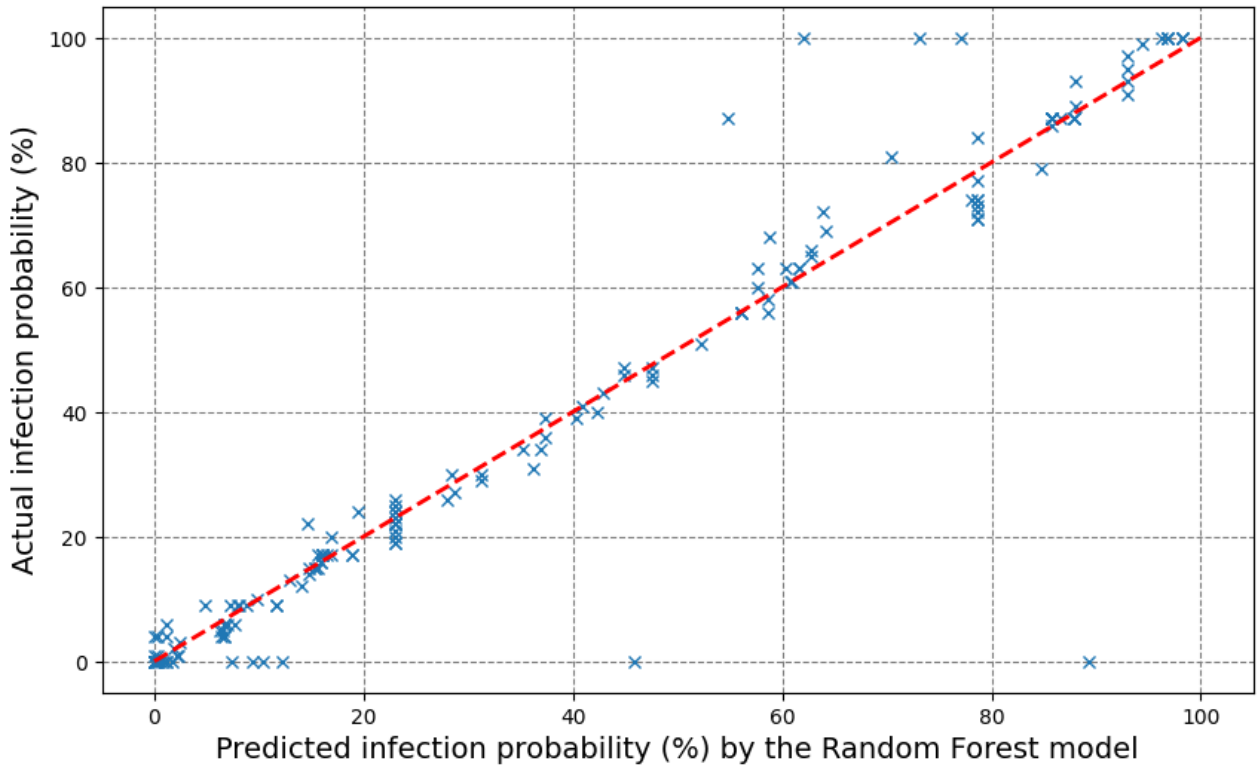

Рис. 3.4. Графік актуальних та спрогнозованих моделлю Random Forest ймовірностей появи захворювання для тестових даних

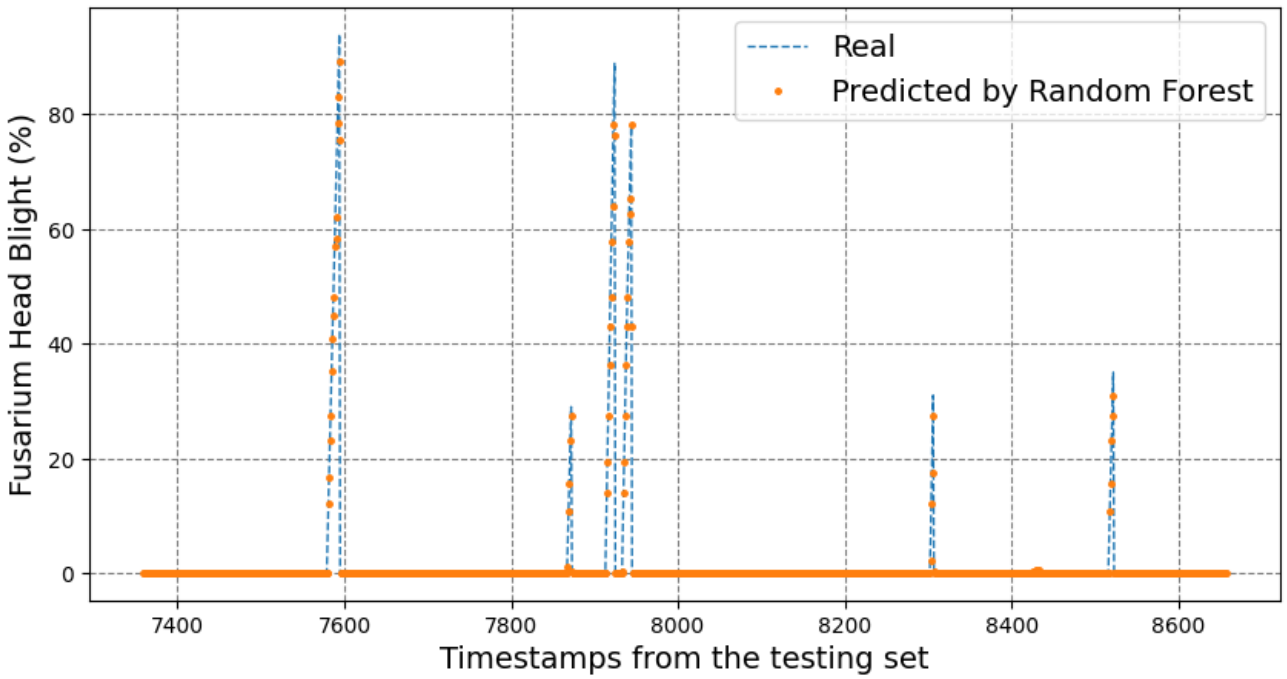

Рис. 3.5. Графік актуальних та спрогнозованих моделлю Random Forest ймовірностей появи захворювання з плином часу для тестових даних

3.4. Результати дослідження для нейронної мережі прямого поширення

В ході проведеного дослідження встановлено, що найкращі результати нейронною мережею прямого поширення вдалося досягти для наступних гіперпараметрів:

