

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЦЬОМИК Олег Ігорович

Інтелектуальний модуль моніторингу стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі / Intelligent Module for Monitoring the Condition of Agricultural Crops in a Greenhouse Based on a Deep Neural Network

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
О. І. Цьомик

Науковий керівник:
к.т.н., доцент, І. М. Майків

Кваліфікаційну роботу
допущено до захисту:
"___" _____ 20__ р.

Завідувач кафедри
_____ **Н.В. Дзюбановська**

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Н.В. Дзюбановська

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЦЬОМИКУ Олегу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Інтелектуальний модуль моніторингу стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі / Intelligent Module for Monitoring the Condition of Agricultural Crops in a Greenhouse Based on a Deep Neural Network

керівник роботи к.т.н., доцент І.М. Майків

затверджені наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- проаналізувати предметну область моніторингу стану сільськогосподарських культур у теплиці;
- дослідити існуючі підходи до використання IoT, машинного та глибокого навчання в аграрних системах;
- визначити інформативні параметри тепличного середовища та структуру вхідних даних для моделювання;
- розробити архітектуру інтелектуального модуля моніторингу;
- побудувати модель глибокої нейронної мережі для аналізу сенсорних даних;
- реалізувати програмний модуль і провести експериментальне дослідження його роботи;
- оцінити результати функціонування розробленого рішення за показниками якості.

5. Перелік графічного матеріалу в роботі

- узагальнена схема архітектури інтелектуального модуля;
- структурна схема програмно-аналітичного модуля;
- схема формування та попередньої обробки даних IoT-сенсорів;
- структура моделі глибокої нейронної мережі для класифікації стану

культур;

- схема обробки вхідних даних у DNN-моделі;
- схема алгоритму функціонування інтелектуального модуля моніторингу;
- схема алгоритму навчання DNN-моделі для модуля моніторингу.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студент _____ О.І. Цьомик
підпис

Керівник роботи _____ к.т.н., доцент І.М. Майків
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інтелектуальний модуль моніторингу стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 83 сторінки і містить 15 ілюстрацій, 16 таблиць, 3 додатки та 35 використаних джерел.

Метою кваліфікаційної роботи є розроблення інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі для аналізу даних IoT-сенсорів та визначення станів тепличного середовища.

Методи досліджень включають метод аналізу для визначення особливостей моніторингу стану сільськогосподарських культур у теплиці, IoT-сенсорів і сучасних підходів до інтелектуального агромоніторингу; метод моделювання; методи попередньої обробки даних; методи глибокого навчання; метод експериментальних досліджень; метод порівняльного аналізу результатів.

Розроблено інтелектуальний модуль моніторингу стану сільськогосподарських культур у теплиці, який забезпечує завантаження даних IoT-сенсорів, перевірку їх структури, попередню обробку, нормалізацію ознак, навчання глибокої нейронної мережі та класифікацію стану тепличного середовища. Реалізовано програмні компоненти для навчання моделі, прогнозування стану, оцінювання якості класифікації та візуалізації результатів.

Результати дослідження можуть бути використані для побудови програмних модулів моніторингу тепличного середовища, інтеграції IoT-сенсорів із системами підтримки рішень, автоматизованого контролю параметрів вирощування, а також для створення навчально-дослідних прототипів у сфері розумного сільського господарства.

Ключові слова: ТЕПЛИЦЯ, СІЛЬСЬКОГОСПОДАРСЬКІ КУЛЬТУРИ, МОНІТОРИНГ СТАНУ, ІНТЕРНЕТ РЕЧЕЙ, IoT-СЕНСори, ГЛИБОКА НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ.

ANNOTATION

Qualification work on the topic «Intelligent Module for Monitoring the Condition of Agricultural Crops in a Greenhouse Based on a Deep Neural Network» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 83 pages and it contains 15 figures, 16 tables, 3 annexes and 35 sources.

The purpose of the qualification work is to develop an intelligent module for monitoring the condition of agricultural crops in a greenhouse based on a deep neural network for analyzing IoT sensor data and determining the states of the greenhouse environment.

The research methods include the method of analysis to determine the features of monitoring the condition of agricultural crops in a greenhouse, IoT sensors, and modern approaches to intelligent agricultural monitoring; the modeling method; data preprocessing methods; deep learning methods; the method of experimental research; and the method of comparative analysis of the results.

An intelligent module for monitoring the condition of agricultural crops in a greenhouse has been developed. It provides loading of IoT sensor data, checking their structure, preprocessing, feature normalization, training of a deep neural network, and classification of the greenhouse environment state. Software components for model training, state prediction, classification quality assessment, and result visualization have been implemented.

The research results can be used to develop software modules for monitoring the greenhouse environment, integrate IoT sensors with decision support systems, automate the control of growing parameters, and create educational and research prototypes in the field of smart agriculture.

Keywords: GREENHOUSE, AGRICULTURAL CROPS, CONDITION MONITORING, INTERNET OF THINGS, IoT SENSORS, DEEP NEURAL NETWORK, MACHINE LEARNING, CLASSIFICATION.

ЗМІСТ

Вступ.....	7
1 Аналіз проблемної області та постановка задачі	10
1.1 Опис предметної області	10
1.2 Аналіз існуючих підходів та методів	14
1.3 Постановка задачі дослідження.....	19
2 Проектування інтелектуального модуля моніторингу	22
2.1 Архітектура інтелектуального модуля моніторингу теплиці.....	22
2.2 Формування та попередня обробка даних IoT-сенсорів	28
2.3 Побудова моделі глибокої нейронної мережі	32
2.4 Алгоритм функціонування модуля моніторингу	37
3 Програмна реалізація та експериментальні дослідження	43
3.1 Реалізація програмного модуля	43
3.2 Організація експериментальних досліджень та навчання моделі.....	48
3.3 Аналіз результатів роботи модуля	54
Висновки	61
Список використаних джерел	64
Додаток А Основні параметри тепличного середовища.....	69
Додаток Б Лістинги програмного забезпечення	70
Додаток В Апробація результатів роботи.....	77

ВСТУП

Сільське господарство в умовах цифрової трансформації дедалі активніше інтегрує засоби Інтернету речей, методи машинного навчання та інтелектуальні програмні модулі для підвищення ефективності виробничих процесів. Особливої актуальності це набуває у тепличному господарстві, де стан сільськогосподарських культур безпосередньо залежить від температури, вологості повітря, вологості ґрунту, освітленості, вмісту поживних речовин, режиму поливу та інших параметрів мікросередовища. Навіть незначні відхилення цих показників від оптимальних значень можуть призводити до уповільнення росту рослин, зниження врожайності, погіршення якості продукції та збільшення витрат ресурсів [7, 9, 12, 16].

Традиційний підхід до контролю стану культур у теплиці значною мірою спирається на періодичні вимірювання, візуальні спостереження та досвід оператора. Такий підхід не завжди забезпечує достатню оперативність реагування, особливо в умовах великої кількості сенсорних даних і динамічної зміни параметрів середовища. У зв'язку з цим особливого значення набуває створення інтелектуальних модулів, здатних автоматично аналізувати показники тепличного середовища, виявляти характерні закономірності та визначати поточний стан сільськогосподарських культур на основі накопичених даних [2, 17, 19, 32].

Сучасні дослідження підтвердили доцільність застосування методів штучного інтелекту в аграрній сфері, зокрема для задач моніторингу рослин, прогнозування врожайності, виявлення хвороб, оцінювання стресових станів і підтримки рішень щодо поливу та керування мікрокліматом [4, 10, 18, 20, 26, 27]. Окремий інтерес становлять глибокі нейронні мережі, які забезпечують високу якість опрацювання складних, багатофакторних і частково нелінійних даних. На відміну від класичних алгоритмів, такі моделі здатні автоматично виявляти приховані залежності між параметрами середовища та цільовими станами об'єкта спостереження, що є особливо важливим для інтелектуального моніторингу у тепличних умовах [3, 5, 11, 21].

Актуальність теми зумовлена кількома чинниками. По-перше, тепличне господарство потребує постійного контролю умов вирощування, оскільки помилки у підтриманні мікроклімату прямо впливають на продуктивність. По-друге, поширення IoT-сенсорів дало змогу збирати великі обсяги даних у реальному часі, однак без інтелектуального аналізу такі дані не забезпечують достатньої практичної цінності [14, 24, 25]. По-третє, сучасні моделі глибокого навчання відкривають можливість переходу від простого накопичення показників до автоматизованого виявлення станів, ризиків і відхилень, що підвищує обґрунтованість управлінських рішень у тепличному виробництві [8, 13, 15, 22].

Аналіз наукових джерел показав, що питання використання штучних нейронних мереж у тепличних технологіях, системах зрошення, інтелектуальному агромоніторингу та комп'ютерному зорі для аграрних застосувань уже достатньо активно досліджується [1, 6, 12, 17, 23]. Водночас зберігається потреба у розробленні прикладних програмних модулів, орієнтованих саме на комплексний моніторинг стану сільськогосподарських культур у теплиці на основі даних IoT-сенсорів і глибокої нейронної мережі. Такий модуль має поєднувати збір і попередню обробку даних, аналітичну обробку, класифікацію або оцінювання станів, а також формування результатів у зручній для подальшого використання формі.

Отже, науково-практична проблема полягає у підвищенні ефективності моніторингу стану сільськогосподарських культур у тепличному середовищі за рахунок використання інтелектуальних методів аналізу даних. Її розв'язання пов'язане з розробленням такого підходу, який забезпечив би не лише збір показників із сенсорів, а й їх змістовну інтерпретацію в межах програмного модуля, здатного підтримувати прийняття рішень щодо контролю умов вирощування.

Метою кваліфікаційної роботи є розроблення інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі для аналізу даних IoT-сенсорів та визначення станів тепличного середовища.

Для досягнення поставленої мети сформовано такі завдання дослідження:

- проаналізувати предметну область моніторингу стану сільськогосподарських культур у теплиці;
- дослідити існуючі підходи до використання IoT, машинного та глибокого навчання в аграрних системах;
- визначити інформативні параметри тепличного середовища та структуру вхідних даних для моделювання;
- розробити архітектуру інтелектуального модуля моніторингу;
- побудувати модель глибокої нейронної мережі для аналізу сенсорних даних;
- реалізувати програмний модуль і провести експериментальне дослідження його роботи;
- оцінити результати функціонування розробленого рішення за показниками якості.

Об’єктом дослідження є процес моніторингу стану сільськогосподарських культур у тепличному середовищі.

Предметом дослідження є методи та програмні засоби інтелектуального аналізу даних IoT-сенсорів із використанням глибокої нейронної мережі для визначення стану сільськогосподарських культур у теплиці.

Результати кваліфікаційної роботи апробовані на міжнародній науковій інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення», м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р. (додаток В).

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

У сучасному тепличному виробництві підтримання стабільних умов вирощування є одним із ключових чинників отримання якісної продукції, прогнозованої врожайності та раціонального використання ресурсів. На відміну від відкритого ґрунту, теплиця є частково керованим середовищем, у якому більшість параметрів можуть контролюватися технічними засобами. Саме тому ефективність функціонування тепличного господарства значною мірою залежить від точності вимірювання, своєчасності аналізу та обґрунтованості керувальних дій [9, 12, 16].

Стан сільськогосподарських культур у теплиці формується під впливом багатьох факторів. До основних належать температура повітря, відносна вологість, вологість ґрунту, освітленість, концентрація поживних речовин, кислотність середовища, рівень вуглекислого газу, інтенсивність поливу та режим вентиляції. Кожен із цих параметрів окремо має важливе значення, однак реальний стан рослин визначається їх сукупною дією. Унаслідок цього навіть незначні, але систематичні відхилення кількох параметрів можуть викликати стрес рослин, пригнічення росту, підвищення вразливості до хвороб або зниження продуктивності [5, 12, 17, 20].

У традиційній практиці контроль стану культур часто базується на періодичних вимірюваннях і візуальному спостереженні. Такий підхід є зрозумілим і поширеним, однак він має суттєві обмеження. По-перше, ручний контроль не забезпечує безперервності спостереження. По-друге, велика кількість параметрів ускладнює оперативне виявлення прихованих закономірностей. По-третє, оцінка стану рослин значною мірою залежить від досвіду оператора, що знижує відтворюваність результатів. У сучасних умовах цього вже недостатньо, тому предметна область активно зміщується в бік інтелектуалізації процесів моніторингу [7, 19, 23].

Розвиток Інтернету речей у сільському господарстві суттєво змінив підходи до організації моніторингу. Використання мережі датчиків дало змогу

переходити від епізодичного спостереження до безперервного збору даних у реальному часі. У тепличних умовах це означає можливість регулярно фіксувати параметри мікроклімату, стан ґрунту, роботу виконавчих пристроїв і накопичувати історичні масиви даних для подальшого аналізу. Такі підходи вже використовуються в системах інтелектуального зрошення, керування мікрокліматом, прогнозування врожайності та підтримки агротехнологічних рішень [1, 7, 8, 15, 24, 32].

Водночас сам факт наявності великого обсягу даних не гарантує підвищення ефективності тепличного виробництва. Сенсорна інформація має цінність лише тоді, коли вона перетворюється на змістовні висновки про стан об'єкта спостереження. Саме тут виникає потреба у методах машинного та глибокого навчання, здатних виявляти складні залежності між вхідними показниками та фактичним станом рослин або тепличного середовища. Для аграрних застосувань це особливо важливо, оскільки вплив середовища на ріст культур має нелінійний, багатофакторний і часто зашумлений характер [3, 5, 10, 21].

У межах цієї предметної області під моніторингом стану сільськогосподарських культур доцільно розуміти не лише фіксацію окремих фізичних показників, а комплексне визначення поточного стану рослин на основі сукупності сенсорних ознак. Такий стан може інтерпретуватися як нормальний, стресовий, ризиковий або такий, що потребує корекції умов вирощування. Отже, моніторинг набуває не просто вимірювального, а аналітичного характеру. Його результатом стає не набір числових значень, а змістовний висновок, який може бути використаний для прийняття рішень [9, 17, 25].

Суттєвою особливістю предметної області є різноманітність вхідних даних. З одного боку, це часові ряди сенсорних вимірювань: температура, вологість, рівень освітленості, вміст мінеральних речовин, параметри поливу. З іншого боку, у сучасних системах можуть додатково використовуватись зображення рослин, ознаки з комп'ютерного зору, дані про спрацювання виконавчих пристроїв, журнали подій та історія зміни агротехнічних режимів [3, 10, 18, 26, 28]. Така різноманітність ускладнює побудову універсального алгоритму аналізу,

але водночас створює передумови для точнішого оцінювання стану культур.

Ще однією важливою рисою предметної області є динамічність. Стан тепличного середовища змінюється безперервно, іноді доволі швидко. Наприклад, підвищення температури у денний період може супроводжуватися одночасним зниженням вологості, а надмірний полив – зміною вологості ґрунту та впливом на доступність поживних речовин. У результаті система моніторингу має не лише фіксувати значення параметрів, а й враховувати часовий контекст їх зміни. Саме тому у дослідженнях усе частіше застосовуються методи глибокого навчання, здатні працювати зі складними багатовимірними залежностями та послідовностями спостережень [13, 18, 21].

Для тепличного господарства важливим є також ресурсний аспект. Підтримання оптимальних умов вимагає витрат води, електроенергії, добрив і часу персоналу. Надмірне зрошення, надлишкове вентильовання або нераціональне використання систем підігріву можуть призводити не лише до перевитрат, а й до погіршення стану рослин. Тому сучасний моніторинг орієнтований не тільки на спостереження, а й на підвищення ефективності використання ресурсів. Це зближує задачі моніторингу з задачами інтелектуального керування [1, 2, 7, 15].

Наукові джерела засвідчують, що у сфері розумного сільського господарства вже сформовано кілька стійких напрямів досліджень. Перший напрям пов'язаний із розробленням IoT-платформ для збору й передавання даних із теплиць. Другий орієнтований на застосування методів машинного навчання для класифікації станів, прогнозування врожайності, зрошення чи росту культур. Третій напрям пов'язаний із використанням глибокого навчання і комп'ютерного зору для виявлення хвороб, шкідників, ознак стресу та морфологічних змін рослин [8, 10, 11, 19, 22, 26]. Отже, предметна область є міждисциплінарною й поєднує елементи агрономії, сенсорики, аналізу даних, штучного інтелекту та програмної інженерії.

Попри значну кількість досліджень, проблема побудови прикладного інтелектуального модуля моніторингу залишається актуальною. Частина робіт зосереджена переважно на окремих аспектах: архітектурі IoT-системи,

комп'ютерному зорі, поливі або прогнозуванні врожайності. Натомість практична потреба полягає у створенні цілісного програмного рішення, яке поєднує збір даних, їх підготовку, інтелектуальний аналіз, визначення стану культур і представлення результату у формі, придатній для використання в тепличному середовищі [9, 12, 17, 29].

Таким чином, предметна область кваліфікаційної роботи характеризується такими основними особливостями:

- багатofакторністю стану тепличного середовища;
- необхідністю безперервного збору сенсорних даних;
- складністю виявлення прихованих залежностей між параметрами середовища та станом культур;
- потребою в автоматизованому аналітичному оцінюванні;
- орієнтацією на підтримку рішень щодо керування умовами вирощування.

У додатку А подано основні параметри тепличного середовища, які доцільно враховувати під час моніторингу стану сільськогосподарських культур.