- $timestamps$=3;
- Функції активації: ReLU (або $g(x) = max\ (x, 0)$) у внутрішніх шарах, лінійна ($g(x) = x$) – на виході;
- Структура: 12, 6, 4, 4 нейронів у внутрішніх шарах та 1 нейрон на виході;
- Функція втрат: квадратична похибка (див. формулу 1.2);
- Оптимізація градієнтного спуску: AdamW (модифікація оптимізації Adam [16] з затуханням вагових коефіцієнтлщів для запобігання перенавчанню). Зміни вагових коефіцієнтів θ нейронної мережі за алгоритмом AdamW наведено у формулі 3.4 [17]:

$$
\begin{aligned}
m_{t+1} &\leftarrow \beta_1 m_t + (1-\beta_1) g_{t+1}, \\
v_{t+1} &\leftarrow \beta_2 v_t + (1-\beta_2) g_{t+1}^2, \\
\widehat{m}_{t+1} &\leftarrow \frac{m_{t+1}}{1-\beta_1^{t+1}}, \\
\hat{v}_{t+1} &\leftarrow \frac{v_{t+1}}{1-\beta_2^{t+1}}, \\
\theta_{t+1} &\leftarrow \theta_t - \eta\ \frac{\widehat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \varepsilon} - \eta\lambda\theta_t,
\end{aligned}
\tag{3.4}
$$

де m_{t+1}, v_{t+1} – EMA градієнтів та квадратів градієнтів відповідно на поточній ітерації навчання;

m_t, v_t – EMA градієнтів та квадратів градієнтів відповідно на попередній ітерації навчання (ініціалізуються нулями на початку навчання)

g_{t+1} – градієнт на поточній ітерації навчання;

β_1 , β_2 – коефіцієнти експоненційного затухання для градієнтів та квадратів градієнтів відповідно (стандартні значення β_1=0,9 та β_2=0,999);

$\hat{m}_{t+1}$ та $\hat{v}_{t+1}$ – ці значення потрібні для кращого наближення градієнта та квадрату градієнта на початкових ітераціях навчання;

θ_{t+1} і θ_t – вектори вагових коефіцієнтів на поточній та попередній ітерації навчання;

ε – число для запобігання ділення на нуль у знаменнику (стандартне значення 10^{-8});

η (або $learning\_rate$) – коефіцієнт навчання (стандартне значення 0,001);

λ (або $weight\_decay$) – коефіцієнт затухання ваг нейронної мережі.

Для даної оптимізації було емпіричним чином визначено значення коефіцієнта затухання ваг $weight\_decay$: 0,1.

- $batch\_size$=256 (контролює скільки екземплярів з навчальної вибірки використовується для однієї ітерації навчання);
- $epochs$=500 (кількість епох навчання).

Код для створення нейронної мережі з зазначеними гіперпараметрими показано нижче з використанням бібліотеки keras:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import AdamW
from keras.metrics import RootMeanSquaredError

fnn = Sequential([
    Dense(
      12,
      activation="relu",
      kernel_initializer="he_normal",
      input_dim=len(input_columns),
    ),
    Dense(6, activation="relu", kernel_initializer="he_normal"),
    Dense(4, activation="relu", kernel_initializer="he_normal"),
    Dense(4, activation="relu", kernel_initializer="he_normal"),
    Dense(1, activation="linear", kernel_initializer="he_normal"),
])
```

```
fnn.compile(
   optimizer=AdamW(weight_decay=0.1),
   loss="mse",
   metrics=[RootMeanSquaredError(name="rmse")],
)
fnn.build()
fnn.summary()
```

Графік зміни середньої квадратичної похибки в процесі навчання для навчальних та валідаційних даних наведений на рис. 3.6.

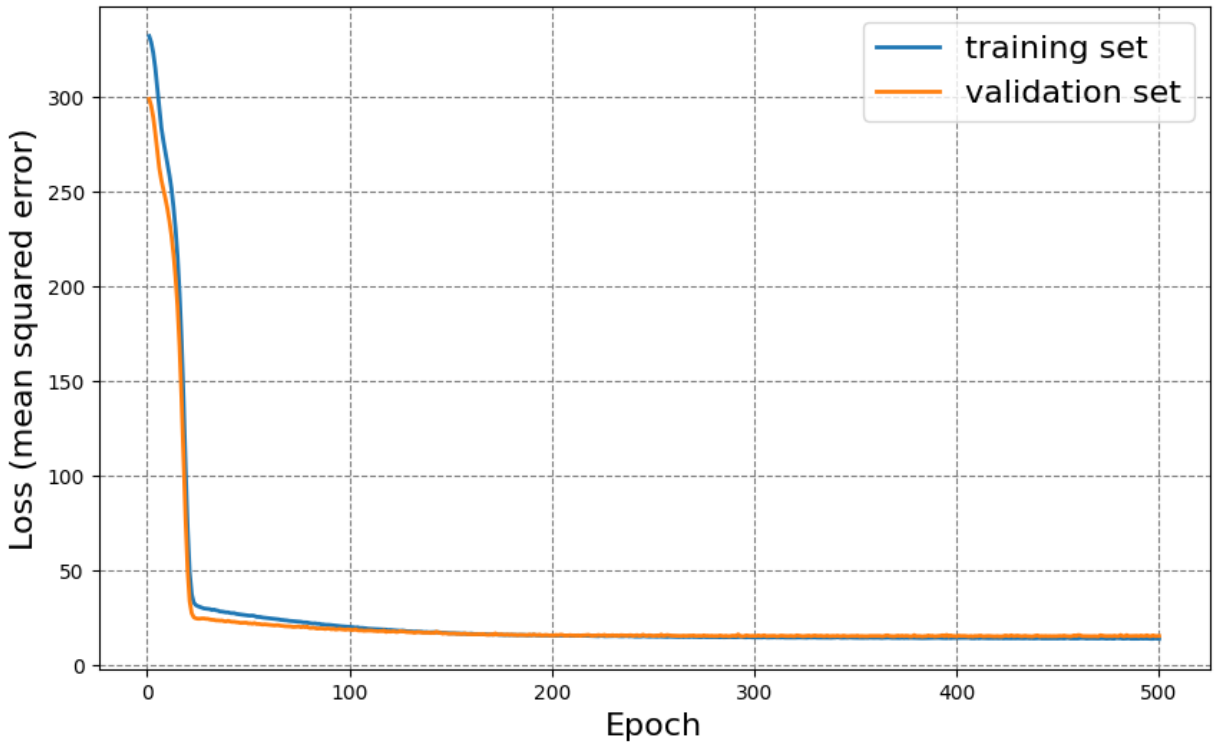

Рис. 3.6. Графік зміни похибки на навчальній та валідаційній вибірці в процесі тренування нейронної мережі прямого поширення

Значення регресійних метрик R^2 та $RMSE$, отриманих нейронною мережею з оптимальними гіперпараметрами на навчальних, валідаційних та тестових даних наведені в табл. 3.3.

Таблиця 3.3

Результати моделі «Нейронна мережа прямого поширення»

	Навчальні дані	Валідаційні дані	Тестові дані
R^2	0,95	0,945	0,963
$RMSE$	3,96	3,94	3,52

Точковий графік справжніх та спрогнозований нейронною мережею прямого поширення ймовірностей виникнення захворювання кукурудзи Fusarium Head Blight наведено на рис. 3.7.

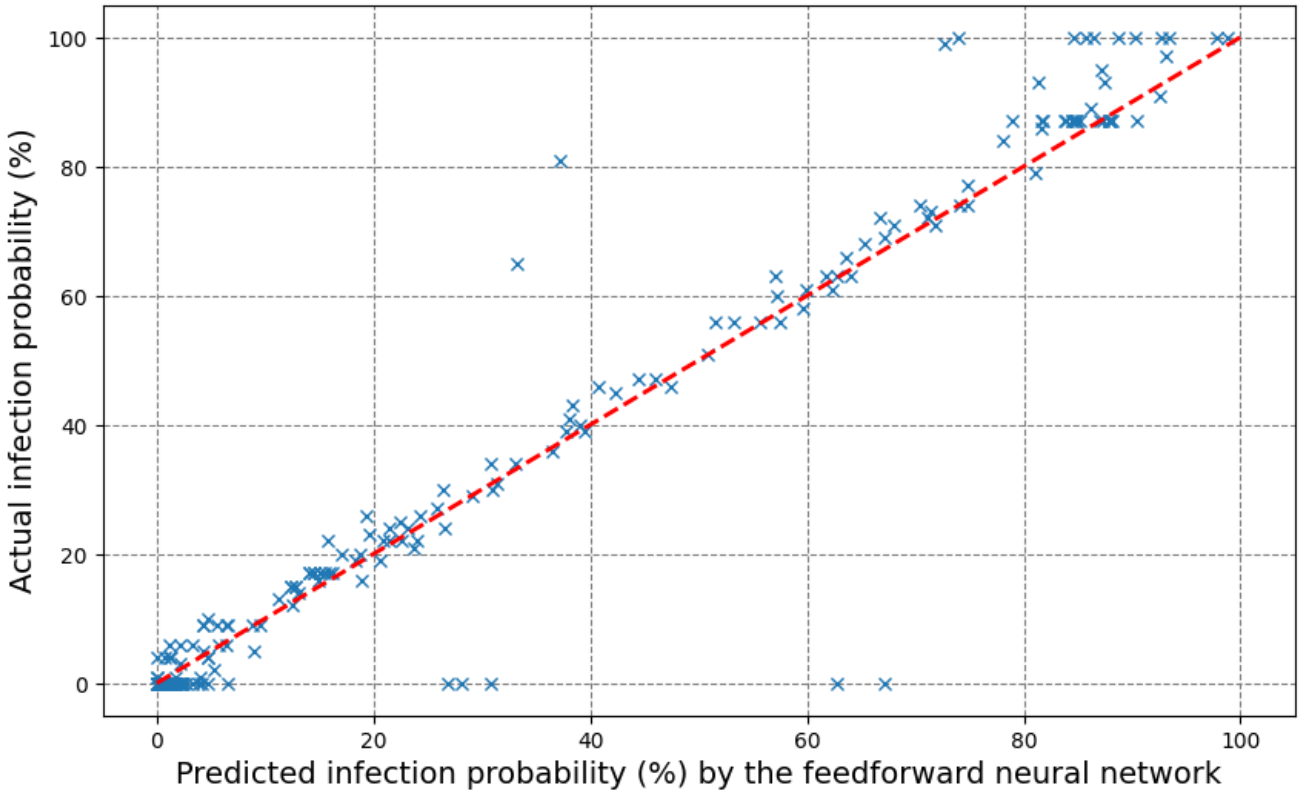

Рис. 3.7. Точковий графік актуальних та спрогнозованих ймовірностей виникнення захворювання Fusarium Head Blight нейронною мережею для тестових даних

На рис. 3.8 додатково наведено графік порівняння справжніх та отриманих нейронною мережею ймовірностей виникнення захворювання з плином часу на тестових даних.

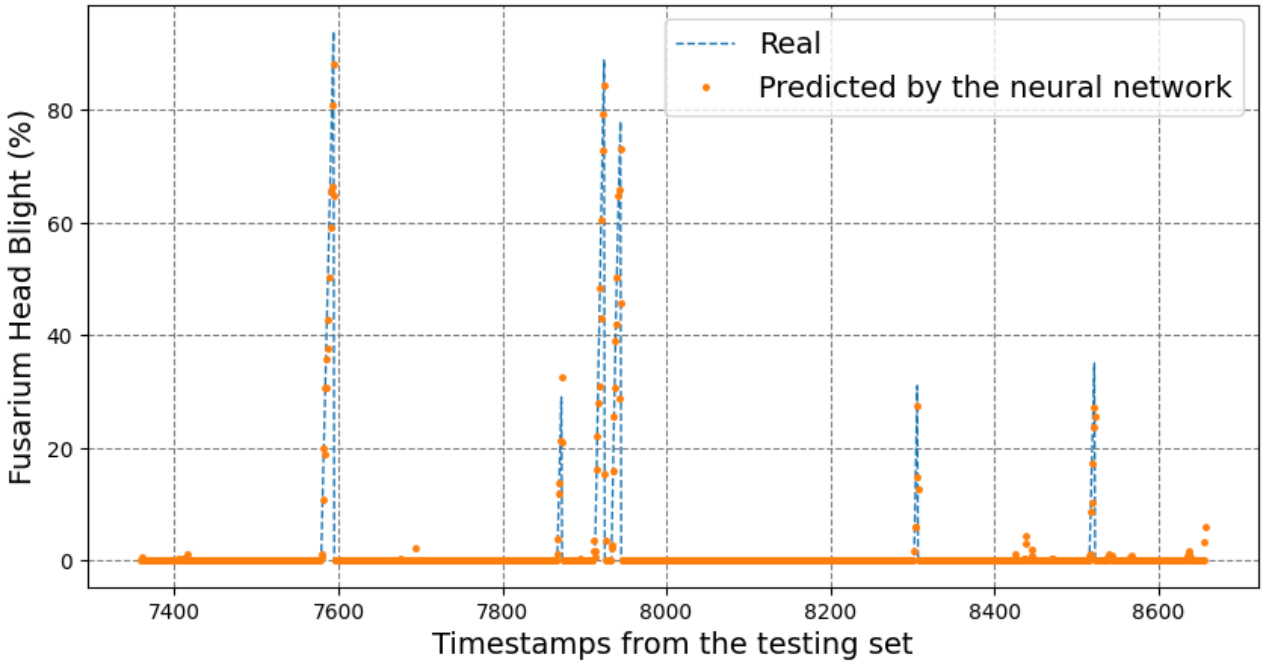

Рис. 3.8. Графік порівняння актуальних та спрогнозованих нейронною мережею прямого поширення ймовірностей захворювання Fusarium Head Blight

3.5. Порівняльний аналіз отриманих результатів

Таблиця 3.4 підсумовує набуті в процесі дослідження результати, а саме найкращі значення метрик R^2 та $RMSE$, яких вдалося досягти на тестових даних, а також відповідні гіперпараметри моделей ML при яких вдалося досягти цих значень.

Таблиця 3.4

Найкращі результати моделей ML на тестових даних та відповідні гіперпараметри

Модель	Гіперпараметри	Результати на тестових даних
Лінійна регресія	timestamps=3	R^2=0.96, $RMSE$=3.61

Продовження таблиці 3.4

Random Forest	timestamps=3, max_depth=5, n_estimators=10	R^2=0,965, $RMSE$=3,44
Нейронна мережа прямого поширення	timestamps=3, optimizer=AdamW(learning_rate=0.001, weight_decay=0.1, β_1=0.9, β_2=0.999), layers=[Dense(12, activation=ReLU), Dense(6, activation=ReLU), Dense(4, activation=ReLU), Dense(4, activation=ReLU), Dense(1, activation=Linear)], batch_size=256, epochs=500	R^2=0,962, $RMSE$=3,59

Продемонстровані в табл. 3.4 результати показують, що всі моделі показали найкращі результати при значенні гіперпараметра $timestamps$=3. Це означає що для найкращого прогнозу потрібно враховувати кліматичні показники разом з ймовірностями виникнення захворювання за останні 3 години разом з поточними кліматичними показниками. Найкращі значення згідно регресійних метрик на тестових даних показала модель Random Forest: R^2 =0.965, $RMSE$=3.44.

На рис. 3.9 та 3.10 наведені стовпчикові діаграми зі значеннями метрик R^2 та $RMSE$ на тестових даних для лінійної регресії, Random Forest та нейронної мережі прямого поширення.

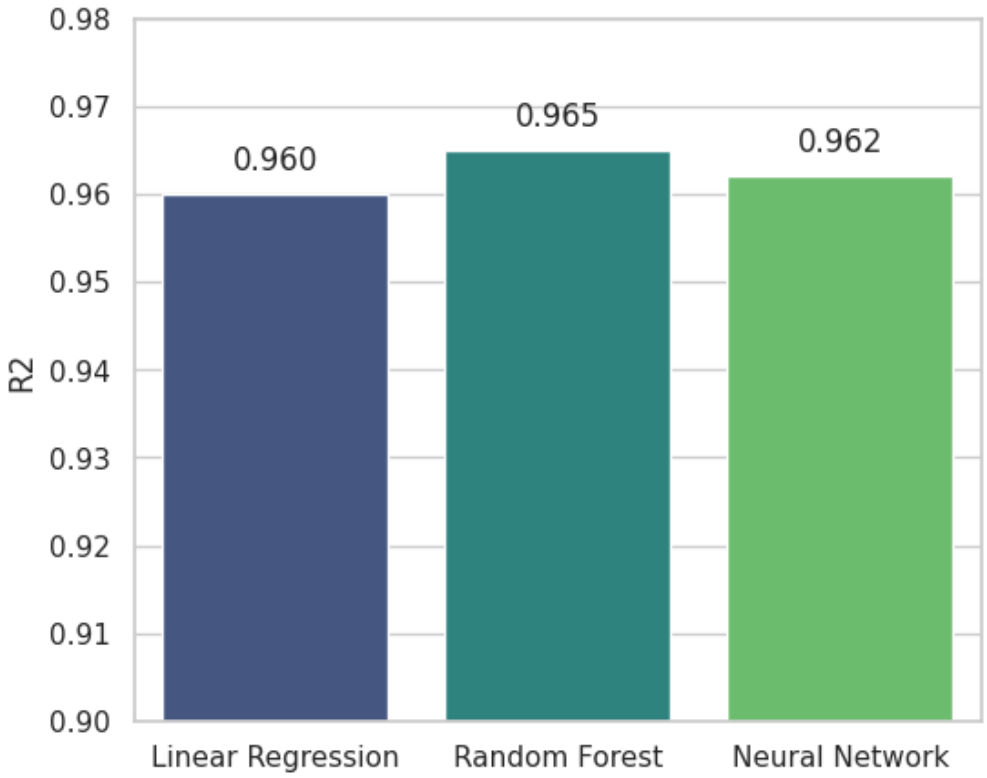

Рис. 3.9. Значення метрики R^2 на тестових даних для розглянутих моделей ML

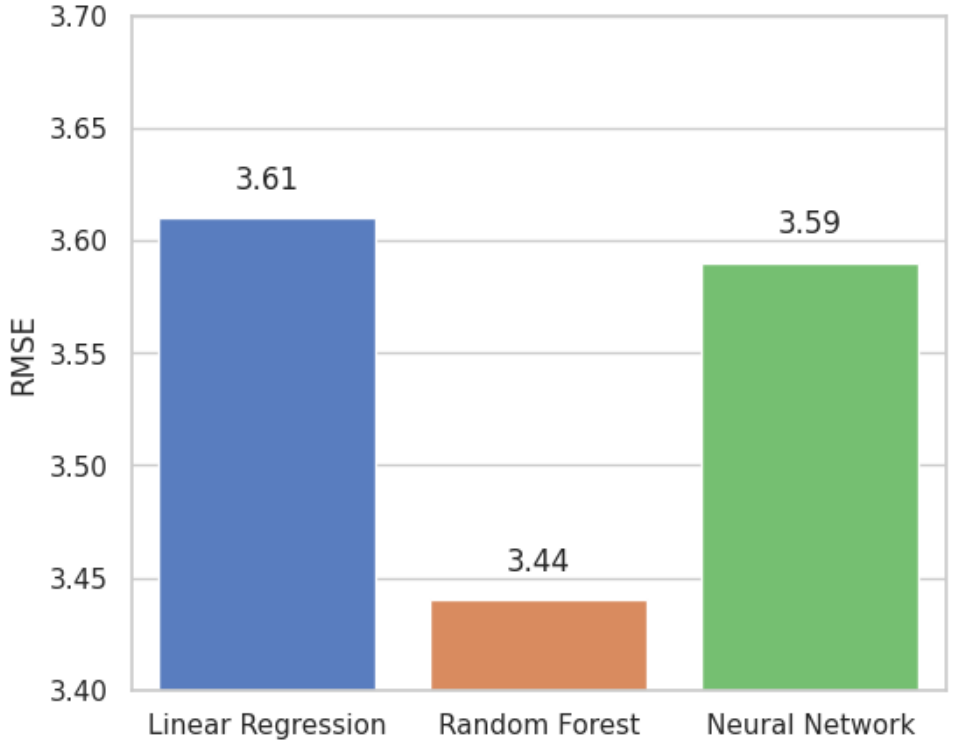

Рис. 3.10. Значення метрики $RMSE$ на тестових даних для розглянутих моделей ML

3.6. Перспективи подальшого вдосконалення

На рис. 3.11 візуалізовано оптимальний підхід, що включає в себе історію кліматичних показників та ймовірностей захворювання за останні 3 години разом з поточними кліматичними показниками для прогнозування ймовірності виникнення захворювання Fusarium Head Blight у поточний момент часу t. Зворотні зв'язки на діаграмі означають, що модель використовує власні спрогнозовані значення в якості вхідних значень для ймовірностей захворювання за 1-ну, 2-гу та 3-тю годину тому.