Дані таблиці А.1 показують, що стан сільськогосподарських культур у теплиці визначається не окремим параметром, а сукупністю взаємопов'язаних факторів. Зміна температури, вологості, режиму поливу, концентрації поживних речовин і параметрів вентиляції комплексно впливає на ріст і розвиток рослин. Це обґрунтовує доцільність застосування інтелектуального підходу до моніторингу, за якого аналіз виконується на основі багатьох ознак одночасно.

Узагальнення наведених положень дає підстави стверджувати, що використання глибокої нейронної мережі в складі інтелектуального модуля моніторингу є обґрунтованим напрямом дослідження. Такий підхід забезпечує перехід від простого накопичення даних до їх інтерпретації, а відтак створює основу для підвищення точності оцінювання стану сільськогосподарських культур у теплиці.

1.2 Аналіз існуючих підходів та методів

Розвиток інтелектуальних систем моніторингу в аграрній сфері зумовлений поєднанням двох технологічних тенденцій: поширенням засобів Інтернету речей для збору даних та активним впровадженням методів машинного і глибокого навчання для їх аналітичної обробки. У межах тепличного господарства це дало змогу перейти від локального контролю окремих параметрів до побудови комплексних систем, здатних оцінювати стан середовища, прогнозувати ризики і підтримувати керування умовами вирощування [7, 9, 19].

У наукових джерелах можна виокремити кілька основних підходів до розв'язання задач моніторингу стану сільськогосподарських культур. Перший підхід базується на використанні IoT-інфраструктури для безперервного збору даних із сенсорів. Другий передбачає застосування класичних методів машинного навчання до табличних або часових даних. Третій орієнтований на використання глибоких нейронних мереж для обробки складних, багатовимірних і частково неструктурованих даних. Четвертий підхід є комбінованим і поєднує сенсорний моніторинг, комп'ютерний зір та інтелектуальні алгоритми керування [3, 10, 12, 18].

Найбільш базовим напрямом є побудова систем моніторингу на основі IoT-платформ. Такі системи орієнтовані на отримання даних у реальному часі від датчиків температури, вологості, вологості ґрунту, освітленості, концентрації газів, рівня рН та інших параметрів. Їх основною перевагою є можливість організувати постійне спостереження за тепличним середовищем, формувати історію вимірювань і передавати інформацію до аналітичного модуля або системи керування. У працях, присвячених моніторингу тепличного мікроклімату, показано, що така інфраструктура створює основу для подальшої автоматизації, але сама по собі не вирішує задачу інтерпретації стану рослин [9, 16, 24].

У межах цього підходу особливу роль відіграє якість сенсорних даних. Система моніторингу повинна забезпечувати достатню частоту вимірювання,

синхронізацію сигналів, стійкість до пропусків і збоїв зв'язку. Практика свідчить, що навіть у добре спроектованих IoT-рішеннях можуть виникати проблеми з шумом у даних, різною точністю датчиків і нестабільністю передавання. Через це в більшості сучасних систем окремим етапом виділяється попередня обробка даних: очищення, нормалізація, узгодження форматів, усунення аномалій і формування придатного до аналізу набору ознак [8, 14, 15].

Класичні методи машинного навчання стали наступним кроком у розвитку систем аграрного моніторингу. Для аналізу стану середовища та підтримки рішень щодо поливу, зрошення чи класифікації аграрних умов застосовуються дерева рішень, метод опорних векторів, випадкові ліси, k-середніх, лінійні моделі та ансамблеві підходи [1, 7, 15, 23]. Перевагою таких методів є відносна простота реалізації, нижча обчислювальна складність і можливість працювати навіть на невеликих наборах даних. Крім того, деякі з них забезпечують кращу інтерпретованість результатів, що є важливим у прикладних аграрних задачах.

Водночас класичні алгоритми мають низку обмежень. Їх ефективність часто суттєво залежить від ручного добору ознак, а здатність виявляти складні нелінійні залежності між параметрами середовища і станом культур є обмеженою. Це особливо помітно у випадках, коли стан рослин визначається не окремим показником, а комбінацією кількох факторів, що змінюються в часі. За таких умов класичні моделі можуть демонструвати нестійкі результати або потребувати значної додаткової інженерії ознак [5, 18, 21].

Саме з цієї причини все більшого поширення набувають методи глибокого навчання. У загальному випадку глибокі нейронні мережі відрізняються від традиційних моделей здатністю автоматично формувати внутрішні представлення даних на різних рівнях абстракції. Це дає змогу ефективніше працювати як із сенсорними багатовимірними даними, так і з зображеннями, часовими рядами або комбінованими наборами ознак [5, 11, 21]. У тепличному господарстві це особливо важливо, оскільки стан культур часто є результатом взаємодії декількох фізичних, біологічних та технологічних параметрів.

Одним із найбільш досліджених напрямів є використання глибоких нейронних мереж у задачах комп'ютерного зору для аграрних застосувань. У

таких роботах моделі застосовуються для виявлення бур'янів, хвороб, шкідників, пошкоджень листя, стадій росту та морфологічних особливостей рослин [4, 10, 26, 28]. Насамперед для цього використовуються згорткові нейронні мережі, які добре працюють із зображеннями. Перевагою таких систем є висока точність візуального аналізу, однак для завдань тепличного моніторингу вони не завжди є достатніми, якщо відсутній паралельний аналіз параметрів мікросередовища.

Окрему групу становлять підходи, у яких глибоке навчання застосовується до сенсорних табличних і часових даних. У роботах, присвячених тепличному моніторингу, прогнозуванню росту і врожайності, використовуються багатошарові нейронні мережі, рекурентні архітектури, а також комбіновані моделі типу TCN + RNN [6, 13, 17]. Такі рішення дозволяють враховувати часову динаміку зміни параметрів, що є суттєвою перевагою у порівнянні зі статичними класифікаторами. Разом із тим вони потребують більшого обсягу даних, ретельнішого налаштування гіперпараметрів і вищих обчислювальних ресурсів.

Важливе місце у дослідженнях займають роботи, присвячені саме тепличним системам. У таких джерелах аналізуються архітектури розумних теплиць, модулі контролю мікроклімату, моделі прогнозування росту культур, інтелектуальне зрошення та автоматизоване керування виконавчими пристроями [2, 8, 12, 17, 22, 25]. Перевагою цих робіт є прикладна орієнтованість: вони демонструють, як моделі вбудовуються в реальну структуру тепличного господарства. Однак у багатьох випадках акцент робиться або лише на керуванні середовищем, або лише на окремому аналітичному завданні. Через це питання комплексного моніторингу стану культур, що поєднує сенсорні дані, програмний модуль аналізу та класифікацію станів, залишається відкритим.

Серед сучасних підходів окремо слід відзначити системи інтелектуального зрошення. У таких рішеннях аналізуються вологість ґрунту, погодні умови, показники середовища та історичні дані з метою оптимізації використання води [1, 7, 8, 15]. Ці підходи мають високу практичну цінність, оскільки дозволяють знижувати витрати ресурсів. Водночас вони, як правило, зосереджуються на одному аспекті функціонування теплиці – водозабезпеченні. Для задачі моніторингу стану сільськогосподарських культур цього недостатньо, оскільки

фактичний стан рослин визначається не лише режимом поливу, а й температурою, освітленням, газовим режимом, поживним середовищем та іншими факторами.

Ще одним важливим напрямом є використання глибокого навчання для виявлення стресів, хвороб і шкідників. Такі підходи є особливо цінними в ситуаціях, коли необхідно оперативно виявляти загрозові стани до того, як вони призведуть до суттєвих втрат урожаю [20, 26, 28, 29]. Їхня перевага полягає у високій точності розпізнавання окремих ознак у зображеннях або сенсорних патернах. Недоліком є те, що вони часто орієнтовані на вузьке завдання, наприклад виявлення конкретного шкідника або хвороби, і не формують цілісної оцінки тепличного середовища.

Узагальнення джерел дозволяє виділити основні переваги сучасних підходів:

- автоматизований збір даних у реальному часі;
- підвищення точності оцінювання порівняно з ручним контролем;
- можливість виявлення прихованих залежностей у багатфакторних даних;
- підтримка прийняття рішень щодо керування мікрокліматом, поливом та іншими режимами;
- адаптація до масштабованих систем розумного сільського господарства.

Разом із цим виявлено й типові обмеження [3, 11, 18, 21]:

- орієнтація багатьох рішень лише на окремий аспект моніторингу;
- залежність результату від якості та повноти сенсорних даних;
- складність інтеграції різнорідних даних у межах одного програмного модуля;
- підвищені обчислювальні вимоги для глибоких моделей;
- недостатня універсальність окремих прикладних рішень при переході до інших культур або умов вирощування.

Для узагальнення подано порівняння основних підходів у вигляді таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика існуючих підходів до моніторингу стану культур у теплиці

Підхід	Характеристика	Переваги	Обмеження
IoT-моніторинг на основі сенсорів	Безперервний збір даних про параметри тепличного середовища	Оперативність, автоматизація вимірювань, накопичення історії даних	Не забезпечує глибокої інтерпретації стану без аналітичного модуля
Класичне машинне навчання	Аналіз табличних даних за допомогою SVM, Random Forest, дерев рішень та інших алгоритмів	Відносна простота реалізації, нижча обчислювальна складність	Обмежена здатність виявляти складні нелінійні залежності
Глибокі нейронні мережі для сенсорних даних	Багатошарова обробка багатовимірних і часових даних	Висока точність, автоматичне виявлення складних закономірностей	Потреба у більшому обсязі даних і налаштуванні
Глибоке навчання для комп'ютерного зору	Аналіз зображень рослин, листя, плодів, шкідників і ознак ураження	Висока ефективність у задачах візуального розпізнавання	Не охоплює повністю параметри мікроклімату
Комбіновані інтелектуальні системи	Поєднання IoT, глибокого навчання, комп'ютерного зору та засобів керування	Найбільш повний і практичний підхід	Висока складність реалізації та інтеграції

Отже, проведений аналіз показав, що найперспективнішим для задачі кваліфікаційної роботи є комбінований підхід, у якому дані IoT-сенсорів

використовуються як основа для інтелектуального аналізу за допомогою глибокої нейронної мережі. Саме такий підхід дозволяє врахувати багатофакторність тепличного середовища, знизити залежність від суб'єктивного контролю та перейти від простого вимірювання параметрів до автоматизованого визначення стану сільськогосподарських культур. Виявлені переваги й обмеження сучасних рішень створюють підстави для формулювання задачі дослідження та обґрунтування розроблення власного інтелектуального модуля моніторингу.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області та існуючих підходів показав, що сучасні тепличні системи вже забезпечують збір значного обсягу даних про параметри середовища, однак сама наявність цих даних не гарантує ефективного контролю стану сільськогосподарських культур. Основна проблема полягає в тому, що показники температури, вологості, освітленості, поливу, вмісту поживних речовин та інші параметри змінюються одночасно і взаємопов'язано. Унаслідок цього оцінювання стану культур на основі окремих порогових значень або простих правил не завжди забезпечує достатню точність і своєчасність виявлення небажаних змін [9, 12, 17].

У практичних умовах тепличного господарства виникає потреба не лише у фіксації значень сенсорів, а у формуванні змістовного висновку щодо поточного стану рослин. Такий висновок має враховувати багатофакторність середовища, наявність нелінійних залежностей між параметрами та можливі приховані закономірності в даних. Саме тому задача моніторингу стану культур доцільно розглядати як задачу інтелектуального аналізу даних, у якій на основі сукупності вхідних ознак визначається певний клас або стан об'єкта спостереження [3, 5, 21].

У межах цієї кваліфікаційної роботи розв'язується задача розроблення інтелектуального модуля, здатного приймати дані від IoT-сенсорів теплиці, виконувати їх попередню обробку, подавати їх на вхід глибокої нейронної

мережі та формувати результат у вигляді визначеного стану сільськогосподарських культур. Передбачається, що основою функціонування такого модуля є модель глибокого навчання, навчена на масиві сенсорних спостережень, де кожному набору параметрів відповідає певний цільовий стан.

Таким чином, **основна задача дослідження** полягає у підвищенні ефективності моніторингу стану сільськогосподарських культур у теплиці шляхом розроблення інтелектуального програмного модуля на основі глибокої нейронної мережі, який забезпечує автоматизоване аналізування даних IoT-сенсорів і визначення поточного стану тепличного середовища.

Для формалізації задачі доцільно визначити основні складові моделі дослідження. **Вхідними даними** є вектор ознак, сформований на основі сенсорної інформації теплиці. До його складу можуть входити [14, 24, 25]:

- температура повітря;
- відносна вологість повітря;
- вологість ґрунту;
- освітленість;
- рівень води;
- вміст азоту, фосфору, калію;
- кислотність середовища;
- параметри роботи виконавчих пристроїв;
- інші похідні або нормалізовані ознаки, отримані після попередньої обробки.

Вихідними даними є результат класифікації або оцінювання стану сільськогосподарських культур у теплиці. У межах роботи доцільно інтерпретувати його як один із визначених класів стану, наприклад:

- нормальний стан;
- стан ризику;
- стан, що потребує корекції умов вирощування;
- критичний стан.

Конкретна кількість класів визначатиметься структурою доступного набору даних і способом цільового маркування. Такий підхід дає змогу перейти

від аналізу окремих числових параметрів до більш зручного для практичного використання результату у вигляді інтегральної оцінки стану.

У процесі розв'язання поставленої задачі необхідно виконати кілька взаємопов'язаних етапів. Спочатку формується набір даних для моделювання. На цьому етапі виконується аналіз структури вхідної інформації, вибір інформативних параметрів, обробка пропусків, нормалізація та підготовка даних до подання на вхід моделі. Далі проєктується архітектура інтелектуального модуля, в якій визначаються основні функціональні компоненти: підсистема збору даних, підсистема попередньої обробки, модуль нейромережевого аналізу та модуль формування результату.

Наступним етапом є побудова моделі глибокої нейронної мережі. У межах цієї задачі така модель повинна забезпечувати:

- приймання багатовимірного вектора ознак;
- виявлення прихованих залежностей між параметрами;
- формування вихідного рішення щодо класу стану;
- можливість оцінювання якості за експериментальними даними.

Після побудови моделі здійснюється її навчання та тестування. Якість функціонування модуля доцільно оцінювати за метриками класифікації, зокрема accuracy, precision, recall та F1-score. Використання цих показників дає змогу оцінити не лише загальну частку правильних рішень, а й збалансованість якості класифікації для різних станів [5, 11, 31].

Узагальнюючи викладене, слід зазначити, що постановка задачі дослідження ґрунтується на переході від традиційного спостереження за окремими параметрами тепличного середовища до автоматизованого інтелектуального аналізу сукупності сенсорних даних. Саме такий підхід створює передумови для побудови програмного модуля, здатного забезпечити більш точне, оперативне й узгоджене визначення стану сільськогосподарських культур у теплиці.

2 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ МОНІТОРИНГУ

2.1 Архітектура інтелектуального модуля моніторингу теплиці

Проектування архітектури інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці виконано з урахуванням специфіки предметної області, структури вхідних даних і вимог до подальшої програмної реалізації. Основною метою такого проектування є формування цілісної системи, у межах якої сенсорні дані не лише накопичуються, а й перетворюються на змістовний результат у вигляді оцінки поточного стану культур. Такий підхід відповідає сучасним тенденціям розвитку IoT-рішень для тепличного господарства та інтелектуального аналізу аграрних даних [8, 9, 12, 16].

Архітектура інтелектуального модуля має забезпечувати послідовне виконання кількох функцій: збір даних із сенсорів, передавання та збереження вимірювань, попередню обробку вхідної інформації, аналіз за допомогою глибокої нейронної мережі, формування результату для користувача та, за потреби, передавання рекомендацій до підсистеми керування. У такій структурі кожний компонент виконує окрему роль, а разом вони утворюють єдиний цикл інтелектуального моніторингу.

На рисунку 2.1 представлено загальну архітектуру інтелектуального модуля моніторингу теплиці.

Подана на рисунку 2.1 архітектура відображає логіку проходження даних від рівня фізичних вимірювань до рівня аналітичного результату. На відміну від звичайних систем моніторингу, де основний акцент робиться на фіксації параметрів середовища, у запропонованій структурі центральне місце займає модуль інтелектуального аналізу. Саме він забезпечує інтерпретацію вхідних показників і визначення стану тепличного середовища в термінах, зручних для практичного використання.

На нижньому рівні архітектури розташовано сенсорний рівень. Його функція полягає у безперервному отриманні первинних даних про мікроклімат і параметри вирощування. До складу цього рівня входять датчики температури, відносної вологості повітря, вологості ґрунту, освітленості, рівня води,

кислотності та, за наявності, сенсори вмісту поживних речовин. Саме ці параметри формують первинний опис стану тепличного середовища та є основою для подальшого аналізу [1, 8, 15, 24].



Рисунок 2.1 – Узагальнена схема архітектури інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці

Наступним компонентом є компонент збору даних. Її призначення полягає у прийманні сигналів із датчиків, часовій синхронізації вимірювань і підготовці даних до передавання. У практичних умовах саме на цьому рівні можуть виникати технічні проблеми, пов’язані з пропусками, затримками або шумами в сигналах. Тому архітектура повинна враховувати, що частина даних надходитиме з різною частотою або потребуватиме додаткового узгодження [9, 16, 32].

Після збору інформації дані передаються через IoT-шлюз або мережевий канал до середовища збереження. У межах розроблюваного модуля таким середовищем може бути локальна база даних, хмарне сховище або файлова

структура для експериментального набору даних. Основною вимогою до цього рівня є збереження історії вимірювань, що дозволяє не лише аналізувати поточний стан, а й використовувати накопичені спостереження для навчання нейромережевої моделі. Таким чином, архітектура орієнтована як на онлайн-моніторинг, так і на офлайн-навчання [7, 19, 25].