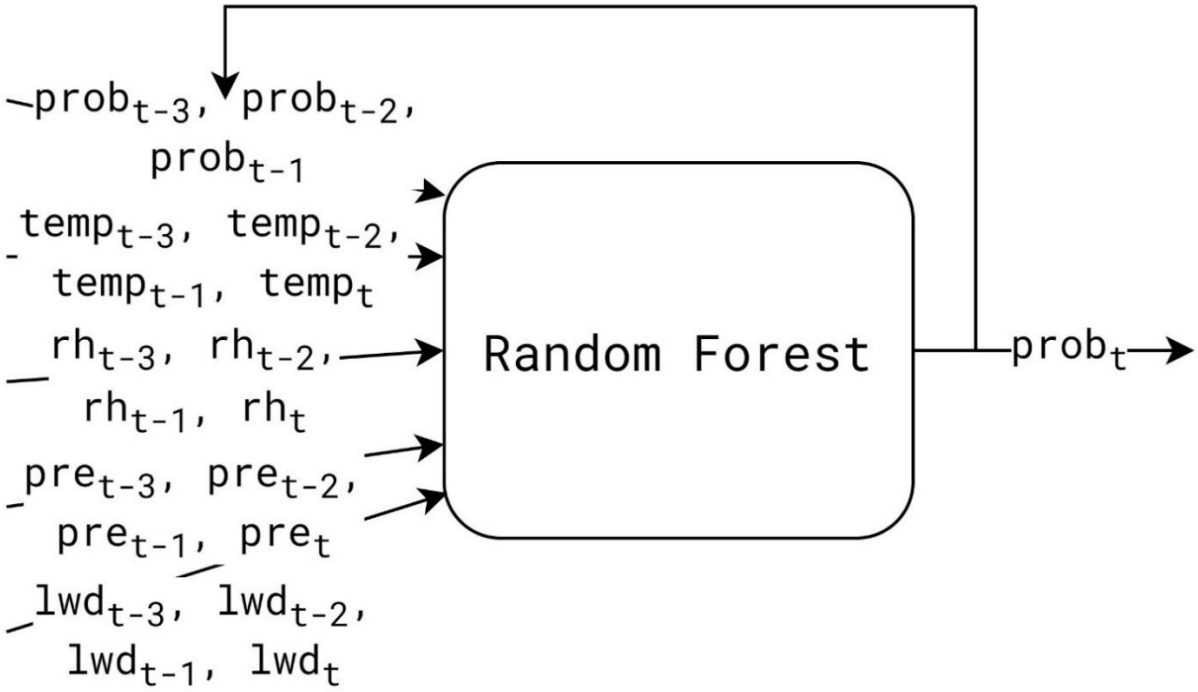

Рис. 3.11. Модель Random Forest для прогнозування Fusarium Head Blight

Перспективами для подальшого вдосконалення є:

1. Проведення додаткових досліджень з метою розширення набору даних і включення ширшого спектру агрокультур та типів діагностованих хвороб.

2. Як показали значення коефіцієнтів лінійної регресії у формулі 3.3, деякі вхідні дані мали значно більший плив на визначення ймовірності захворювання за інші. Таким чином постає питання вибору найбільш важливих кліматичних

показників за минулі проміжку часу замість використання історії змін всіх кліматичних даних за останні t годин.

3. Створення веб-застосунку для прогнозування ймовірності виникнення захворювання с/г культур на базі розробленого оптимального підходу.

4. Застосування архітектур рекурентних нейронних мереж (RNN), зокрема моделі LSTM, що в роботі [11] показала найкращі результати, для задачі прогнозування ймовірності виникнення захворювання в с/г на базі кліматичних даних.

Крім того, до пріоритетних напрямків відносяться [18] ті, що пов'язані з вивченням впливу вхідних фізико-хімічних величин на достовірність виявлення хвороб рослин кількісному рівні. За допомогою графіків часткових залежностей можна показати вплив окремих вхідних сигналів на прогнозування вплив окремих вхідних сигналів на прогноз при незмінності інших вхідних параметрів.

Аналіз чутливості може бути використаний для оцінки стійкості моделі до змін вхідних сигналів. Він може включати збурення вхідних значень у певному діапазоні та спостереження за реакцією моделі, щоб проаналізувати, наскільки чутливою або нечутливою є модель до кожного вхідного сигналу.

Дослідження у вищезазначених сферах покращать якість, адаптивність та масштаби діагностики хвороб сільськогосподарських культур, що матиме позитивний вплив на інвестиційну привабливість та довгострокову стійкість сільськогосподарського сектору завдяки збільшенню інноваційної складової сільськогосподарського сектору.

3.7. Висновки

1. Результати дослідження ефективності ML алгоритмів під час прогнозування ймовірності виникнення хвороби Fusarium Head Blight кукурудзи показали, що для найбільш точного прогнозу необхідно враховувати історію змін кліматичних показників та ймовірностей виникнення захворювання за останні 3

години про що свідчить значення гіперпараметра $timestamps$=3 в найбільш оптимальних налаштуваннях кожної розглянутої моделі.

2. Регресійна модель Random Forest показала трохи кращі результати ніж нейронна мережа та лінійна регресія на тестових даних: R^2=0,965, $RMSE$=3,44

3. Найбільш впливовими чинниками при розрахунку ймовірності виникнення захворювання Fusarium Head Blight для лінійної регресії були: ймовірність захворювання за попередню годину, поточна вологість повітря та вологість повітря за останні 3 години, а також поточна температура повітря про що свідчать значення коефіцієнтів 98,02, 20,73, -19,52, 5,57, -4,99 та 3,15 відповідно. Визначення найбільш впливових на виникнення захворювання вхідних параметрів та створення моделі, яка вибірково використовує лише окремі кліматичні параметри є питанням, що потребує додаткового розвитку та досліджень.

4. Перспективами подальшого розвитку та вдосконалення набутих результатів:

4.1. Розширення обсягів досліджень, з метою збагачення набору даних і включення різноманітних агрокультур та типів виявлених хвороб. Це дозволить отримати більш повний та репрезентативний набір інформації.

4.2. Розробка веб-застосунку для передбачення ймовірності виникнення захворювань сільськогосподарських культур на основі розробленого оптимального підходу.

4.3. Використання вибіркових значень кліматичних показників за минулі періоди часу замість врахування історії змін у всіх кліматичних даних за останні t годин. Аналіз коефіцієнтів лінійної регресії у формулі 3.3 показав, що деякі вхідні дані мають значно більший вплив на визначення ймовірності захворювання ніж інші.

4.4. Впровадження архітектур рекурентних нейронних мереж (RNN), зокрема моделі LSTM, яка, як вказано в роботі [11], продемонструвала найкращі результати, для задачі прогнозування ймовірності виникнення захворювань в сільському господарстві на основі кліматичних даних.

ВИСНОВКИ

1. Проведено аналіз та логічне узагальнення архітектурних складових відомих систем та підходів прогнозування ймовірностей виникнення захворювання с/г культур на базі кліматичних даних.

2. Обґрунтовано методи досліджень щодо оцінки ефективності ML алгоритмів для прогнозування ймовірності виникнення захворювання с/г культур.

3. Моделей ML Random Forest з гіперпараметрами $timestamps$ =3, max_depth =5 та $n\_estimators$ =10 показала найкращі значення регресійних метрик на тестових даних при прогнозуванні виникнення захворювання кукурудзи Fusarium Head Blight: R^2=0.965, $RMSE$=3.44. Трохи гірші результати отримано нейронною мережею прямого поширення (R^2=0.962, $RMSE$=3.59) та лінійною регресією (R^2=0.96, $RMSE$=3.61).

4. В ході проведеного дослідження прогнозування ймовірності виникнення захворювання Fusarium Head Blight встановлено, що для найбільш точного прогнозу необхідно разом з поточними кліматичними показниками враховувати кліматичні показники та ймовірності виникнення захворювання за останні 3 години.

5. Для лінійної регресії найбільш впливовими чинниками при розрахунку ймовірності виникнення захворювання Fusarium Head Blight були: ймовірність захворювання годину тому, поточна відносна вологість повітря та відносні вологості за останні 3 години, а також поточна температура повітря. Про це свідчать найбільші за модулем відповідні значення коефіцієнтів моделі (для зазначених чинників ці значення в становили 98,02, 20,73, -19,52, 5,57, -4,99 та 3.15 відповідно. Визначення найбільш впливових на виникнення захворювання вхідних параметрів, враховуючи історію змін кліматичних даних та ймовірностей виникнення захворювання є питанням, що потребує додаткового розвитку.

6. В перспективах подальшого вдосконалення:

6.1. Проведення додаткових досліджень з метою розширення набору даних і включення ширшого спектру агрокультур та типів діагностованих хвороб.

6.2. Створення веб-застосунку для прогнозування ймовірності виникнення захворювання с/г культур на базі розробленого оптимального підходу.

6.3. Використання архітектур рекурентних нейронних мереж (RNN), зокрема моделі LSTM, для задачі прогнозування ймовірності виникнення захворювання с/г культур в залежності від кліматичних чинників.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agricultural production statistics 2000 – 2021. URL: https://www.fao.org/3/cc3751en/cc3751en.pdf (дата звернення 30.11.2023).

2. FAOSTAT: Crop and livestock products. URL: https://www.fao.org/faostat/en/#data/QCL (дата звернення 30.11.2023).

3. Fenu G., Malloci F. M. An application of machine learning technique in forecasting crop disease. ACM International Conference Proceeding Series. 2019. No. June C. 76-82. DOI: 10.1145/3372454.3372474.