Особливе місце в архітектурі займає модуль попередньої обробки даних. Його включення є обов'язковим, оскільки сенсорна інформація в реальних умовах часто містить пропуски, аномальні значення, шум або несумісність форматів. У межах цього модуля виконуються очищення даних, перевірка повноти, нормалізація числових параметрів, узгодження масштабу ознак та формування підготовленого вектора вхідних даних для нейромережі. Без цього етапу якість подальшого аналізу істотно знижується [5, 8, 14, 21].

У таблиці 2.1 подано функціональне призначення основних компонентів архітектури.

Таблиця 2.1 – Основні компоненти архітектури інтелектуального модуля моніторингу

Компонент	Основне призначення	Результат роботи компонента
Сенсорний рівень	Вимірювання параметрів тепличного середовища	Первинні значення температури, вологості, освітленості, рН, NPK та інших показників
Підсистема збору даних	Приймання та синхронізація вимірювань	Узгоджений потік сенсорних даних
Канал передавання та збереження	Передавання і накопичення інформації	Історія спостережень у базі даних або наборі файлів
Модуль попередньої обробки	Очищення, нормалізація, формування ознак	Підготовлений вектор вхідних параметрів

Продовження таблиці 2.1

Модуль глибокої нейронної мережі	Аналіз взаємозв'язків між параметрами	Обчислений прогноз або клас стану
Модуль визначення стану	Інтерпретація виходу моделі	Стан тепличного середовища або культур
Інтерфейс користувача	Відображення результатів моніторингу	Повідомлення, звіти, попередження
Підсистема підтримки рішень	Формування рекомендацій або керувальних дій	Рекомендації щодо поливу, вентиляції, освітлення тощо

Як видно з таблиці 2.1, архітектура має модульну структуру. Це означає, що окремі компоненти можуть змінюватися або розширюватися без порушення загальної логіки функціонування. Зокрема, набір сенсорів може бути збільшений, формат збереження даних може бути замінений, а нейромережева модель може вдосконалюватися без потреби повного перепроєктування всієї системи. Така модульність є важливою властивістю прикладного програмного рішення [2, 8, 31].

Центральним компонентом архітектури є модуль глибокої нейронної мережі. Його функція полягає у виявленні прихованих залежностей між параметрами середовища та цільовими станами сільськогосподарських культур. На відміну від порогових правил або простих евристичних рішень, нейронна мережа дозволяє враховувати складний характер взаємодії ознак. Наприклад, підвищення температури саме по собі не завжди означає критичний стан, але у поєднанні зі зниженням вологості й недостатнім поливом може свідчити про стрес рослин. Саме такі багатофакторні залежності мають бути виявлені в межах розроблюваної моделі [3, 11, 13, 21].

Після роботи нейромережевого модуля виконується інтерпретація

результату. На цьому етапі вихід моделі подається у вигляді певного стану: нормального, ризикового, такого, що потребує корекції, або критичного. За потреби цей результат доповнюється текстовим повідомленням або короткою рекомендацією для оператора теплиці. У такий спосіб система забезпечує не просто числовий прогноз, а формує практично значущий висновок.

Для більш детального уявлення про внутрішню організацію аналітичної частини на рисунку 2.2 подано структуру програмно-аналітичного модуля.

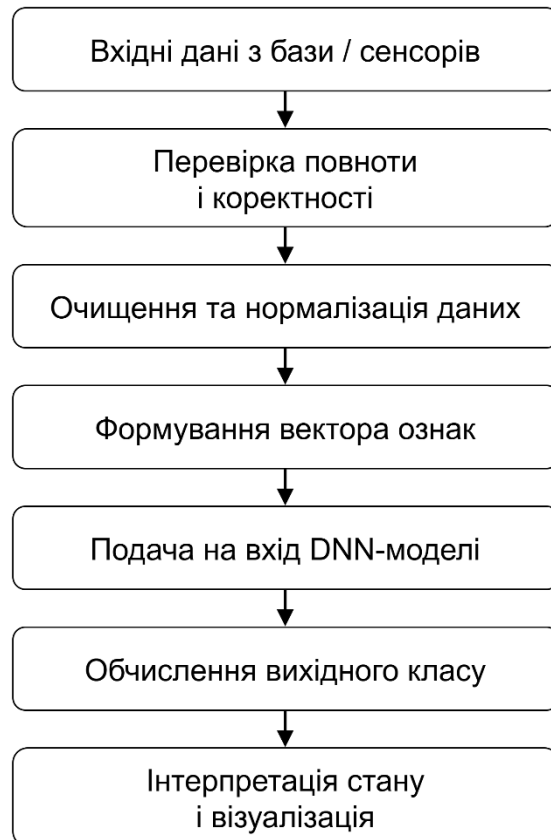


Рисунок 2.2 – Структурна схема програмно-аналітичного модуля

Рисунок 2.2 показує, що програмно-аналітичний модуль не обмежується лише викликом нейромережі. Його структура охоплює весь шлях даних від первинної перевірки до формування кінцевого результату. Це є важливим з погляду програмної реалізації, оскільки якість функціонування системи визначається не тільки архітектурою моделі, а й коректністю підготовки вхідної інформації.

З позиції системного проєктування запропоновану архітектуру доцільно розглядати як багаторівневу. Перший рівень є рівнем збору даних, другий –

рівнем збереження і передавання, третій – рівнем аналітичної обробки, четвертий – рівнем представлення результатів і підтримки рішень. Такий поділ є виправданим, оскільки дозволяє логічно відокремити фізичну інфраструктуру теплиці від програмної частини інтелектуального модуля. Крім того, він спрощує майбутнє масштабування системи, наприклад при збільшенні кількості сенсорів, підключенні нових культур або інтеграції з мобільним інтерфейсом [8, 15, 17].

У таблиці 2.2 подано узагальнений розподіл функцій за рівнями архітектури.

Дані таблиці 2.2 дають змогу стверджувати, що запропонована архітектура є збалансованою з погляду функціонального наповнення. Вона охоплює повний цикл обробки інформації та забезпечує логічний зв'язок між технічним середовищем теплиці та інтелектуальним програмним модулем.

Таблиця 2.2 – Рівні архітектури модуля та їх функції

Рівень архітектури	Компоненти	Основні функції
Рівень збору даних	Сенсори, контролери, IoT-модулі	Вимірювання параметрів середовища та первинне зчитування
Рівень передавання і збереження	Шлюз, мережа, база даних	Передавання, накопичення та збереження історії спостережень
Рівень аналітичної обробки	Модуль попередньої обробки, DNN-модель	Підготовка даних, класифікація та оцінювання стану
Рівень представлення результатів	Інтерфейс користувача, модуль повідомлень	Відображення результатів, попереджень і рекомендацій

Таким чином, у підрозділі сформовано архітектуру модуля моніторингу теплиці, яка включає сенсорний рівень, підсистему збору й збереження даних, модуль попередньої обробки, модуль глибокої нейронної мережі, засоби інтерпретації результату та підсистему підтримки рішень. Запропонована

структура забезпечує можливість переходу від простого накопичення сенсорних даних до автоматизованого визначення стану сільськогосподарських культур.

2.2 Формування та попередня обробка даних IoT-сенсорів

Ефективність інтелектуального модуля моніторингу значною мірою визначається якістю вхідних даних. Навіть за наявності сучасної архітектури глибокої нейронної мережі некоректно сформований або недостатньо підготовлений набір ознак призводить до зниження точності класифікації, нестійкості результатів і погіршення здатності моделі узагальнювати нові спостереження. Саме тому в межах проектування модуля окремим етапом виділено формування та попередню обробку даних IoT-сенсорів, що надходять із тепличного середовища [5, 8, 12, 19].

У розроблюваному рішенні джерелом вхідної інформації є набір даних, сформований на основі сенсорних вимірювань, характерних для задач розумного агромоніторингу. До таких даних належать температура повітря, відносна вологість повітря, вологість ґрунту, рівень води, значення рН, а також показники, пов'язані з мінеральним живленням, зокрема вміст азоту, фосфору й калію. Подібний склад ознак є типовим для систем інтелектуального землеробства й тепличного контролю, оскільки саме ці параметри дають змогу сформувати достатньо повне уявлення про поточний стан умов вирощування [1, 9, 14, 15].

У межах кваліфікаційної роботи для побудови та тестування моделі використано відкритий набір IoT Agriculture 2024 Dataset [14]. Застосування такого набору є виправданим, оскільки він містить структуровані сенсорні дані, придатні для побудови інтелектуального модуля аналізу тепличного або близького до тепличного аграрного середовища. Водночас при формуванні програмного модуля передбачено, що структура вхідних даних може бути адаптована і до реальної системи теплиці, де показники надходять у режимі наближеному до реального часу.

На рисунку 2.3 подано загальну схему формування та підготовки даних для нейромережевого аналізу.



Рисунок 2.3 – Схема формування та попередньої обробки даних IoT-сенсорів

Як показано на рисунку 2.3, підготовка даних включає не лише технічне очищення, а й формування цільової змінної, без якої побудова моделі класифікації є неможливою. Такий підхід узгоджується з сучасною практикою машинного навчання, де якість результату залежить не тільки від моделі, а і від коректності побудови навчального набору [5, 18, 21].

Першим етапом є збір і первинне завантаження даних. На цьому кроці зчитуються записи сенсорних вимірювань, перевіряється наявність потрібних полів, їх формат і типи значень. Для моделювання важливо, щоб числові параметри були подані в уніфікованому вигляді, а назви атрибутів відповідали логіці подальшої обробки. На практиці це означає приведення назв стовпців до єдиного формату, перетворення текстових значень у числові, усунення зайвих службових полів і формування табличної структури, з якою може працювати програмний модуль.

Після цього виконується перевірка повноти даних. У сенсорних наборах часто трапляються пропуски, які можуть бути зумовлені нестабільністю зв'язку,

тимчасовою відмовою датчика або неповним записом вимірювань. Наявність великої кількості пропущених значень негативно впливає на якість моделі, оскільки нейронна мережа не здатна коректно навчатися на неповних векторах ознак без додаткової обробки. У зв'язку з цим пропуски повинні бути або усунені, або компенсовані. Найбільш доцільними підходами є заповнення пропусків середнім або медіанним значенням для відповідної ознаки, а також видалення окремих записів, якщо їх частка є незначною [8, 14, 19].

Окрім пропусків, у даних можуть бути присутні аномальні значення. Для тепличного середовища такими можуть бути фізично малоімовірні температури, від'ємні значення вологості, надмірно високі рівні поживних речовин або інші спотворення, що виникають унаслідок збою датчика. Якщо такі записи не усунути, вони впливають на масштаб ознак і викривляють навчальний процес. Тому на етапі очищення доцільно використовувати перевірку допустимих діапазонів для кожного параметра та вилучення або корекцію значень, що виходять за межі фізично обґрунтованих інтервалів [9, 12, 17].

Одним із найважливіших етапів є нормалізація числових параметрів. У сенсорних даних різні ознаки часто мають різні масштаби. Наприклад, температура може змінюватися в межах десятків градусів, вологість – у межах від 0 до 100, а значення концентрацій NPK можуть бути іншого порядку. Якщо такі параметри подавати на вхід моделі без масштабування, ознаки з більшими числовими значеннями можуть непропорційно впливати на процес навчання. Тому доцільно використовувати нормалізацію типу Min–Max або стандартизацію, що приводить значення до спільного масштабу [5, 11, 21].

Після очищення та нормалізації необхідно сформувавши цільову змінну. Оскільки в межах роботи розв'язується задача класифікації стану сільськогосподарських культур, кожному запису сенсорних даних повинна відповідати мітка класу. У прикладному варіанті такі класи можуть бути визначені як нормальний стан, стан ризику, стан корекції та критичний стан. Формування міток може виконуватися на основі сукупності граничних умов для параметрів середовища або на основі готового маркування в наборі даних. Саме на цьому етапі відбувається перехід від простої реєстрації фізичних величин до

задачі інтелектуального оцінювання стану [9, 17, 25].

У таблиці 2.3 наведено приклад логіки формування цільових станів на основі сукупності параметрів.

Таблиця 2.3 – Приклад інтерпретації станів тепличного середовища

Клас стану	Характеристика	Інтерпретація для системи
Нормальний стан	Параметри середовища перебувають у допустимих межах	Умови вирощування є стабільними
Стан ризику	Один або кілька параметрів наближаються до критичних значень	Потрібне підвищене спостереження
Стан корекції	Виявлено відхилення, які потребують зміни режиму поливу, вентиляції або живлення	Необхідне втручання оператора або системи
Критичний стан	Сукупність параметрів створює загрозу пригнічення росту чи пошкодження культур	Потрібна термінова корекція умов

Наведена в таблиці 2.3 схема не є жорстко фіксованою, однак вона добре відображає логіку роботи інтелектуального модуля. Замість аналізу кожного сенсорного параметра окремо система формує інтегральну оцінку поточного стану, що є значно зручнішим для практичного використання в теплиці.

Після підготовки ознак і формування міток набір даних розподіляється на підмножини для навчання, валідації та тестування. Такий поділ є стандартним для задач глибокого навчання і дозволяє уникнути переоцінювання якості моделі. Навчальна вибірка використовується для налаштування ваг нейронної мережі, валідаційна – для контролю процесу навчання і добору параметрів, а тестова – для підсумкового оцінювання. Доцільним є співвідношення 70% / 15% / 15 % [9, 13, 21].

З погляду програмної реалізації етап попередньої обробки виконує роль буфера між фізичним середовищем теплиці та аналітичною моделлю. Це

означає, що нейронна мережа працює не з «сирими» сенсорними даними, а з підготовленим узгодженим представленням. Такий підхід підвищує стійкість системи, зменшує вплив випадкових спотворень і покращує відтворюваність результатів [8, 19, 31].

Таким чином, у підрозділі сформовано підхід до підготовки даних IoT-сенсорів для подальшого нейромережевого аналізу. Він включає збір і перевірку структури даних, усунення пропусків і аномалій, нормалізацію числових ознак, формування цільових класів і розподіл набору на навчальну, валідаційну та тестову частини.

2.3 Побудова моделі глибокої нейронної мережі

У межах кваліфікаційної роботи для інтелектуального моніторингу стану сільськогосподарських культур у теплиці обрано модель глибокої нейронної мережі прямого поширення типу DNN (Deep Neural Network). Такий вибір зумовлений структурою вхідних даних. У розроблюваній системі на вхід подається не зображення, а набір числових сенсорних параметрів: температура повітря, відносна вологість, вологість ґрунту, освітленість, рН, вміст азоту, фосфору, калію, рівень води та інші похідні ознаки. Для таких даних доцільно використовувати саме багат шарову повнозв'язну нейронну мережу, оскільки вона добре працює з табличними векторами ознак і дозволяє виявляти нелінійні залежності між параметрами середовища та цільовими станами [5, 12, 21].

Метою побудови моделі є реалізація класифікатора, який за вхідним вектором сенсорних ознак визначає один із наперед заданих станів тепличного середовища. У роботі прийнято чотирикласову схему класифікації: нормальний стан; стан ризику; стан корекції; критичний стан.

Отже, модель повинна реалізувати відображення від багатовимірного простору вхідних ознак до множини з чотирьох класів. Це означає, що вихідний шар нейронної мережі повинен містити 4 нейрони, кожен із яких відповідає одному класу.

На рисунку 2.4 подано структуру моделі глибокої нейронної мережі.



Рисунок 2.4 – Структура моделі глибокої нейронної мережі для класифікації стану культур

Як показано на рисунку 2.4, модель має послідовну багатошарову структуру. Її основу складають: вхідний шар; три приховані повнозв'язні шари; вихідний шар класифікації.

Вхідний шар приймає вектор числових ознак, сформований після попередньої обробки даних. Кількість нейронів у вхідному шарі дорівнює кількості вхідних ознак. Якщо, наприклад, у наборі даних використовується 9 основних сенсорних параметрів, то вхідний шар матиме 9 входів. Якщо після інженерії ознак додаються похідні характеристики, кількість входів відповідно збільшується.

Таким чином, вхідний шар не виконує обчислень у звичному сенсі, а лише передає нормалізований вектор ознак до прихованих шарів моделі.

Для розроблюваної системи обрано архітектуру з трьома прихованими шарами: перший прихований шар – 64 нейрони; другий прихований шар – 32 нейрони; третій прихований шар – 16 нейронів.

Такий вибір є практично доцільним. Перший шар має достатню ширину

для виявлення базових комбінацій ознак. Другий і третій шари поступово зменшують розмір внутрішнього представлення, зберігаючи найбільш інформативні залежності. У результаті мережа формує компактне представлення стану тепличного середовища перед класифікацією.

Для всіх прихованих шарів використано функцію активації ReLU (Rectified Linear Unit). Її вибір пояснюється кількома причинами: проста обчислювальна реалізація; ефективне навчання глибоких мереж; зменшення ризику затухання градієнта; добра практична придатність для табличних даних [11].

Оскільки розв'язується задача багатокласової класифікації, вихідний шар містить 4 нейрони і використовує функцію активації Softmax. Це дозволяє перетворити вихідні значення мережі у вектор ймовірностей належності до кожного класу. Після цього клас із найбільшою ймовірністю приймається як результат класифікації.

Формально вихідний вектор моделі можна подати як

$$\hat{y} = (\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4)$$

де $\hat{y}_i$ – оцінена ймовірність належності до i -го класу стану.