4. Fenu, G., Malloci, F. M. Review forecasting plant and crop disease: An explorative study on current algorithms. Big Data and Cognitive Computing. 2021. Vol. 5, No. 1. C. 1–24. DOI: 10.3390/bdcc5010002.

5. Patil, R. R., Kumar, S., Rani, R. Comparison of Artificial Intelligence Algorithms in Plant Disease Prediction. Revue d'Intelligence Artificielle. 2022. Vol. 36, No. 2. C. 185–193. DOI: 10.18280/ria.360202.

6. Madasamy, B., Balasubramaniam, P., Dutta, R. Microclimate-based pest and disease management through a forewarning system for sustainable cotton production. Agriculture (Switzerland). 2020. Vol. 10, No. 12. C. 1–12. DOI: 10.3390/agriculture10120641.

7. Nettleton, D. F., Katsantonis, D., Kalaitzidis, A., et al. Predicting rice blast disease: Machine learning versus process-based models. BMC Bioinformatics. 2019. Vol. 20, No. 1. DOI: 10.1186/s12859-019-3065-1.

8. Deep learning cheatsheet. URL: https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-deep-learning (дата звернення 04.12.2023).

9. Segovia, J. A., Toaquiza, J. F., Llanos, J. R., et al. Meteorological Variables Forecasting System Using Machine Learning and Open-Source Software. Electronics (Switzerland). 2023. Vol. 12, No. 4. DOI: 10.3390/electronics12041007.

10. Segovia, J. A., Toaquiza, J. F., Llanos, J. R., et al. Meteorological Variables Forecasting System Using Machine Learning and Open-Source Software. Electronics (Switzerland). 2023. Vol. 12, No. 4. DOI: 10.3390/electronics12041007.

11. Xiao, Q., Li, W., Kai, Y., et al. Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. BMC Bioinformatics. 2019. Vol. 20, No. Suppl 25. C. 1–15. DOI: 10.1186/s12859-019-3262-y.

12. Vasisht, D., Kapetanovic, Z., Won, J. ho, et al. Farmbeats: An IoT platform for data-driven agriculture. Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017. 2017. C. 515–529.

13. IBM Watson Decision Platform for Agriculture. URL: https://worldagritechusa.com/wp-content/uploads/2019/03/Dan-Wolfson-IBM.pdf (дата звернення 16.12.2023)

14. CropX System Disease Control. URL: https://cropx.com/cropx-system/disease-control/ (дата звернення 15.12.2023).

15. Wan X. Infuence of feature scaling on convergence of gradient iterative algorithm. Journal of Physics: Conference Series. 2019. C. 1-6, DOI: 10.1088/1742-6596/1213/3/032021.

16. Kingma, D. P., Ba, J. L. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. 2015. C. 1–15. URL: https://arxiv.org/pdf/1412.6980.pdf (дата звернення 16.12.2023)

17. Loshchilov, I., Hutter, F. Decoupled weight decay regularization. 7th International Conference on Learning Representations, ICLR 2019. 2019. URL: https://arxiv.org/pdf/1711.05101.pdf (дата звернення 16.12.2023).

18. Laktionov, I., Diachenko, G., Rutkowska, D., et al. an Explainable Ai Approach To Agrotechnical Monitoring and Crop Diseases Prediction in Dnipro Region of Ukraine. Journal of Artificial Intelligence and Soft Computing Research. 2023. Vol. 13, No. 4. C. 247–272. DOI: 10.2478/jaiscr-2023-0018.

ДОДАТОК А. ПРОГРАМНИЙ КОД

А.1. Імпорт бібліотек

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.models import Sequential, load_model
from keras.layers import Dense
from keras.optimizers import AdamW
from keras.callbacks import ModelCheckpoint
from keras.metrics import RootMeanSquaredError
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_squared_error
from fast_ml.model_development import train_valid_test_split
from tqdm import tqdm
```

А.2. Читання даних з файлу CSV

```
df = pd.read_csv(
    "/content/drive/MyDrive/Master's Thesis/fusarium_head_blight.csv"
)
df.drop(columns=["datetime"], inplace=True)
df.tail(5)
```

А.3. Створення матриці розсіювання

```
pd.plotting.scatter_matrix(
        df, figsize=(10, 10), alpha=1, marker="."
)
```

А.4. Обчислення статистичних показників

```
df.describe()
```

А.5. Створення графіку зміни кліматичних даних з часом

```
fig, axs = plt.subplots(2, 2, figsize=(10, 6))
```

```
fig.suptitle("Метеостанція: METOS by Pessl Instruments. 8658 записів за кожну годину",
fontsize=20)

axs[0, 0].plot(df.index, df.air_temperature)
axs[0, 0].set_title("Air Temperature (°C)")
axs[0, 0].set_xlabel("Timestamp")
axs[0, 0].set_ylabel("Air Temperature (°C)")
axs[0, 0].grid()
axs[0, 1].plot(df.index, df.humidity)
axs[0, 1].set_title("Relative Humidity (%)")
axs[0, 1].set_xlabel("Timestamp")
axs[0, 1].set_ylabel("Relative Humidity (%)")
axs[0, 1].grid()
axs[1, 0].plot(df.index, df.precipitation)
axs[1, 0].set_title("Precipitation (mm)")
axs[1, 0].set_xlabel("Timestamp")
axs[1, 0].set_ylabel("Precipitation (mm)")
axs[1, 0].grid()
axs[1, 1].plot(df.index, df.leaf_wetness_duration, ".")
axs[1, 1].set_title("Leaf Wetness Duration (min)")
axs[1, 1].set_xlabel("Timestamp")
axs[1, 1].set_ylabel("Leaf Wetness Duration (min)")
axs[1, 1].grid()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

А.6. Функція для знаходження k найдовших послідовностей для даних, де ймовірність появи захворювання була додатною

```
def get_top_k_largest_sequences(
        df: pd.DataFrame, k: int = 5
) -> list[list[pd.Series]]:
  if len(df) == 0:
    return []

  current_sequence = [df.iloc[0]]
  sequences = [current_sequence]
  for i in range(1, len(df)):
    row = df.iloc[i]

    current_index = df.index[i]
    prev_index = df.index[i - 1]
    index_difference = current_index - prev_index

    if index_difference == 1:
      current_sequence.append(row)
      continue
```

```
            current_sequence = [row]
            sequences.append(current_sequence)

    return sorted(sequences, key=len, reverse=True)[:k]
```

A.7. Додавання попередніх кліматичних показників та ймовірностей виникнення захворювання за останні 3 години

```
def add_previous_timestamps_values(
            df: pd.DataFrame, timestamps: int = 2
) -> pd.DataFrame:
    df_with_timestamps = pd.DataFrame()
    columns = df.columns.tolist()

    for index, row in tqdm(
                        df.iterrows(),
                        total=len(df),
                        desc="Processing Rows",
                        ncols=100
    ):
        if index < timestamps:
            continue

        for column in columns:
            df_with_timestamps.at[index, column] = df.at[
                                index, column
                        ]

            for t in range(1, timestamps + 1):
                df_with_timestamps.at[index, f"{column}_{t}"]=df.at[
                                        index - t, column
                                ]

    return df_with_timestamps.drop(timestamps)

df_with_timestamps = add_previous_timestamps_values(
            df, timestamps=3
)
```

A.8. Попередня обробка навчальних, валідаційних та тестових даних