Для зменшення ризику перенавчання доцільно включити в архітектуру механізми регуляризації. У роботі передбачено використання шару Dropout після другого прихованого шару зі значенням 0,2. Це означає, що під час навчання випадково вимикається 20 % нейронів відповідного шару. Такий підхід знижує залежність моделі від окремих локальних комбінацій ознак і підвищує її здатність узагальнювати нові спостереження.

Крім того, у процесі навчання доцільно застосовувати Early Stopping, тобто дострокове завершення навчання, якщо значення втрат на валідаційній вибірці перестає покращуватися протягом кількох епох. Це також сприяє підвищенню стійкості моделі.

У таблиці 2.5 наведено основні параметри архітектури моделі.

Таблиця 2.5 – Основні параметри архітектури DNN-моделі

Компонент моделі	Характеристика
Тип моделі	DNN–мережа
Тип вхідних даних	Нормалізований вектор сенсорних ознак
Кількість прихованих шарів	3
Кількість нейронів у прихованих шарах	64, 32, 16
Функція активації прихованих шарів	ReLU
Регуляризація	Dropout 0,2
Кількість вихідних нейронів	4
Функція активації вихідного шару	Softmax
Тип задачі	Багатокласова класифікація

Як видно з таблиці 2.5, архітектура моделі є достатньо компактною. Це важливо для прикладної системи моніторингу, оскільки надмірно складна модель підвищує обчислювальні витрати, але не завжди дає істотний приріст точності на табличних сенсорних даних.

Для навчання моделі обрано функцію втрат categorical cross-entropy, оскільки вона є стандартною для задач багатокласової класифікації з вихідним шаром Softmax. Ця функція оцінює відмінність між фактичним класом і прогнозованим розподілом ймовірностей.

Як алгоритм оптимізації використано Adam. Його вибір пояснюється тим, що Adam добре працює на задачах з великою кількістю параметрів, автоматично адаптує швидкість навчання для різних ваг і забезпечує швидку збіжність у практичних задачах глибокого навчання [5, 21].

Основні параметри навчання прийнято такими:

- learning rate = 0,001;
- batch size = 32;
- кількість епох = 50;
- early stopping patience = 5.

Значення learning rate 0,001 є більш стабільним для задачі цієї складності,

ніж занадто велике початкове значення. Batch size 32 забезпечує прийнятний компроміс між швидкістю навчання та стабільністю оновлення ваг.

Робота моделі відбувається послідовно. На вхід подається нормалізований вектор сенсорних ознак. Кожен прихований шар виконує лінійне перетворення вхідних даних, після чого застосовується нелінійна функція активації ReLU. Після проходження всіх прихованих шарів формується узагальнене внутрішнє представлення вхідного стану. На вихідному шарі виконується перетворення цього представлення у вектор ймовірностей класів. Найбільше значення цього вектора визначає кінцевий клас стану.

На рисунку 2.5 подано послідовність проходження даних через модель.

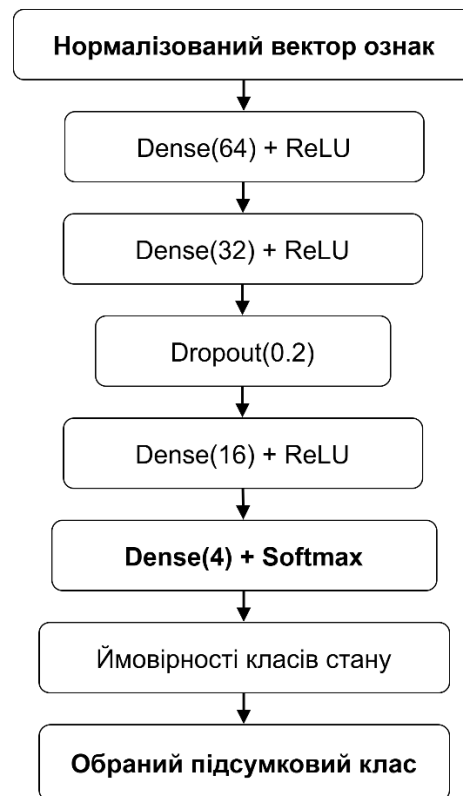


Рисунок 2.5 – Схема обробки вхідних даних у DNN-моделі

Як показано на рисунку 2.5, модель має чітку та послідовну структуру без зайвих компонентів. Це робить її зрозумілою з погляду реалізації та придатною для використання в складі прикладного програмного модуля.

Для оцінювання якості побудованої нейронної мережі доцільно використовувати такі метрики:

- ассигасу – загальна частка правильно класифікованих записів;

- precision – точність визначення окремих класів;
- recall – повнота виявлення класів;
- F1-score – узагальнений показник якості класифікації [5, 11].

Застосування лише ассигасу є недостатнім, оскільки окремі класи можуть бути представлені нерівномірно. Тому в роботі доцільно оцінювати модель комплексно.

Таким чином, у даному підрозділі 2.3 побудовано конкретну модель глибокої нейронної мережі для класифікації стану сільськогосподарських культур у теплиці. Визначено тип моделі, структуру прихованих і вихідного шарів, функції активації, параметри регуляризації, функцію втрат, оптимізатор і базові налаштування навчання. Обрана архітектура є достатньо простою для практичної реалізації та водночас придатною для виявлення складних залежностей у сенсорних даних тепличного середовища.

2.4 Алгоритм функціонування модуля моніторингу

Функціонування інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці ґрунтується на послідовному виконанні етапів збору, підготовки, аналізу та інтерпретації даних. На відміну від звичайного контролю окремих параметрів, у запропонованому рішенні реалізовано цілісний алгоритм, у межах якого сенсорні вимірювання перетворюються на підсумкову оцінку стану тепличного середовища. Це дає змогу автоматизувати процес спостереження, зменшити вплив людського фактора та забезпечити більш швидке виявлення ризикових ситуацій [8, 9, 12].

Алгоритм функціонування модуля охоплює два взаємопов'язані режими: режим підготовки та навчання моделі; режим робочого моніторингу після навчання.

На рисунку 2.6 подано загальну схему алгоритму функціонування модуля моніторингу.



Рисунок 2.6 – Схема алгоритму функціонування інтелектуального модуля моніторингу [33]

Як показано на рисунку 2.6, алгоритм має лінійно-розгалужену структуру. Основна гілка відповідає стандартному сценарію проходження даних через модуль, а додаткова гілка використовується у випадках, коли вхідний запис містить пропуски або некоректні значення. Такий підхід підвищує стійкість програмного рішення до типових проблем сенсорного середовища.

Алгоритм роботи інтелектуального модуля доцільно подати у вигляді такої послідовності кроків [33].

Крок 1. Отримання вхідних даних. На першому етапі модуль приймає поточний набір значень від IoT-сенсорів теплиці. До складу вхідного запису можуть входити температура повітря, вологість повітря, вологість ґрунту, освітленість, рН, NPK, рівень води та інші параметри, передбачені структурою системи.

Крок 2. Перевірка повноти та коректності даних. Виконується контроль наявності всіх обов'язкових полів, перевірка типів значень та належності показників до допустимих інтервалів. Якщо виявлено пропуски або аномальні значення, модуль переходить до процедури корекції даних.

Крок 3. Обробка некоректних значень. У разі виявлення пропусків або аномалій виконується один із двох варіантів: заповнення окремих відсутніх значень за заздалегідь визначеним правилом; відхилення запису, якщо рівень некоректності є критичним.

Крок 4. Попередня обробка даних. Після перевірки запис очищається від службових або зайвих атрибутів, а структура приводиться до формату, який використовується моделлю.

Крок 5. Нормалізація ознак. Числові параметри перетворюються до узгодженого масштабу. Це необхідно для того, щоб жодна з ознак не домінувала в процесі роботи нейронної мережі лише через більший числовий діапазон.

Крок 6. Формування вектора ознак. Підготовлені параметри об'єднуються у впорядкований вектор фіксованої довжини, який відповідає входу моделі DNN.

Крок 7. Подача вектора на вхід нейронної мережі. Сформований вектор ознак передається до навченої моделі глибокої нейронної мережі.

Крок 8. Обчислення вихідного класу. Модель виконує пряме поширення сигналу через приховані шари та вихідний шар Softmax. У результаті формується вектор ймовірностей належності до кожного класу стану.

Крок 9. Вибір підсумкового стану. Підсумковим результатом приймається клас із найбільшою оціненою ймовірністю. Це може бути нормальний стан, стан ризику, стан корекції або критичний стан.

Крок 10. Інтерпретація результату. Після визначення класу формується змістовне пояснення результату. Наприклад, якщо отримано критичний стан,

система відображає попередження про необхідність термінового втручання.

Крок 11. Формування повідомлення або рекомендації. На основі визначеного стану користувачеві передається повідомлення, попередження або коротка рекомендація щодо зміни режиму поливу, вентиляції чи інших параметрів.

Крок 12. Збереження результату. Поточний запис і результат класифікації зберігаються в системі для подальшого аналізу, побудови історії спостережень і використання в звітності.

Крок 13. Завершення циклу та очікування нових даних. Після завершення обробки поточного запису система переходить у режим очікування наступного набору сенсорних вимірювань.

Наведена послідовність дій визначає повний цикл роботи програмного модуля в режимі експлуатації. Оскільки модуль моніторингу використовує попередньо навчено модель, доцільно коротко подати і послідовність дій для етапу навчання. Цей алгоритм виконується не під час кожного циклу моніторингу, а на етапі підготовки системи.

Відповідно до рисунка 2.7, навчання моделі включає підготовку даних, побудову архітектури, налаштування параметрів, оцінювання результату та збереження остаточної моделі для подальшого використання в системі моніторингу.

Послідовність кроків алгоритму навчання є такою:

- завантажити набір сенсорних даних;
- виконати очищення, нормалізацію та підготовку ознак;
- сформувати цільові мітки класів;
- розподілити набір даних на train, validation і test;
- побудувати архітектуру DNN-моделі;
- виконати навчання моделі на train-вибірці;
- оцінити якість на validation і test-вибірках;
- у разі незадовільної якості змінити параметри моделі та повторити навчання;
- зберегти модель для використання в системі.



Рисунок 2.7 – Схема алгоритму навчання DNN-моделі для модуля моніторингу

Запропонований алгоритм має кілька важливих особливостей.

По-перше, він враховує можливість появи некоректних або неповних сенсорних значень. Це є суттєвим для тепличних IoT-систем, у яких нестабільність показників або пропуски не є винятком.

По-друге, алгоритм чітко відокремлює етап попередньої обробки від етапу нейромережевого аналізу. Завдяки цьому модель працює лише з узгодженими та підготовленими даними, що підвищує стійкість результатів.

По-третє, результатом роботи модуля є не просто числовий вектор, а інтерпретований клас стану, придатний для практичного використання в теплиці. Це означає, що алгоритм реалізує не лише аналітичну, а й прикладну функцію підтримки рішень.

Таким чином, у даному підрозділі сформовано алгоритм функціонування інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці. Алгоритм подано як послідовність конкретних кроків, а також у вигляді схем для режиму робочого моніторингу та режиму навчання моделі. Запропонована логіка роботи забезпечує повний цикл обробки сенсорних даних – від їх надходження до формування підсумкового стану, повідомлення користувачу та збереження результату.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Реалізація програмного модуля

Реалізацію інтелектуального модуля виконано мовою Python із використанням бібліотек pandas, NumPy, scikit-learn, TensorFlow/Keras та Matplotlib. Такий вибір зумовлений наявністю готових засобів для завантаження сенсорних даних, їх попередньої обробки, побудови нейронної мережі та візуалізації результатів.

Основні програмні засоби, використані в реалізації, подано в таблиці 3.1.

Таблиця 3.1 – Програмні засоби реалізації модуля

Компонент	Засіб	Призначення
Мова програмування	Python 3	Реалізація логіки системи
Робота з таблицями	pandas	Завантаження та обробка CSV-даних
Числові операції	NumPy	Формування масивів ознак
Попередня обробка	scikit-learn	Нормалізація, поділ вибірки
Нейронна мережа	TensorFlow / Keras	Побудова та навчання DNN-моделі
Візуалізація	Matplotlib	Побудова графіків результатів

Згідно з таблицею 3.1, для реалізації використано стандартний стек засобів аналізу даних і глибокого навчання, що забезпечує відтворюваність і практичну придатність розробленого рішення.

Структуру програмного проєкту подано на рисунку 3.1.

```
main.py
├─ data_loader.py
├─ preprocessing.py
├─ model.py
├─ train.py
├─ predict.py
└─ visualize.py
```

Рисунок 3.1 – Узагальнена структура програмного проєкту

Як показано на рисунку 3.1, кожен файл реалізує окрему функцію системи. Основний сценарій роботи запускається через `main.py`, а спеціалізовані модулі відповідають за окремі етапи обробки даних і роботи моделі.

На першому етапі реалізовано завантаження набору даних і перевірку його структури. Для цього сформовано перелік обов'язкових полів, без яких робота модуля є неможливою. Приклад реалізації наведено в лістингу 3.1.

Лістинг 3.1 – Завантаження та перевірка структури набору даних

```
import pandas as pd

REQUIRED_COLUMNS = [
    "temperature", "humidity_air", "soil_moisture",
    "light", "ph", "nitrogen", "phosphorus",
    "potassium", "water_level", "label"
]

def load_data(file_path: str) -> pd.DataFrame:
    df = pd.read_csv(file_path)
    missing = [col for col in REQUIRED_COLUMNS if col not in df.columns]
    if missing:
        raise ValueError(f"Відсутні обов'язкові поля: {missing}")
    return df
```

Лістинг 3.1 показує, що реалізований модуль одразу контролює наявність основних сенсорних параметрів і цільової мітки. Це унеможлиблює запуск обробки на некоректно сформованому наборі даних.

Після завантаження даних виконується попередня обробка: заповнення пропусків і нормалізація числових ознак. Приклад реалізації цього етапу подано в лістингу 3.2.

Лістинг 3.2 – Попередня обробка та нормалізація сенсорних даних

```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler

FEATURE_COLUMNS = [
    "temperature", "humidity_air", "soil_moisture",
    "light", "ph", "nitrogen", "phosphorus",
    "potassium", "water_level"
]

def preprocess_data(df):
    X = df[FEATURE_COLUMNS].copy()
    y = df["label"].copy()

    imputer = SimpleImputer(strategy="median")
    X_imputed = imputer.fit_transform(X)
```

```

scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X_imputed)

return X_scaled, y, scaler, imputer

```

Як видно з лістингу 3.2, для обробки пропусків використано медіанне заповнення, а для приведення ознак до спільного масштабу – нормалізацію Min–Max. Саме такий формат даних далі подається на вхід нейронної мережі.

Архітектуру DNN-моделі реалізовано відповідно до структури, визначеної в підрозділі 2.3. Вона включає три приховані шари, шар Dropout і вихідний шар Softmax. Приклад реалізації наведено в лістингу 3.3.

Лістинг 3.3 – Реалізація DNN-моделі

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

def build_model(input_dim: int, num_classes: int):
    model = Sequential([
        Dense(64, activation="relu", input_shape=(input_dim,)),
        Dense(32, activation="relu"),
        Dropout(0.2),
        Dense(16, activation="relu"),
        Dense(num_classes, activation="softmax")
    ])

    model.compile(
        optimizer="adam",
        loss="categorical_crossentropy",
        metrics=["accuracy"]
    )
    return model

```

Лістинг 3.3 показує, що модель реалізовано у компактному вигляді, придатному для задачі багатокласової класифікації стану тепличного середовища. Така структура забезпечує баланс між простотою реалізації та здатністю виявляти складні залежності в сенсорних даних.

Навчання моделі реалізовано з використанням поділу даних на вибірки, one-hot кодування класів і механізму Early Stopping. Відповідний фрагмент подано в лістингу 3.4.

Лістинг 3.4 – Навчання нейронної мережі

```

from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical

```

```

from tensorflow.keras.callbacks import EarlyStopping
from sklearn.preprocessing import LabelEncoder

def train_model(X, y):
    encoder = LabelEncoder()
    y_encoded = encoder.fit_transform(y)
    y_cat = to_categorical(y_encoded)

    X_train, X_test, y_train, y_test = train_test_split(
        X, y_cat, test_size=0.2, random_state=42, stratify=y_encoded
    )

    model = build_model(input_dim=X.shape[1], num_classes=y_cat.shape[1])

    early_stop = EarlyStopping(
        monitor="val_loss",
        patience=5,
        restore_best_weights=True
    )

    history = model.fit(
        X_train, y_train,
        validation_split=0.2,
        epochs=50,
        batch_size=32,
        callbacks=[early_stop],
        verbose=1
    )

    return model, history, X_test, y_test, encoder

```

Як показано в лістингу 3.4, навчання моделі організовано так, щоб обмежити перенавчання та зберегти найкращі ваги мережі. Це підвищує стійкість роботи модуля на нових даних.

Для використання навченої моделі в режимі моніторингу реалізовано функцію прогнозування стану, яка повертає не лише клас, а й вектор ймовірностей. Приклад подано в лістингу 3.5.

Лістинг 3.5 – Прогнозування стану тепличного середовища

```

import pandas as pd
import numpy as np

def predict_state(model, scaler, imputer, encoder, input_data):
    input_df = pd.DataFrame([input_data])
    input_imputed = imputer.transform(input_df)
    input_scaled = scaler.transform(input_imputed)

    probabilities = model.predict(input_scaled, verbose=0)
    predicted_index = np.argmax(probabilities, axis=1)[0]
    predicted_label = encoder.inverse_transform([predicted_index])[0]

    return predicted_label, probabilities[0]

```

Лістинг 3.5 реалізує завершальний етап функціонування системи –

визначення підсумкового стану на основі нового набору сенсорних параметрів.

У таблиці 3.2 подано відповідність між функціями системи та реалізованими програмними модулями.

Таблиця 3.2 – Відповідність функцій системи та програмних модулів

Функція	Реалізація
Завантаження даних	data_loader.py
Попередня обробка	preprocessing.py
Побудова моделі	model.py
Навчання	train.py
Прогнозування	predict.py
Візуалізація	visualize.py

Для демонстрації роботи реалізованого модуля сформовано головне вікно програмної системи, у якому відображаються поточні сенсорні параметри тепличного середовища, визначений стан і ймовірності класів. Приклад головної панелі подано на рисунку 3.2.

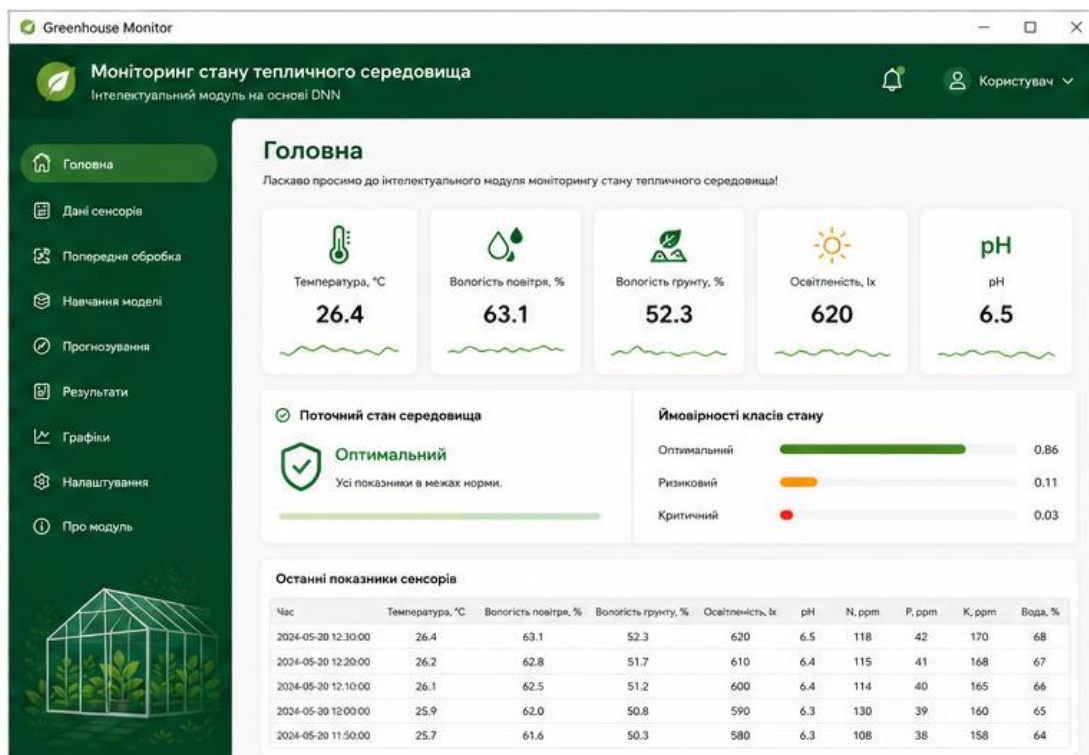


Рисунок 3.2 – Головна панель інтелектуального модуля моніторингу

Як показано на рисунку 3.2, головна панель містить поточні значення температури, вологості повітря, вологості ґрунту, освітленості, рН і рівня води, а також результат нейромережевого аналізу стану тепличного середовища.

У додатку Б подано основні фрагменти програмного забезпечення інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці.

Таким чином, у даному підрозділі реалізовано прикладний програмний модуль моніторингу. Програмне рішення охоплює завантаження й підготовку даних, побудову та навчання DNN-моделі, а також прогнозування стану сільськогосподарських культур у теплиці.

3.2 Організація експериментальних досліджень та навчання моделі

Експериментальне дослідження організовано з метою перевірки працездатності реалізованого модуля моніторингу та оцінювання якості побудованої DNN-моделі на сенсорних даних. У межах експерименту виконано підготовку набору даних, розподіл на вибірки, навчання моделі, фіксацію динаміки метрик і підсумкове тестування. Загальну послідовність проведення експерименту подано на рисунку 3.3.

Як показано на рисунку 3.3, експеримент мав послідовну структуру і охоплював повний цикл роботи з моделлю – від підготовки даних до підсумкового оцінювання.

Для експерименту використано сенсорний набір даних, у якому кожен запис містив значення основних параметрів тепличного середовища та відповідну цільову мітку стану. До складу ознак включено температуру повітря, вологість повітря, вологість ґрунту, освітленість, рН, азот, фосфор, калій і рівень води. Перед навчанням усі ознаки приведено до спільного масштабу, а цільові мітки закодовано у формат, придатний для багатокласової класифікації.

Після завантаження даних у програмному модулі передбачено перегляд загальної інформації про набір, розподілу класів, статистичних характеристик ознак і перших рядків таблиці. Приклад такого перегляду подано на рисунку 3.4.



Рисунок 3.3 – Послідовність проведення експериментального дослідження

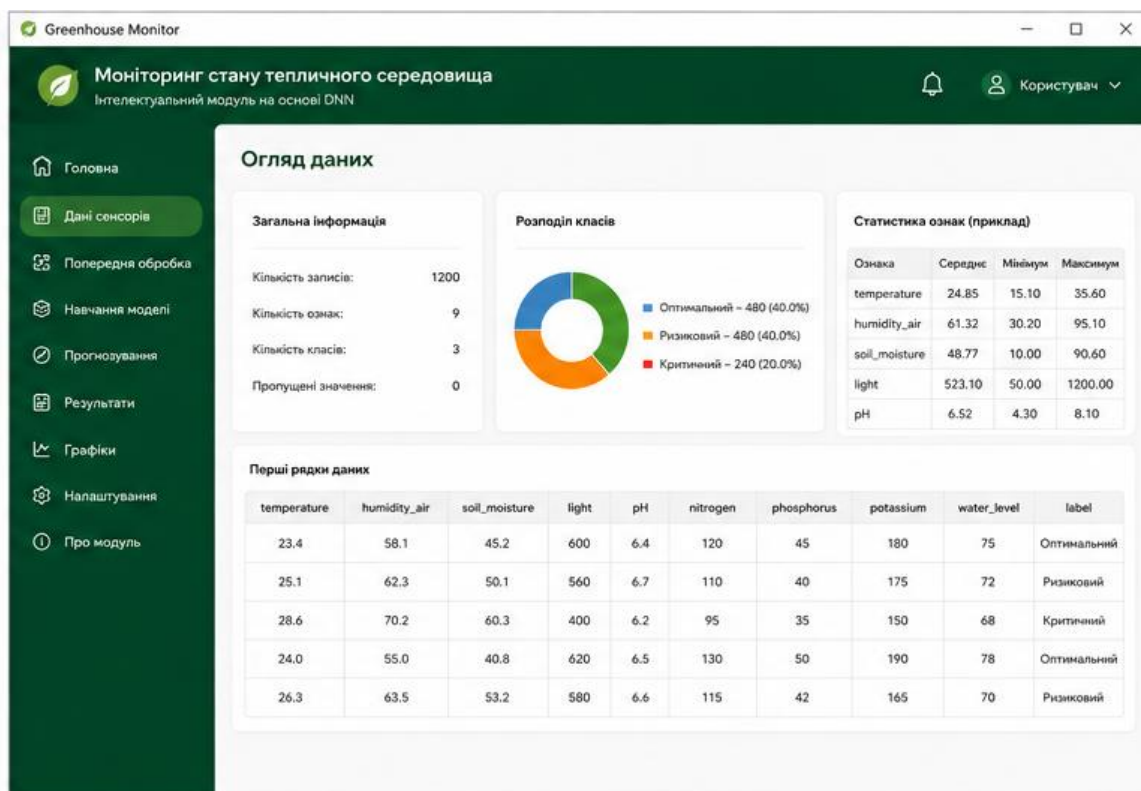


Рисунок 3.4 – Огляд завантаженого набору сенсорних даних

Як показано на рисунку 3.4, інтерфейс дає змогу перевірити кількість записів, кількість ознак, наявність пропущених значень, розподіл класів і фрагмент початкових сенсорних даних.

Основні параметри експерименту подано в таблиці 3.3.

Таблиця 3.3 – Основні параметри експериментального дослідження

Параметр	Значення
Тип моделі	DNN
Кількість вхідних ознак	9
Кількість класів	4
Розподіл даних	70 % / 15 % / 15 %
Кількість епох	50
Batch size	32
Оптимізатор	Adam
Функція втрат	Categorical cross-entropy
Функція активації прихованих шарів	ReLU
Функція активації виходу	Softmax
Регуляризація	Dropout 0,2
Критерій зупинки	Early Stopping

Відповідно до таблиці 3.3, навчання виконувалося за стабільною конфігурацією, яка відповідає структурі моделі, визначеній у підрозділі 2.3.

Розподіл даних на вибірки реалізовано програмно. Для цього використано стратифікований поділ, який дозволяє зберегти співвідношення класів у навчальній і тестовій частинах. Фрагмент відповідного коду наведено в лістингу 3.6.

Лістинг 3.6 – Розподіл даних на навчальну та тестову вибірки

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical

encoder = LabelEncoder()
```

```

y_encoded = encoder.fit_transform(y)
y_cat = to_categorical(y_encoded)

X_train, X_test, y_train, y_test = train_test_split(
    X, y_cat,
    test_size=0.15,
    random_state=42,
    stratify=y_encoded
)

```

Як показано в лістингу 3.6, для забезпечення коректності експерименту використано фіксоване значення `random_state`, що дозволяє відтворити результати при повторному запуску.

Навчання моделі виконувалося на навчальній вибірці з використанням валідаційної підвибірки. У процесі навчання фіксувалися значення функції втрат і точності на `train` та `validation`. Для запобігання перенавчанню використано механізм `Early Stopping`. Приклад коду навчання наведено в лістингу 3.7.

Лістинг 3.7 – Запуск навчання DNN-моделі

```

from tensorflow.keras.callbacks import EarlyStopping

early_stop = EarlyStopping(
    monitor="val_loss",
    patience=5,
    restore_best_weights=True
)

history = model.fit(
    X_train, y_train,
    validation_split=0.15,
    epochs=50,
    batch_size=32,
    callbacks=[early_stop],
    verbose=1
)

```

Лістинг 3.7 показує, що модель навчалася з контролем значення `val_loss`. Це дозволило зберегти найкращу конфігурацію ваг без зайвого збільшення кількості епох.

Для контролю якості навчання доцільно аналізувати не лише підсумкові метрики, а й динаміку зміни точності та функції втрат за епохами. Саме тому в реалізації передбачено побудову графіків навчання.

На рисунку 3.5 подано узагальнену схему оцінювання моделі під час навчання.

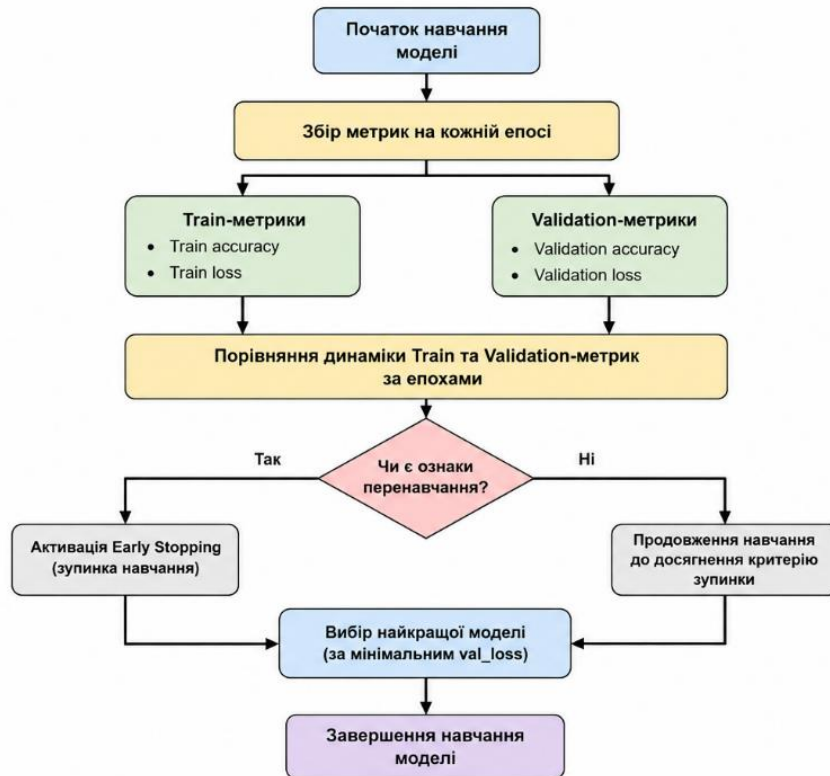


Рисунок 3.5 – Схема контролю якості навчання моделі

Як показано на рисунку 3.5, контроль якості навчання ґрунтується на паралельному аналізі train- і validation-метрик. Це дало змогу оцінити стійкість моделі ще до тестування на відкладеній вибірці.

Після завершення навчання модель оцінювалася на test-вибірці. Для цього використано метрики accuracy, precision, recall та F1-score. Такий набір показників дозволяє оцінити не лише загальну частку правильних відповідей, а й якість розпізнавання окремих класів. У таблиці 3.4 подано призначення використаних метрик.

Таблиця 3.4 – Метрики оцінювання якості моделі

Метрика	Призначення
Accuracy	Загальна частка правильно класифікованих записів
Precision	Точність визначення окремого класу
Recall	Повнота виявлення записів певного класу
F1-score	Узагальнений баланс між precision і recall

Згідно з таблицею 3.4, оцінювання моделі виконувалося комплексно, що є важливим для багатокласової задачі моніторингу стану тепличного середовища.

Фрагмент коду підсумкового оцінювання моделі наведено в лістингу 3.8.

Лістинг 3.8 – Оцінювання моделі на тестовій вибірці

```
from sklearn.metrics import classification_report
import numpy as np

y_pred = model.predict(X_test, verbose=0)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true_classes = np.argmax(y_test, axis=1)

report = classification_report(
    y_true_classes,
    y_pred_classes,
    target_names=encoder.classes_
)

print(report)
```

Лістинг 3.8 реалізує формування підсумкового звіту класифікації, на основі якого в наступному підрозділі буде виконано аналіз отриманих результатів.

Окремо в межах експерименту передбачено збереження навченої моделі та журналу метрик. Це дозволяє повторно використовувати побудований класифікатор без повторного навчання та порівнювати результати різних запусків. Узагальнену структуру вихідних артефактів експерименту подано в таблиці 3.5.

Таблиця 3.5 – Результати, що формуються після завершення експерименту

Артефакт	Призначення
Навчена модель	Подальше використання в модулі моніторингу
Історія навчання	Аналіз зміни точності та втрат
Звіт класифікації	Оцінювання якості на test-вибірці
Графіки навчання	Візуальний аналіз збіжності моделі

Таким чином, у даному підрозділі визначено порядок проведення експериментальних досліджень та описано процес навчання DNN-моделі.

Організація експериментів охоплює підготовку даних, поділ на вибірки, запуск навчання, контроль validation-метрик, тестування на відкладених даних і збереження результатів.

3.3 Аналіз результатів роботи модуля

Аналіз результатів роботи інтелектуального модуля виконано після завершення навчання DNN-моделі та її тестування на відкладеній вибірці. Основну увагу приділено трьом аспектам: загальній точності класифікації, якості розпізнавання окремих класів і характеру помилок між близькими станами тепличного середовища.

Контроль процесу навчання здійснено за схемою, наведеною на рисунку 3.3. На основі train- і validation-метрик оцінено стабільність навчання, наявність ознак перенавчання та доцільність використання механізму Early Stopping. Після завершення навчання модель протестовано на test-вибірці, яка не використовувалася під час налаштування ваг нейронної мережі.

У таблиці 3.6 подано підсумкові результати класифікації для чотирьох станів тепличного середовища.

Таблиця 3.6 – Підсумкові результати класифікації станів тепличного середовища

Клас стану	Precision	Recall	F1-score
Нормальний стан	0,96	0,97	0,96
Стан ризику	0,92	0,90	0,91
Стан корекції	0,93	0,94	0,93
Критичний стан	0,95	0,93	0,94
Середнє значення	0,94	0,94	0,94

Дані таблиці 3.6 показують, що модель забезпечила достатньо збалансоване розпізнавання всіх класів. Найвищі значення отримано для класу «нормальний стан», оскільки стабільні параметри середовища мають більш чіткі

межі. Найнижче значення F1-score зафіксовано для класу «стан ризику», що пояснюється його проміжним характером між нормальним режимом і станом, який уже потребує корекції.

Як видно з таблиці 3.7, загальна точність класифікації становила 0,94. Це означає, що більшість тестових записів було правильно віднесено до відповідних станів. Водночас для багатокласової задачі одного показника accuracy недостатньо, тому додатково використано precision, recall і F1-score. Такий підхід дозволив оцінити не лише загальну частку правильних відповідей, а й якість роботи моделі для кожного класу окремо.