```
target_column = "infection_prob"
standard_scale_columns = [
```

```
        c
        for c in df_with_timestamps.columns
        if c.startswith(("air_temperature", "precipitation"))
]
input_columns = [
        c for c in df_with_timestamps.columns if c != "infection_prob"
]

X_train, y_train, X_val, y_val, X_test, y_test = \
        train_valid_test_split(
                df_with_timestamps,
                target_column,
                train_size=0.7,
                valid_size=0.15,
                test_size=0.15,
                random_state=1
)

humidity_columns = [
  c for c in df_with_timestamps.columns if c.startswith("humidity")
]
X_train[humidity_columns] /= 100
X_val[humidity_columns] /= 100
X_test[humidity_columns] /= 100

leaf_wetness_duration_columns = [c for c in df_with_timestamps.columns if
c.startswith("leaf_wetness_duration")]
X_train[leaf_wetness_duration_columns] /= 60
X_val[leaf_wetness_duration_columns] /= 60
X_test[leaf_wetness_duration_columns] /= 60

infection_prob_columns = [
        c for c in X_train.columns if c.startswith("infection_prob")
]
X_train[infection_prob_columns] /= 100
X_val[infection_prob_columns] /= 100
X_test[infection_prob_columns] /= 100

air_temperature_columns = [
        c
        for c in df_with_timestamps.columns
        if c.startswith("air_temperature")
]
mean_air_temperature = X_train.air_temperature.mean()
std_air_temperature = X_train.air_temperature.std()

X_train[air_temperature_columns] = (
   X_train[air_temperature_columns] - mean_air_temperature
) / std_air_temperature
```

```
X_val[air_temperature_columns] = (
   X_val[air_temperature_columns] - mean_air_temperature
) / std_air_temperature
X_test[air_temperature_columns] = (
   X_test[air_temperature_columns] - mean_air_temperature
) / std_air_temperature

precipitation_columns = [
          c
          for c in df_with_timestamps.columns
          if c.startswith("precipitation")
]

mean_precipitation = X_train.precipitation.mean()
std_precipitation = X_train.precipitation.std()

X_train[precipitation_columns] = (
   X_train[precipitation_columns] - mean_precipitation
) / std_precipitation
X_val[precipitation_columns] = (
   X_val[precipitation_columns] - mean_precipitation
) / std_precipitation
X_test[precipitation_columns] = (
   X_test[precipitation_columns] - mean_precipitation
) / std_precipitation
```

А.9. Навчання та друк коефіцієнтів лінійної регресії

```
linear_regression = LinearRegression()
linear_regression.fit(X_train, y_train)

print("Linear regression equation: \n")
for i, coef in enumerate(linear_regression.coef_):
  print(f"{round(coef, 2)}*{input_columns[i]} +")
print(round(linear_regression.intercept_, 2))
```

А.10. Створення нейронної мережі

```
fnn = Sequential([
   Dense(
            12,
            activation="relu",
            kernel_initializer="he_normal",
            input_dim=len(input_columns)
```

```
    ),
    Dense(6, activation="relu", kernel_initializer="he_normal"),
    Dense(4, activation="relu", kernel_initializer="he_normal"),
    Dense(4, activation="relu", kernel_initializer="he_normal"),
    Dense(1, activation="linear", kernel_initializer="he_normal")
])

fnn.compile(optimizer=AdamW(weight_decay=0.1), loss="mse",
metrics=[RootMeanSquaredError(name="rmse")])
fnn.build()

fnn.summary()
```

А.11. Навчання нейронної мережі

```
model_checkpoint = ModelCheckpoint(
        "fnn.h5",
        monitor="val_loss",
        mode="min",
        save_best_only=True
)

history = fnn.fit(
    X_train,
    y_train,
    epochs=500,
    batch_size=256,
    validation_data=(X_val, y_val),
    callbacks=[model_checkpoint]
)
```

А.12. Побудова графіку прогресу навчання на тренувальних та валідаційних даних

```
train_loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(train_loss) + 1)
plt.figure(figsize=(10,6))
plt.plot(epochs, train_loss, linewidth=2)
plt.plot(epochs, val_loss, linewidth=2)
plt.title("Training progress", fontsize=18)
plt.xlabel("Epoch", fontsize=16)
plt.ylabel("Loss (mean squared error)", fontsize=16)
plt.legend(["training set", "validation set"], fontsize=16)
plt.grid(color="grey", linestyle="dashed")
```

А.13. Навчання моделі Random Forest:

```
random_forest = RandomForestRegressor(
        random_state=1, n_estimators=10, max_depth=4
)
random_forest.fit(X_train, y_train)
```

А.14. Оцінка ефективності обраної моделі ML на навчальних та валідаційних даних

```
y_train_pred = [
        max(0, prob) for prob in model_to_evaluate.predict(X_train)
]
y_val_pred = [
        max(0, prob) for prob in model_to_evaluate.predict(X_val)
]

print(
  f"TRAINING SET RESULTS: R2={r2_score(y_train, y_train_pred)},
RMSE={mean_squared_error(y_train, y_train_pred)**0.5}"
)
print(
  f"VALIDATION SET RESULTS: R2={r2_score(y_val, y_val_pred)},
RMSE={mean_squared_error(y_val, y_val_pred)**0.5}"
)
```

А.15. Перевірка роботи моделі на тестових даних

```
y_test_pred = [
        min(max(0, prob), 100)
        for prob in model_to_evaluate.predict(X_test)
]
print(f"TESTING SET RESULTS: R2={r2_score(y_test, y_test_pred)},
RMSE={mean_squared_error(y_test, y_test_pred)**0.5}\n")
plt.figure(figsize=(10, 6))
plt.plot(y_test_pred, y_test, "x")
plt.plot([0, 100], [0, 100], "r--", linewidth=2)
plt.title(
  "Predicted vs Actual Infection Probability (%) on the test set",
  fontsize=16
```

```
)
plt.xlabel(
"Predicted infection probability (%) by the feedforward neural network",
  fontsize=14
)
plt.ylabel("Actual infection probability (%)", fontsize=14)
plt.grid(color="grey", linestyle="dashed")
```

ДОДАТОК Б. ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім’я файла	**Опис**
Пояснювальні документи	
ВізнюкАВ_121м-22-3_ПЗ.docx	Пояснювальна записка роботи. Документ Word.
ВізнюкАВ_121м-22-3_ПЗ.pdf	Пояснювальна записка роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми
Презентація	
ВізнюкАВ_121м-22-3_ДМ.pptx	Презентація роботи