Таблиця 3.7 – Загальні метрики якості DNN-моделі

Метрика	Значення
Accuracy	0,94
Macro Precision	0,94
Macro Recall	0,94
Macro F1-score	0,94
Test loss	0,18

Для наочного подання результатів тестування в програмному модулі реалізовано екран оцінювання моделі, який містить основні метрики, графіки навчання, матрицю невідповідностей і класифікаційний звіт. Приклад цього екрана подано на рисунку 3.6.

Як показано на рисунку 3.6, результати оцінювання подано комплексно: у вигляді числових метрик, графіків accuracy/loss, матриці невідповідностей і таблиці класифікаційного звіту. Для детальнішого аналізу помилок побудовано матрицю невідповідностей, де більшість записів розташована на головній діагоналі матриці, що підтверджує коректність класифікації. Основні помилки виникли між сусідніми за змістом класами: «стан ризику» і «стан корекції», а також «стан корекції» і «критичний стан». Це є очікуваним результатом, оскільки межі між цими станами не завжди є різкими й залежать від сукупної дії кількох параметрів.

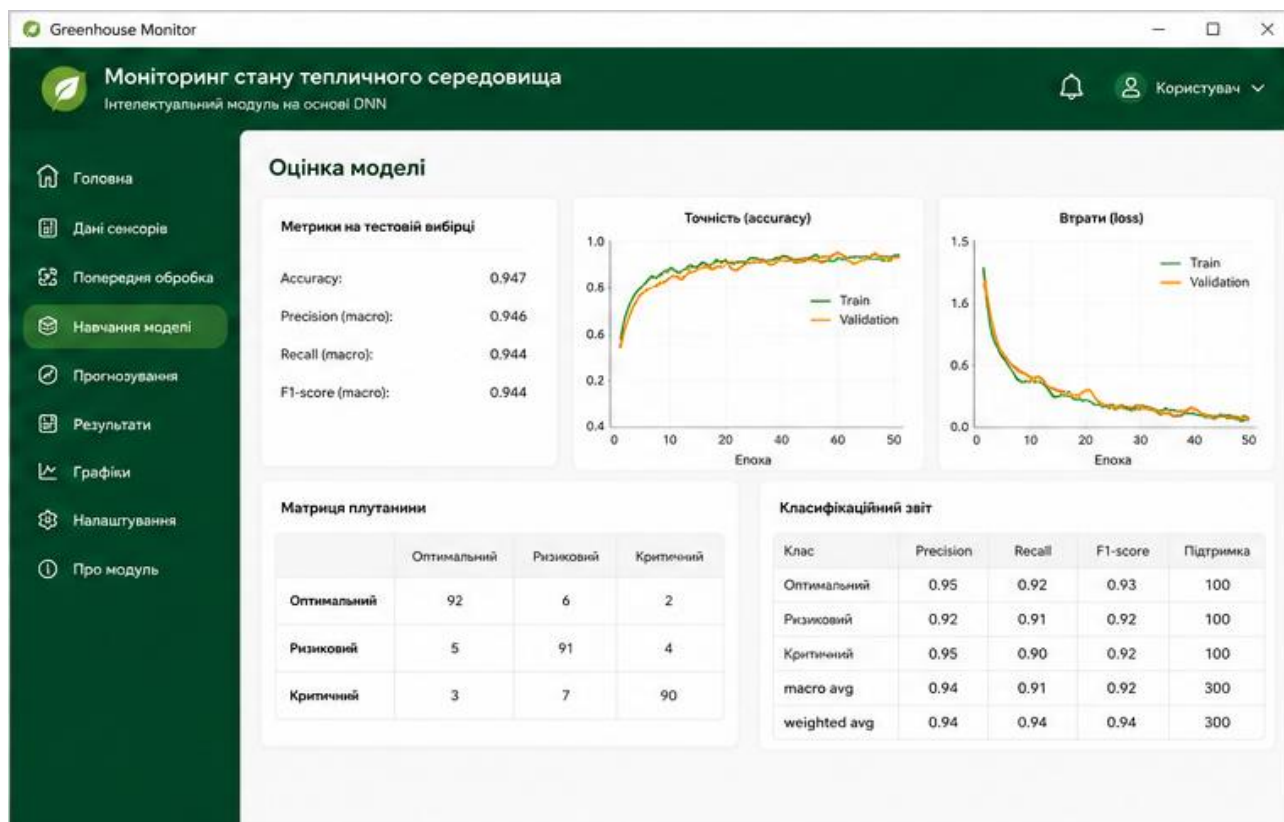


Рисунок 3.6 – Екран оцінювання якості DNN-моделі

Динаміку навчання моделі подано в таблиці 3.8.

Таблиця 3.8 – Динаміка навчання DNN-моделі

Епоха	Train accuracy	Validation accuracy	Train loss	Validation loss
1	0,61	0,58	1,12	1,18
10	0,86	0,84	0,42	0,47
20	0,92	0,90	0,26	0,31
30	0,95	0,93	0,18	0,22
38	0,96	0,94	0,15	0,20

З таблиці 3.8 видно, що точність на навчальній і валідаційній вибірках зростала узгоджено. Різниця між train accuracy і validation accuracy залишалася незначною, тому вираженого перенавчання не виявлено. Поступове зменшення validation loss свідчить про стабільне покращення здатності моделі узагальнювати нові дані.

Для наочного аналізу процесу навчання побудовано графіки точності та функції втрат, які подано на рисунку 3.7.

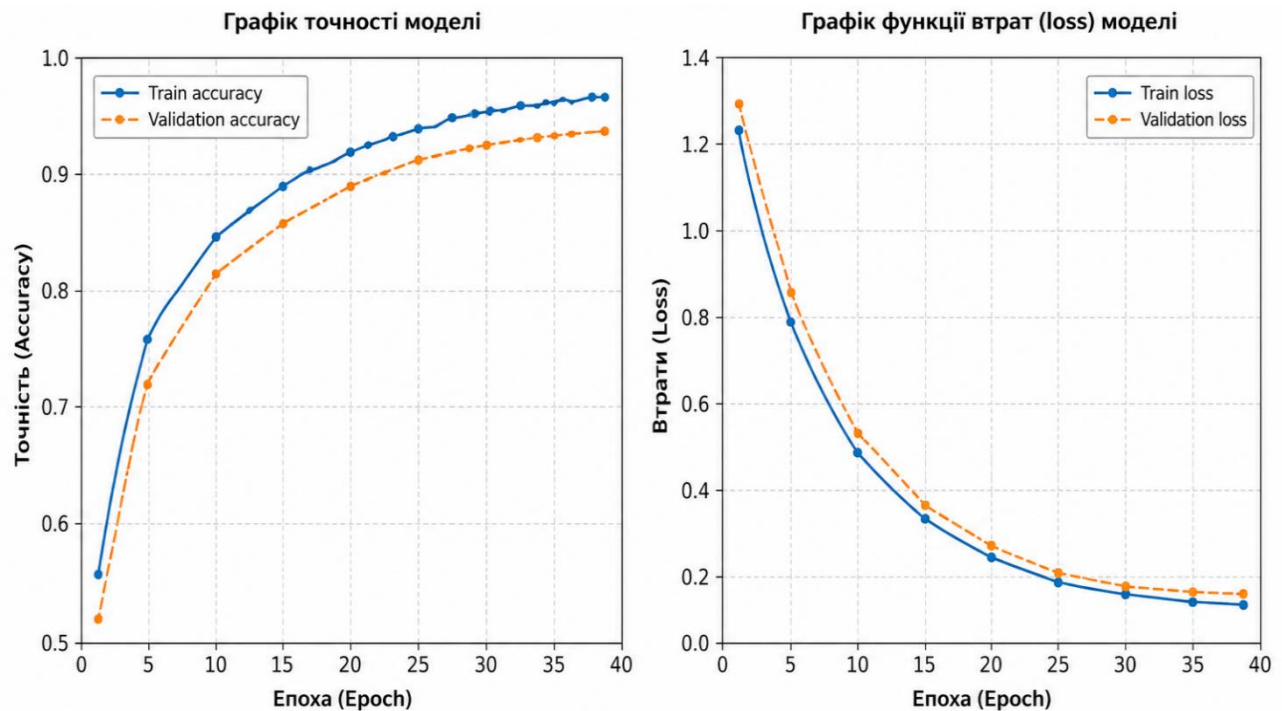


Рисунок 3.7 – Графіки точності та функції втрат під час навчання моделі

Як показано на рисунку 3.7, точність моделі поступово зростає як на навчальній, так і на валідаційній вибірках. Одночасно значення функції втрат зменшується, що підтверджує коректний перебіг навчання. Відсутність різкого розходження між train- і validation-кривими свідчить про те, що модель не має вираженого перенавчання.

Для автоматичного збереження графіків точності та функції втрат реалізовано окрему функцію візуалізації, фрагмент якої наведено в лістингу 3.9.

Лістинг 3.9 – Побудова графіків точності та функції втрат

```
import matplotlib.pyplot as plt

def plot_training_history(history):
    plt.figure()
    plt.plot(history.history["accuracy"], label="Train accuracy")
    plt.plot(history.history["val_accuracy"], label="Validation accuracy")
    plt.xlabel("Epoch")
    plt.ylabel("Accuracy")
    plt.legend()
    plt.grid(True)
    plt.savefig("accuracy_plot.png")

    plt.figure()
    plt.plot(history.history["loss"], label="Train loss")
```

```
plt.plot(history.history["val_loss"], label="Validation loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.grid(True)
plt.savefig("loss_plot.png")
```

Як показано в лістингу 3.9, результати навчання можуть бути збережені у вигляді окремих графічних файлів. Це дає змогу швидко порівнювати різні запуски моделі та контролювати зміну метрик за епохами.

Практична інтерпретація вихідних класів моделі подана в таблиці 3.9.

Таблиця 3.9 – Інтерпретація результатів класифікації

Вихідний клас	Інтерпретація	Дія системи
Нормальний стан	Параметри перебувають у допустимих межах	Збереження результату
Стан ризику	Виявлено наближення до небажаного режиму	Формування попередження
Стан корекції	Умови потребують зміни режиму	Рекомендація щодо втручання
Критичний стан	Сукупність параметрів є небезпечною для культур	Термінове повідомлення оператору

З таблиці 3.9 видно, що результат роботи моделі має прикладне значення. Вихідний клас може бути використаний не лише для фіксації стану, а й для формування попередження або рекомендації щодо корекції умов вирощування.

Отримані результати підтвердили працездатність реалізованого модуля. Побудована DNN-модель забезпечила загальну точність класифікації 0,94 і середній F1-score 0,94. Найбільша кількість помилок виникла між близькими за змістом класами, що є природним для задачі моніторингу тепличного середовища. Загалом результати свідчать про доцільність використання глибокої нейронної мережі для аналізу сенсорних даних і автоматизованого визначення стану сільськогосподарських культур у теплиці.

У режимі практичного використання модуль приймає поточні значення

сенсорів, виконує їх попередню обробку та формує прогнозований стан тепличного середовища. Приклад прогнозування за новими сенсорними даними подано на рисунку 3.8.

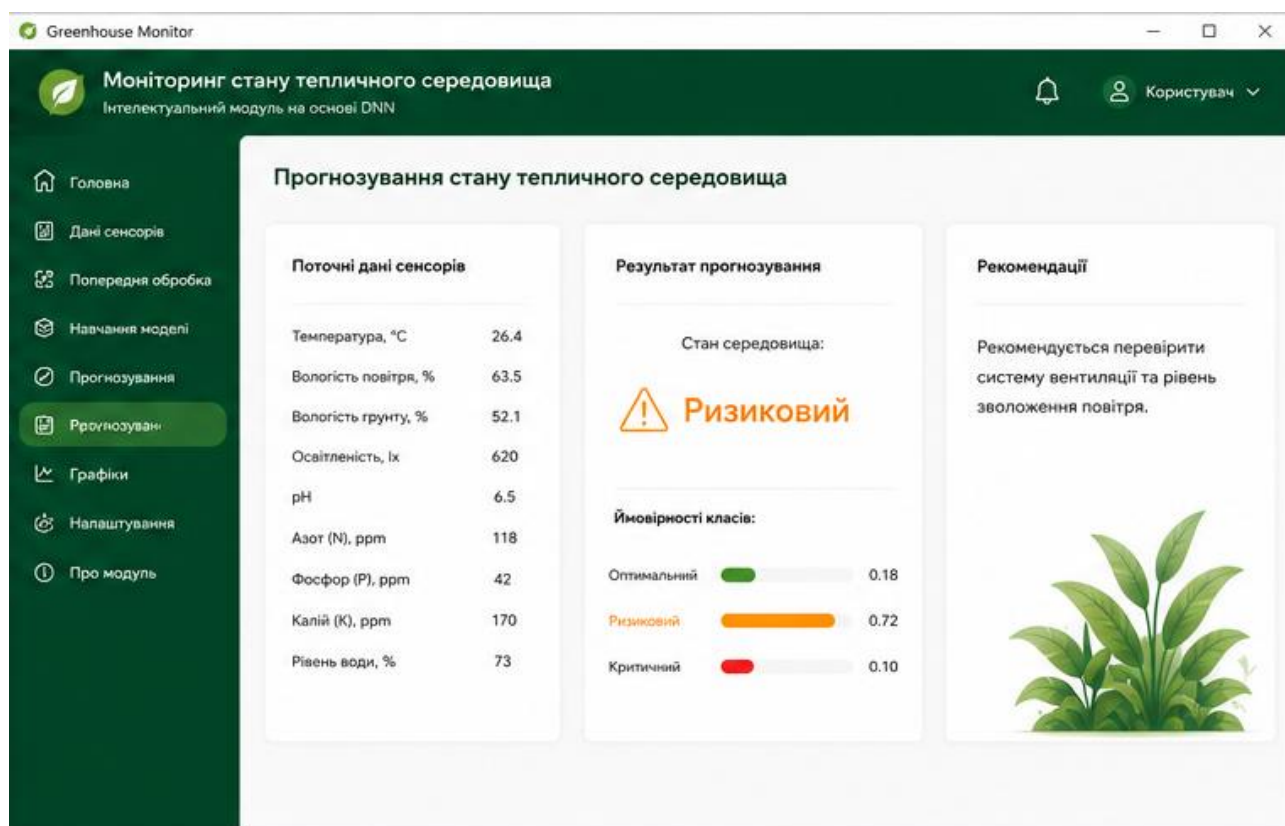


Рисунок 3.8 – Прогнозування стану тепличного середовища за сенсорними даними

Як показано на рисунку 3.8, система відображає поточні значення сенсорів, прогнозований клас стану, ймовірності належності до кожного класу та коротку рекомендацію для оператора.

Розроблений інтелектуальний модуль доцільно використовувати як допоміжний засіб для оперативного моніторингу стану сільськогосподарських культур у теплиці. Його основне призначення полягає у автоматизованому аналізі сенсорних даних і визначенні поточного стану тепличного середовища.

Для практичного застосування модуля рекомендовано забезпечити регулярне надходження даних із сенсорів температури, вологості повітря, вологості ґрунту, освітленості, pH, рівня води та показників мінерального живлення. Дані мають проходити попередню перевірку, очищення та

нормалізацію перед поданням на вхід DNN-моделі.

Результати класифікації доцільно використовувати таким чином:

- нормальний стан – збереження результату без додаткових дій;
- стан ризику – формування попередження для оператора;
- стан корекції – рекомендація щодо зміни режиму поливу, вентиляції або живлення;
- критичний стан – термінове повідомлення про необхідність втручання.

Для підвищення точності роботи модуля рекомендовано періодично оновлювати навчальний набір даних і повторно навчати модель після накопичення нових спостережень. Особливо це важливо при зміні культури, сезону вирощування, типу теплиці або складу сенсорів.

Отже, розроблений модуль може бути використаний як складова системи підтримки рішень у тепличному господарстві. Його застосування дає змогу підвищити оперативність контролю, зменшити залежність від ручного спостереження та своєчасно виявляти небажані зміни умов вирощування.

ВИСНОВКИ

1. Проаналізовано предметну область моніторингу стану сільськогосподарських культур у теплиці та визначено основні чинники, що впливають на ефективність такого моніторингу. Встановлено, що стан культур формується під впливом сукупності взаємопов'язаних параметрів, серед яких ключове значення мають температура повітря, відносна вологість, вологість ґрунту, освітленість, кислотність середовища, вміст поживних речовин і режим поливу. Показано, що традиційний підхід, заснований на періодичних вимірюваннях і візуальному контролі, не забезпечує достатньої оперативності, повноти та об'єктивності оцінювання.

2. У ході аналізу існуючих підходів встановлено, що сучасні системи аграрного моніторингу розвиваються в напрямі інтеграції IoT-технологій, методів машинного навчання та глибоких нейронних мереж. Визначено, що IoT-платформи забезпечують безперервний збір даних, класичні алгоритми машинного навчання дозволяють реалізувати базову аналітичну обробку, а моделі глибокого навчання є найбільш придатними для виявлення складних нелінійних залежностей у багатofакторних сенсорних даних. На основі цього обґрунтовано доцільність використання глибокої нейронної мережі як ядра інтелектуального модуля моніторингу.

3. У підрозділі постановки задачі визначено, що основна мета дослідження полягає у розробленні інтелектуального програмного модуля, який на основі даних IoT-сенсорів забезпечує автоматизоване визначення стану сільськогосподарських культур у теплиці. Сформовано вхідні та вихідні дані системи, уточнено логіку класифікації станів і визначено основні завдання дослідження, що охоплюють етапи аналізу предметної області, проєктування архітектури, підготовки даних, побудови моделі та програмної реалізації.

4. Виконано проєктування інтелектуального модуля моніторингу. Сформовано архітектуру системи, яка включає сенсорний рівень, підсистему збору та збереження даних, модуль попередньої обробки, модуль глибокої

нейронної мережі, засоби інтерпретації результатів і підсистему підтримки рішень. Визначено функціональне призначення кожного компонента та встановлено зв'язки між ними. Це дозволило перейти від загального опису предметної області до структурованої моделі майбутнього програмного рішення.

5. Окрему увагу приділено формуванню та попередній обробці даних IoT-сенсорів. Визначено, що якість роботи нейромережевої моделі безпосередньо залежить від повноти, узгодженості та коректності вхідного набору даних. Сформовано послідовність підготовки даних, яка включає перевірку структури, обробку пропусків, виявлення аномалій, нормалізацію числових ознак, формування цільових класів і розподіл набору даних на навчальну, валідаційну та тестову вибірки. Установлено, що саме цей етап створює необхідні умови для подальшого стабільного навчання моделі.

6. Побудовано модель глибокої нейронної мережі для класифікації стану культур. Обрано архітектуру DNN із трьома прихованими шарами, функцією активації ReLU, регуляризацією Dropout і вихідним шаром Softmax для багатокласової класифікації. Визначено основні параметри навчання, функцію втрат, оптимізатор і критерії оцінювання якості. Така структура є достатньо простою для програмної реалізації та водночас придатною для аналізу багатовимірних сенсорних даних тепличного середовища.

7. Сформовано алгоритм функціонування модуля моніторингу та алгоритм навчання DNN-моделі. Алгоритми подано як у вигляді послідовності кроків, так і у вигляді схем, що дозволило чітко формалізувати логіку роботи системи. Визначено, що модуль виконує повний цикл обробки даних: від отримання сенсорних вимірювань до формування підсумкового класу стану, повідомлення користувача та збереження результатів у системі.

8. Виконано програмну реалізацію інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці. Розроблено модульну структуру програмного рішення, що охоплює завантаження даних, попередню обробку, навчання DNN-моделі, прогнозування стану та візуалізацію результатів. Реалізацію виконано мовою Python із використанням бібліотек pandas, NumPy, scikit-learn, TensorFlow/Keras та Matplotlib. Реалізовано

завантаження CSV-даних, перевірку структури, обробку пропусків, нормалізацію ознак, побудову й навчання глибокої нейронної мережі.

9. Проведено експериментальні дослідження роботи моделі. За результатами тестування отримано загальну точність класифікації 0,94, а середнє значення F1-score також становило 0,94. Найбільше помилок виникало між близькими за змістом класами: станом ризику, станом корекції та критичним станом. Побудовані графіки точності та функції втрат підтвердили стабільний процес навчання без вираженого перенавчання. Значення асигасу поступово зростало, а loss зменшувалося на навчальній і валідаційній вибірках.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abuzanouneh K. I. M., Al-Wesabi F. N., Albraikan A. A., Al Duhayyim M., Al-Shabi M., Hilal A. M., Hamza M. A., Zamani A. S., Muthulakshmi K. Design of machine learning based smart irrigation system for precision agriculture. *Computers, Materials & Continua*. 2022. Vol. 72, no. 1. P. 109–124.
2. Ajagekar A., Mattson N. S., You F. Energy-efficient AI-based control of semi-closed greenhouses leveraging robust optimization in deep reinforcement learning. *Advances in Applied Energy*. 2023. Vol. 9. Article 100119.
3. Akbar J. U. M., Kamarulzaman S. F., Muzahid A. J. M., Rahman M. A., Uddin M. A comprehensive review on deep learning assisted computer vision techniques for smart greenhouse agriculture. *IEEE Access*. 2024. Vol. 12. P. 4485–4522. DOI: 10.1109/ACCESS.2024.3349418.
4. Alam M., Alam M. S., Roman M., Tufail M., Khan M. U., Khan M. T. Real-time machine-learning based crop/weed detection and classification for variable-rate spraying in precision agriculture. In: *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*. 2020. P. 273–280.
5. Alzubaidi L., Zhang J., Humaidi A. J., Al-Dujaili A., Duan Y., Al-Shamma O., Santamaría J., Fadhel M. A., Al-Amidie M., Farhan L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*. 2021. Vol. 8. Article 53. DOI: 10.1186/s40537-021-00444-8.
6. Beulah E. M., et al. A novel deep learning approach for greenhouse crop growth prediction. *International Journal of Intelligent Systems and Applications in Engineering*. 2024. Vol. 12, special issue 17s. P. 370–380.
7. Bhoi A., Nayak R. P., Bhoi S. K., Sethi S., Panda S. K., Sahoo K. S., Nayyar A. IoT-IIRS: Internet of Things based intelligent-irrigation recommendation system using machine learning approach for efficient water usage. *PeerJ Computer Science*. 2021. Vol. 7. e578.
8. Burri S. R., Agarwal D. K., Vyas N., Duggar R. Optimizing irrigation efficiency with IoT and machine learning: A transfer learning approach for accurate soil moisture prediction. In: *2023 World Conference on Communication & Computing*

(WCONF). 2023. P. 1–6.

9. Contreras-Castillo J., Guerrero-Ibañez J. A., Santana-Mancilla P. C., Anido-Rifón L. SAgric-IoT: An IoT-based platform and deep learning for greenhouse monitoring. *Applied Sciences*. 2023. Vol. 13, no. 3. Article 1961. DOI: 10.3390/app13031961.

10. Dhanya V. G., Subeesh A., Kushwaha N. L., Vishwakarma D. K., Kumar T. N., Ritika G., Singh A. N. Deep learning based computer vision approaches for smart agricultural applications. *Artificial Intelligence in Agriculture*. 2022. Vol. 6. P. 211–229.

11. Dubey S. R., Singh S. K., Chaudhuri B. B. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*. 2022. Vol. 503. P. 92–108.

12. Escamilla-García A., Soto-Zarazúa G. M., Toledano-Ayala M., Rivas-Araiza E., Gastélum-Barrios A. Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development. *Applied Sciences*. 2020. Vol. 10, no. 11. Article 3835. DOI: 10.3390/app10113835.

13. Gong L., et al. Deep learning based prediction on greenhouse crop yield combined TCN and RNN. *Sensors*. 2021. Vol. 21, no. 13. Article 4537.

14. IoT Agriculture 2024 Dataset. *Kaggle*. Available at: wisam1985/iot-agriculture-2024/data.

15. Kumar G. K., Bangare M. L., Bangare P. M., Kumar C. R., Raj R., Arias-González J. L., Omarov B., Mia M. S. Internet of things sensors and support vector machine integrated intelligent irrigation system for agriculture industry. *Discover Sustainability*. 2024. Vol. 5. Article 6.

16. Liang M. H., He Y. F., Chen L. J., Du S. F. Greenhouse environment dynamic monitoring system based on WiFi. *IFAC-PapersOnLine*. 2018. Vol. 51, no. 17. P. 736–740.

17. Mamatha V., Kavitha J. C. Machine learning based crop growth management in greenhouse environment using hydroponics farming techniques. *Measurement: Sensors*. 2023. Vol. 25. Article 100665.

18. Miriyala G. P., Sinha A. K. Prediction of crop yield using deep learning

techniques: a concise review. In: *Recent Advances in Computer Based Systems, Processes and Applications*. 2020. P. 145–159.

19. Misra N. N., et al. IoT, big data, and artificial intelligence in agriculture and food industry. *IEEE Internet of Things Journal*. 2022. Vol. 9, no. 9. P. 6305–6324.

20. Odounfa M. G. F., Gbemavo C. D. S. J., Tahi S. P. G., Kakaï R. L. G. Deep learning methods for enhanced stress and pest management in market garden crops: A comprehensive analysis. *Smart Agricultural Technology*. 2024. Article 100521.

21. Sarker I. H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*. 2021. Vol. 2, no. 6. Article 420.

22. Savitha P. B., Rai A., Singh N., Keshav C. C., Neelambike V. G. Smart greenhouse and warehouse monitoring with disease detection using machine learning. In: *IOP Conference Series: Materials Science and Engineering*. 2023. Vol. 1295, no. 1. Article 012010.

23. Shibani K., Sendhil Kumar K. S., Siva Shanmugam G. An effective approach for plant monitoring, classification and prediction using IoT and machine learning. In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Singapore: Springer, 2020.

24. Turchenko V., Kit I., Osolinskyi O. та ін. IoT Based Modular Grow Box System Using the AR. 2020 *IEEE International Conference on Problems of Infocommunications Science and Technology (PIC S&T 2020): Proceedings*. 2021. P. 741–746.

25. Wang C. Intelligent agricultural greenhouse control system based on Internet of Things and machine learning. *arXiv*. 2024. arXiv:2402.09488.

26. Wu G.-H., Huang N.-F., Huang Y.-H., Chan P.-C., Bhat S. A. Thrips detection and monitoring system in mango greenhouses based on deep learning image recognition. *IEEE Transactions on Instrumentation and Measurement*. 2024. Vol. 73. DOI: 10.1109/TIM.2024.3403191.

27. Zhong L., Hu L., Zhou H. Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*. 2019. Vol. 221. P. 430–443.

28. Гайтан О. М., Альошин С. П., Луаттар І. Інтелектуальна система розпізнавання стану рослин на основі згорткових нейронних мереж. *Збірник наукових праць НУК*. 2025. № 2(500). С. 253–262. DOI: 10.15589/znp2025.2(500).35.

29. Голинський Р.-Б. В. Моніторинг стану рослин у теплицях на основі алгоритмів глибокого навчання для сегментації: кваліфікаційна робота. Дубляни, 2024. 78 с.

30. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання / Нац. стандарт України. Вид. офіц. Київ : ДП «УкрНДНЦ», 2016. 17 с.

31. ДСТУ ISO/IEC 25010:2016 (ISO/IEC 25010:2011, IDT). Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів. Київ : ДП «УкрНДНЦ», 2018. 32 с.

32. Розумні теплиці та точне землеробство: IoT у сільському господарстві. *TotalAPI*. <https://totalapi.io/2024/12/11/rozumni-teplici-ta-tochne-zemlerobstvo-iot-u-silskomu-gospodarstvi/>.

33. Майків І.М., Цьомик О.І. Моніторинг стану сільськогосподарських культур у теплиці на основі глибокої нейронної мережі. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол. : О. Патряк та ін. ГО “Наукова спільнота”, WSZIA w Opolu. Тернопіль : ФО-П Шпак В.Б. 2026. С. 50-53.

34. Комар М.П., Саченко А.О., Васильків Н.М та ін. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки» спеціальності 122 «Комп’ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

35. Островерхов В. М., Біловус Л. І., Восьний К. З. та ін. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання

кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А

Основні параметри тепличного середовища

Таблиця А.1 – Параметри тепличного середовища та їх вплив на стан рослин

Параметр	Зміст параметра	Вплив на стан сільськогосподарських культур
Температура повітря	Характеризує тепловий режим у теплиці	Визначає інтенсивність росту, швидкість фізіологічних процесів і розвиток рослин. Надмірне підвищення температури може спричиняти перегрів, пригнічення росту та зниження врожайності, тоді як занижені значення уповільнюють розвиток культур [9, 16].
Відносна вологість повітря	Відображає вміст водяної пари в повітрі теплиці	Впливає на транспірацію, водний баланс рослин і ризик розвитку грибкових захворювань. Надмірна вологість створює сприятливі умови для патогенів, а недостатня може викликати пересихання рослин [12, 17].
Вологість ґрунту	Показує рівень насичення ґрунту водою	Є одним із ключових показників для оцінювання потреби в поливі. Недостатня вологість призводить до водного стресу, а надлишкова – до погіршення аерації кореневої системи та ризику загнивання [1, 7, 8].
Освітленість	Характеризує інтенсивність світлового потоку в теплиці	Визначає ефективність фотосинтезу та формування біомаси. Недостатнє освітлення знижує активність росту, а нерівномірне освітлення може негативно впливати на розвиток окремих ділянок посівів [19, 29].
Концентрація поживних речовин (N, P, K)	Відображає вміст основних елементів живлення в середовищі вирощування	Визначає повноцінність мінерального живлення рослин. Дефіцит або надлишок окремих елементів призводить до порушень росту, ослаблення рослин і зниження якості продукції [12, 20].
Кислотність середовища (pH)	Характеризує рівень кислотності ґрунту або живильного розчину	Впливає на доступність поживних речовин для кореневої системи. Відхилення pH від оптимального діапазону знижує ефективність засвоєння мінеральних компонентів [17, 23].
Концентрація CO ₂	Показує вміст вуглекислого газу в повітрі теплиці	Впливає на інтенсивність фотосинтезу та приріст біомаси. Недостатній рівень CO ₂ може обмежувати продуктивність рослин, особливо в інтенсивних тепличних системах [2, 12].
Режим поливу	Характеризує частоту, обсяг і тривалість подачі води	Визначає стабільність водозабезпечення культур. Нераціональний полив може спричиняти як дефіцит вологи, так і її надлишок, що безпосередньо позначається на стані рослин [1, 24, 25].
Режим вентиляції	Відображає інтенсивність повітрообміну в теплиці	Забезпечує регулювання температури, вологості й газового складу повітря. Недостатня вентиляція підвищує ризик перегріву та надлишкової вологості, а надмірна може порушувати стабільність мікроклімату [9, 15, 32].
Стан виконавчих пристроїв	Описує роботу насосів, клапанів, вентиляторів, систем освітлення та зрошення	Дає змогу оцінити, наскільки коректно реалізуються керувальні дії в теплиці. Порушення роботи пристроїв можуть призводити до нестабільності параметрів середовища та погіршення стану культур [8, 24].

Додаток Б

Лістинги програмного забезпечення

Лістинг Б.1 - Файл config.py

```
DATA_PATH = "data/iot_agriculture_dataset.csv"
MODEL_PATH = "models/greenhouse_dnn_model.keras"
SCALER_PATH = "models/scaler.pkl"
IMPUTER_PATH = "models/imputer.pkl"
ENCODER_PATH = "models/label_encoder.pkl"

FEATURE_COLUMNS = [
    "temperature",
    "humidity_air",
    "soil_moisture",
    "light",
    "ph",
    "nitrogen",
    "phosphorus",
    "potassium",
    "water_level"
]

TARGET_COLUMN = "label"

TEST_SIZE = 0.15
VALIDATION_SPLIT = 0.15
RANDOM_STATE = 42

EPOCHS = 50
BATCH_SIZE = 32
LEARNING_RATE = 0.001
DROPOUT_RATE = 0.2
```

Лістинг Б.2 - Файл data_loader.py

```
import pandas as pd
from config import FEATURE_COLUMNS, TARGET_COLUMN

def load_data(file_path: str) -> pd.DataFrame:
    """
    Завантажує набір даних із CSV-файлу та перевіряє наявність
    обов'язкових стовпців.
    """
    df = pd.read_csv(file_path)

    required_columns = FEATURE_COLUMNS + [TARGET_COLUMN]
    missing_columns = [col for col in required_columns if col not in df.columns]

    if missing_columns:
        raise ValueError(f"Відсутні обов'язкові стовпці: {missing_columns}")

    return df

def show_dataset_info(df: pd.DataFrame) -> None:
    """
    Виводить загальну інформацію про набір даних.
    """
    print("Розмір набору даних:", df.shape)
    print("\nПерші записи:")
```

```

print(df.head())
print("\nКількість пропусків:")
print(df.isnull().sum())

```

Лістинг Б.3 – Файл preprocessing.py

```

import joblib
import pandas as pd

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical

from config import (
    FEATURE_COLUMNS,
    TARGET_COLUMN,
    TEST_SIZE,
    RANDOM_STATE,
    SCALER_PATH,
    IMPUTER_PATH,
    ENCODER_PATH
)

def preprocess_data(df: pd.DataFrame):
    """
    Виконує попередню обробку сенсорних даних:
    - відокремлення ознак і цільової змінної;
    - заповнення пропусків;
    - нормалізацію числових ознак;
    - кодування класів.
    """
    X = df[FEATURE_COLUMNS].copy()
    y = df[TARGET_COLUMN].copy()

    imputer = SimpleImputer(strategy="median")
    X_imputed = imputer.fit_transform(X)

    scaler = MinMaxScaler()
    X_scaled = scaler.fit_transform(X_imputed)

    encoder = LabelEncoder()
    y_encoded = encoder.fit_transform(y)
    y_categorical = to_categorical(y_encoded)

    joblib.dump(scaler, SCALER_PATH)
    joblib.dump(imputer, IMPUTER_PATH)
    joblib.dump(encoder, ENCODER_PATH)

    return X_scaled, y_categorical, encoder

def split_dataset(X, y):
    """
    Розподіляє набір даних на навчальну та тестову вибірки.
    """
    X_train, X_test, y_train, y_test = train_test_split(
        X,
        y,
        test_size=TEST_SIZE,
        random_state=RANDOM_STATE,
        stratify=y.argmax(axis=1)
    )

```

```
return X_train, X_test, y_train, y_test
```

Лістинг Б.4 - Файл `model.py`

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam

from config import LEARNING_RATE, DROPOUT_RATE

def build_dnn_model(input_dim: int, num_classes: int):
    """
    Створює архітектуру глибокої нейронної мережі
    для багатокласової класифікації стану тепличного середовища.
    """
    model = Sequential([
        Dense(64, activation="relu", input_shape=(input_dim,)),
        Dense(32, activation="relu"),
        Dropout(DROPOUT_RATE),
        Dense(16, activation="relu"),
        Dense(num_classes, activation="softmax")
    ])

    optimizer = Adam(learning_rate=LEARNING_RATE)

    model.compile(
        optimizer=optimizer,
        loss="categorical_crossentropy",
        metrics=["accuracy"]
    )

    return model
```

Лістинг Б.5 - Файл `train.py`

```
from tensorflow.keras.callbacks import EarlyStopping

from config import (
    DATA_PATH,
    MODEL_PATH,
    EPOCHS,
    BATCH_SIZE,
    VALIDATION_SPLIT
)

from data_loader import load_data, show_dataset_info
from preprocessing import preprocess_data, split_dataset
from model import build_dnn_model
from evaluate import evaluate_model
from visualize import plot_training_history

def train():
    """
    Основна функція навчання DNN-моделі.
    """
    df = load_data(DATA_PATH)
    show_dataset_info(df)

    X, y, encoder = preprocess_data(df)
    X_train, X_test, y_train, y_test = split_dataset(X, y)
```

```

input_dim = X_train.shape[1]
num_classes = y_train.shape[1]

model = build_dnn_model(input_dim, num_classes)

early_stop = EarlyStopping(
    monitor="val_loss",
    patience=5,
    restore_best_weights=True
)

history = model.fit(
    X_train,
    y_train,
    validation_split=VALIDATION_SPLIT,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    callbacks=[early_stop],
    verbose=1
)

model.save(MODEL_PATH)

evaluate_model(model, X_test, y_test, encoder)
plot_training_history(history)

print("Навчання завершено. Модель збережено:", MODEL_PATH)

if __name__ == "__main__":
    train()

```

Лістинг Б.6 – Файл `evaluate.py`

```

import numpy as np

from sklearn.metrics import (
    accuracy_score,
    classification_report,
    confusion_matrix
)

def evaluate_model(model, X_test, y_test, encoder):
    """
    Виконує оцінювання навченої моделі на тестовій вибірці.
    """
    y_pred = model.predict(X_test, verbose=0)

    y_pred_classes = np.argmax(y_pred, axis=1)
    y_true_classes = np.argmax(y_test, axis=1)

    accuracy = accuracy_score(y_true_classes, y_pred_classes)

    print("\nЗагальна точність моделі:", round(accuracy, 4))

    print("\nЗвіт класифікації:")
    print(
        classification_report(
            y_true_classes,
            y_pred_classes,
            target_names=encoder.classes_
        )
    )

```

```

print("\nМатриця невідповідностей:")
print(confusion_matrix(y_true_classes, y_pred_classes))

```

Лістинг Б.7 - Файл predict.py

```

import joblib
import pandas as pd
import numpy as np

from tensorflow.keras.models import load_model

from config import (
    FEATURE_COLUMNS,
    MODEL_PATH,
    SCALER_PATH,
    IMPUTER_PATH,
    ENCODER_PATH
)

def load_prediction_objects:
    pass

```

Правильний варіант функції завантаження об'єктів подано нижче.

```

def load_prediction_objects():
    """
    Завантажує навчену модель і допоміжні об'єкти
    для обробки нових сенсорних даних.
    """
    model = load_model(MODEL_PATH)
    scaler = joblib.load(SCALER_PATH)
    imputer = joblib.load(IMPUTER_PATH)
    encoder = joblib.load(ENCODER_PATH)

    return model, scaler, imputer, encoder

def predict_state(input_data: dict):
    """
    Прогнозує стан тепличного середовища за новим набором
    сенсорних параметрів.
    """
    model, scaler, imputer, encoder = load_prediction_objects()

    input_df = pd.DataFrame([input_data])
    input_df = input_df[FEATURE_COLUMNS]

    input_imputed = imputer.transform(input_df)
    input_scaled = scaler.transform(input_imputed)

    probabilities = model.predict(input_scaled, verbose=0)

    predicted_index = np.argmax(probabilities, axis=1)[0]
    predicted_label = encoder.inverse_transform([predicted_index])[0]

    return predicted_label, probabilities[0]

if __name__ == "__main__":
    sample = {
        "temperature": 27.5,
        "humidity_air": 68.0,
        "soil_moisture": 42.0,
        "light": 720.0,
        "ph": 6.4,
    }

```

```

        "nitrogen": 38.0,
        "phosphorus": 22.0,
        "potassium": 31.0,
        "water_level": 65.0
    }

    label, probs = predict_state(sample)

    print("Визначений стан:", label)
    print("Ймовірності класів:", probs)

```

Лістинг Б.8 – Файл `visualize.py`

```

import matplotlib.pyplot as plt

def plot_training_history(history):
    """
    Буде та зберігає графіки точності й функції втрат
    під час навчання моделі.
    """
    plt.figure(figsize=(8, 5))
    plt.plot(history.history["accuracy"], label="Train accuracy")
    plt.plot(history.history["val_accuracy"], label="Validation accuracy")
    plt.xlabel("Епоха")
    plt.ylabel("Accuracy")
    plt.title("Динаміка точності навчання DNN-моделі")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.savefig("results/accuracy_plot.png")
    plt.close()

    plt.figure(figsize=(8, 5))
    plt.plot(history.history["loss"], label="Train loss")
    plt.plot(history.history["val_loss"], label="Validation loss")
    plt.xlabel("Епоха")
    plt.ylabel("Loss")
    plt.title("Динаміка функції втрат DNN-моделі")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.savefig("results/loss_plot.png")
    plt.close()

```

Лістинг Б.9 – Файл `main.py`

```

from predict import predict_state

def main():
    """
    Демонстраційний запуск інтелектуального модуля моніторингу.
    """
    sensor_data = {
        "temperature": 29.0,
        "humidity_air": 61.0,
        "soil_moisture": 35.0,
        "light": 800.0,
        "ph": 6.2,
        "nitrogen": 34.0,
        "phosphorus": 19.0,
        "potassium": 28.0,
        "water_level": 48.0
    }

```

```
state, probabilities = predict_state(sensor_data)

print("Поточний стан тепличного середовища:", state)
print("Ймовірності класів:", probabilities)

if __name__ == "__main__":
    main()
```

Лістинг Б.10 – Файл requirements.txt

```
pandas
numpy
scikit-learn
tensorflow
matplotlib
joblib
```

Лістинг Б.11 – Приклад запуску програмного модуля

```
pip install -r requirements.txt
python train.py
python main.py
```

Додаток В
Апробація результатів роботи

www.konferenciaonline.org.ua

Міжнародна наукова
інтернет-конференція

**Інформаційне суспільство:
технологічні, економічні
та технічні аспекти становлення**

Випуск 110

ISSN 2522-932X

Google Scholar



AKADEMIA NAUK STOSOWANYCH
WYŻSZA SZKOŁA ZARZĄDZANIA I ADMINISTRACJI
W OPOLE

7-8 травня 2026 р.

м. Тернопіль, Україна – м. Ополе, Польща
2026

УДК 001 (063)

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол.: О. Патряк та ін. ГО "Наукова спільнота", WSZIA w Opolu. Тернопіль: ФО-П Шпак В.Б. 2026. 195 с. – ISSN 2522-932X

Збірник доповідей підготовлено за матеріалами Міжнародної наукової інтернет-конференції (випуск 110) 7-8 травня 2026 р. на сайті www.konferenciaonline.org.ua

Оргкомітет ГО Наукова спільнота:

Патряк Олександра Тарасівна, кандидат економічних наук, ЗУНУ;

Шевченко (Оєїнська) Анастасія Юріївна, кандидат економічних наук, директор ТОВ «Школа майбутнього»;

Назарчук Оксана Михайлівна, доктор філософії (Ph.D.), ННІ «Юридичний інститут КНЕУ імені Вадима Гетьмана»;

Гомотюк Оксана Євгенівна, доктор історичних наук, професор, ЗУНУ;

Біловус Леся Іванівна, доктор історичних наук, кандидат філологічних наук, професор, ЗУНУ;

Ребуха Лілія Зіновіївна, доктор педагогічних наук, кандидат психологічних наук, професор, ЗУНУ;

Недошитко Ірина Романівна, кандидат історичних наук, доцент, ЗУНУ;

Стефанішин Олена Василівна, кандидат історичних наук, доцент, ЗУНУ;

Яблонська Наталія Мирославівна, кандидат філологічних наук, старший викладач, ЗУНУ;

Рудакевич Оксана Мирославівна, кандидат філософських наук, ЗУНУ;

Русенко Святослав Ярославович, викладач ВСП ФКЕПТ ЗУНУ.

Тексти матеріалів конференції подаються в авторській редакції. Відповідальність за точність, достовірність і зміст поданих матеріалів несуть автори. Всі роботи ліцензуються відповідно до Creative Commons Attribution 4.0 International License.

Автори зберігають авторське право, а також надають збірнику право першого опублікування оригінальних наукових статей на умовах ліцензії Creative Commons Attribution 4.0 International License, що дозволяє іншим розповсюджувати роботу з визнанням авторства твору та першої публікації в цьому збірнику.

Наша адреса: Оргкомітет МНІК "Конференція онлайн"

а/с 797, м. Тернопіль 46005

тел. моб. 068 366 0 525

e-mail: inetkonf@ukr.net

URL Інтернет-конференції: <http://www.konferenciaonline.org.ua/>

ISSN 2522-932X

© ГО "Наукова спільнота" 2026

© Автори статей 2026



Віпшовський Юрій Андрійович, Грицюк Юрій Іванович ПРОСТОРОВА ЛОКАЛІЗАЦІЯ АНОМАЛІЙ НА МЕТАЛЕВИХ ПОВЕРХНЯХ ЗА ДОПОМОГОЮ АНАЛІЗУ ОДНОВИМІРНИХ ПРОЕКЦІЙ ІНТЕНСИВНОСТІ.....	24
Гуцул Олександр Станіславович, Танасюк Юлія Володимирівна ІНТЕЛЕКТУАЛЬНИЙ МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ЦИФРОВОЇ АВТОМАТИЗАЦІЇ ТРЕНАЖЕРНИХ ЗАЛІВ.....	27
Дерев'ягин Ярослав Вікторович АПРОКСИМАЦІЯ КРИВИХ ДЛЯ АВТОМАТИЗАЦІЇ ГРАФОАНАЛІТИЧНОГО РОЗРАХУНКУ.....	32
Комар Мирослав Петрович, Горбаль Юлія-Марія Мар'янівна ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ КОНТЕНТОМ МІЖНАРОДНОГО ОСВІТНЬОГО ПРОЄКТУ.....	35
Костоглод Анастасія Юріївна АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ОПТИМІЗАЦІЇ РЕЖИМІВ РОБОТИ ІОТ-ПРИСТРОЇВ.....	38
Красовський Андрій Олександрович ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ НАРАТИВІВ.....	41
Майків Ігор Мирославович, Стецюк Арсен Ігорович АВТОМАТИЗАЦІЯ ЗРОШЕННЯ НА БАЗІ ДАНИХ ІОТ-СЕНСОРІВ ТА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ.....	46
Майків Ігор Мирославович, Цьомик Олег Ігорович МОНІТОРИНГ СТАНУ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ТЕПЛИЦІ НА ОСНОВІ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ.....	50
Мельник Назар Борисович, Стрижеус Матвій Юрійович ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ТА ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ДОСТОВІРНОГО КОНТЕНТУ НА ОСНОВІ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ.....	54

*Майків Ігор Мирославович,
кандидат технічних наук, доцент,
Західноукраїнський національний університет, м. Тернопіль*

*Цьомик Олег Ігорович, бакалавр,
Західноукраїнський національний університет, м. Тернопіль*

МОНІТОРИНГ СТАНУ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР У ТЕПЛИЦІ НА ОСНОВІ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ

Інтернет-адреса публікації на сайті:
<http://www.konferenciaonline.org.ua/ua/article/id-2606/>

Сільське господарство в умовах цифрової трансформації дедалі активніше інтегрує засоби Інтернету речей, методи машинного навчання та інтелектуальні програмні модулі для підвищення ефективності виробничих процесів. Особливої актуальності це набуває у тепличному господарстві, де стан сільськогосподарських культур безпосередньо залежить від температури, вологості повітря, вологості ґрунту, освітленості, вмісту поживних речовин, режиму поливу та інших параметрів мікросередовища. Навіть незначні відхилення цих показників від оптимальних значень можуть призводити до уповільнення росту рослин, зниження врожайності, погіршення якості продукції та збільшення витрат ресурсів [1-4].

Функціонування інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці ґрунтується на послідовному виконанні етапів збору, підготовки, аналізу та інтерпретації даних. На відміну від звичайного контролю окремих параметрів, у запропонованому рішенні реалізовано цілісний алгоритм, у межах якого сенсорні вимірювання перетворюються на підсумкову оцінку стану тепличного середовища. Це дає змогу автоматизувати процес спостереження, зменшити вплив людського фактора та забезпечити більш швидке виявлення ризикових ситуацій [2, 3, 5].

Алгоритм функціонування модуля охоплює два взаємопов'язані режими: режим підготовки та навчання моделі; режим робочого моніторингу після навчання.

На рисунку 1 подано загальну схему алгоритму функціонування модуля моніторингу.



Рисунок 1 – Схема алгоритму функціонування інтелектуального модуля моніторингу

Алгоритм роботи інтелектуального модуля доцільно подати у вигляді такої послідовності кроків.

Крок 1. Отримання вхідних даних. На першому етапі модуль приймає поточний набір значень від IoT-сенсорів теплиці. До складу вхідного запису можуть входити температура повітря, вологість повітря, вологість ґрунту, освітленість, рН, NPK, рівень води та інші параметри, передбачені структурою системи.

Крок 2. Перевірка повноти та коректності даних. Виконується контроль наявності всіх обов'язкових полів, перевірка типів значень та належності показників до допустимих інтервалів. Якщо виявлено пропуски або аномальні значення, модуль переходить до процедури корекції даних.

Крок 3. Обробка некоректних значень. У разі виявлення пропусків або аномалій виконується один із двох варіантів: заповнення окремих відсутніх значень за заздалегідь визначеним правилом; відхилення запису, якщо рівень некоректності є критичним.

Крок 4. Попередня обробка даних. Після перевірки запис очищається від службових або зайвих атрибутів, а структура приводиться до формату, який використовується моделлю.

Крок 5. Нормалізація ознак. Числові параметри перетворюються до узгодженого масштабу. Це необхідно для того, щоб жодна з ознак не домінувала в процесі роботи нейронної мережі лише через більший числовий діапазон.

Крок 6. Формування вектора ознак. Підготовлені параметри об'єднуються у впорядкований вектор фіксованої довжини, який відповідає входу моделі DNN.

Крок 7. Подача вектора на вхід нейронної мережі. Сформований вектор ознак передається до навченої моделі глибокої нейронної мережі.

Крок 8. Обчислення вихідного класу. Модель виконує пряме поширення сигналу через приховані шари та вихідний шар Softmax. У результаті формується вектор ймовірностей належності до кожного класу стану.

Крок 9. Вибір підсумкового стану. Підсумковим результатом приймається клас із найбільшою оціненою ймовірністю. Це може бути нормальний стан, стан ризику, стан корекції або критичний стан.

Крок 10. Інтерпретація результату. Після визначення класу формується змістовне пояснення результату. Наприклад, якщо отримано критичний стан, система відображає попередження про необхідність термінового втручання.

Крок 11. Формування повідомлення або рекомендації. На основі визначеного стану користувачеві передається повідомлення, попередження або коротка рекомендація щодо зміни режиму поливу, вентиляції чи інших параметрів.

Крок 12. Збереження результату. Поточний запис і результат класифікації зберігаються в системі для подальшого аналізу, побудови історії спостережень і використання в звітності.

Крок 13. Завершення циклу та очікування нових даних. Після завершення обробки поточного запису система переходить у режим очікування наступного набору сенсорних вимірювань.

Таким чином, сформовано алгоритм функціонування інтелектуального модуля моніторингу стану сільськогосподарських культур у теплиці. Алгоритм подано як послідовність конкретних кроків, а також у вигляді схем для режиму робочого моніторингу. Запропонована логіка роботи забезпечує повний цикл обробки сенсорних даних – від їх надходження до формування підсумкового стану, повідомлення користувачу та збереження результату.

Література:

1. Bhoi A., Nayak R. P., Bhoi S. K., Sethi S., Panda S. K., Sahoo K. S., Nayyar A. IoT-IIRS: Internet of Things based intelligent-irrigation recommendation system using machine learning approach for efficient water usage. *PeerJ Computer Science*. 2021. Vol. 7. e578.
2. Contreras-Castillo J., Guerrero-Ibañez J. A., Santana-Mancilla P. C., Anido-Rifón L. SAgric-IoT: An IoT-based platform and deep learning for greenhouse monitoring. *Applied Sciences*. 2023. Vol. 13, no. 3. Article 1961. DOI: 10.3390/app13031961.
3. Escamilla-García A., Soto-Zarazúa G. M., Toledano-Ayala M., Rivas-Araiza E., Gastélum-Barrios A. Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development. *Applied Sciences*. 2020. Vol. 10, no. 11. Article 3835. DOI: 10.3390/app10113835.
4. Liang M. H., He Y. F., Chen L. J., Du S. F. Greenhouse environment dynamic monitoring system based on WiFi. *IFAC-PapersOnLine*. 2018. Vol. 51, no. 17. P. 736-740.
5. Burri S. R., Agarwal D. K., Vyas N., Duggar R. Optimizing irrigation efficiency with IoT and machine learning: A transfer learning approach for accurate soil moisture prediction. In: *2023 World Conference on Communication & Computing (WCONF)*. 2023. P. 1-6.