

Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет Комп'ютерні технології та системи  
Кафедра Комп'ютерні інформаційні технології

## Пояснювальна записка

до кваліфікаційної роботи  
магістра

на тему: «Аналіз математичних моделей для керування природним станом рослин в автоматизованій системі вирощування»  
за освітньою програмою **12 Інженерія програмного забезпечення**  
зі спеціальності: **121 Інженерія програмного забезпечення**

Виконала: студентка групи «ПЗ2221»

/Марина СТЕШЕНКО/

Керівник:

/доц. Олександр ІВАНОВ/

Нормоконтролер:

/ доц. Світлана ВОЛКОВА/

Засвідчую, що у цій роботі немає запозичень з  
праць інших авторів без відповідних посилань  
Студентка

(підпис)

Дніпро – 2024 рік

Ministry of Education and Science of Ukraine

Ukrainian State University of Science and Technologies

Faculty Computer technologies and systems  
Department Computer information technology

**Explanatory Note**  
to Master's Thesis

on the topic: «Analysis of mathematical models for natural management the state of plants in an automated growing system»

according to educational curriculum **12 software engineering**  
in the Speciality: **121 software engineering**

Done by the student of the group PZ2221:

/Maryna STESHENKO/

Scientific Supervisor:

/Oleksandr IVANOV/

Normative controller:

/Svitlana VOLKOVA/



## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1	Вступ	03.03.23 – 14.04.23	
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	15.04.23 – 22.08.23	10%
3	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	23.08.23 – 18.10.23	
4	Постановка задачі, технічне завдання	01.10.23 – 01.11.23	30%
5	Розробка інструментальних засобів дослідження	15.10.23 – 01.11.23	
6	Виконання досліджень	01.11.23 – 01.12.23	60%
7	Оформлення пояснювальної записки	01.12.23 – 10.01.23	
8	Розробка демонстраційних матеріалів	10.01.24 – 16.01.24	100%
9	Подання кваліфікаційної роботи до кафедри	18.01.23	
10	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	24.01.24	

Студентка: \_\_\_\_\_ Марина СТЕШЕНКО

Керівник роботи: \_\_\_\_\_ доц. Олександр ІВАНОВ

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра 74 с., 12 рис. , 25 табл., 3 додатки, 20 джерел.

Об'єктом цього дослідження є система вирощування рослин, яка дозволяє виконувати керування рівнем освітленості та поливу, а також зчитує показники датчиків освітлення та вологості ґрунту.

Предметом дослідження є математичні моделі, які описують залежність природного стану рослини від освітленості та вологості ґрунту і, будучи інтегрованими у програмне забезпечення системи вирощування рослин, приймають рішення щодо керування елементами освітлення та поливу для підтримання природного стану рослини оптимальним.

Метою роботи є визначення найбільш ефективної математичної моделі для керування природним станом рослин в автоматизованій системі вирощування.

Методи дослідження: аналіз математичних моделей, порівняння їхніх показників ефективності.

Результати та їхня новизна: дослідження робить внесок у розвиток декоративно-прикладної галузі, інтегрує використання сучасних технологій у дослідження в біологічних, екологічних та математичних науках, а також створює підґрунтя для подальших досліджень у сфері використання математичних моделей для моделювання природних процесів та явищ. Результати дослідження дозволять покращити існуючу систему вирощування рослин та зробити її автоматизованою, незалежною від ручного управління.

Пояснювальна записка складається зі вступу, 5 розділів, висновків, бібліографічного списку та 2 додатків.

Ключові слова: автоматизована система вирощування, природний стан рослини, математична модель.

## ЗМІСТ

Вступ.....	<b>Помилка! Закладку не визначено.</b>
Розділ 1 Проблематика вирощування рослин	<b>Помилка! Закладку не визначено.</b>
1.1 Вивчення предметної області та визначення основних термінів.....	<b>Помилка! Закладку не визначено.</b>
1.2 Визначення факторів, що впливають на стан рослини	<b>Помилка! Закладку не визначено.</b>
Висновки до розділу 1 .....	<b>Помилка! Закладку не визначено.</b>
Розділ 2 Математичні моделі та методологія дослідження	<b>Помилка! Закладку не визначено.</b>
2.1 Розгляд різних типів математичних моделей.	<b>Помилка! Закладку не визначено.</b>
2.2 Визначення переваг та обмежень кожного типу моделі.	<b>Помилка! Закладку не визначено.</b>
2.3 Обґрунтування вибору конкретних математичних моделей для аналізу.	<b>Помилка! Закладку не визначено.</b>
2.4. Опис методів збору даних. ....	<b>Помилка! Закладку не визначено.</b>
2.5. Визначення критеріїв оцінки ефективності математичних моделей. .....	<b>Помилка! Закладку не визначено.</b>
Висновки до розділу 2 .....	<b>Помилка! Закладку не визначено.</b>
Розділ 3 Практична реалізація математичних моделей	<b>Помилка! Закладку не визначено.</b>
3.1 Проектування математичних моделей для автоматизованої системи вирощування рослин. ....	<b>Помилка! Закладку не визначено.</b>
3.2 Обґрунтування обраних засобів для реалізації.	<b>Помилка! Закладку не визначено.</b>
3.3 Розробка програмних застосунків, що реалізують спроектовані математичні моделі.....	<b>Помилка! Закладку не визначено.</b>
Висновки до розділу 3 .....	<b>Помилка! Закладку не визначено.</b>
Розділ 4 Оцінка ефективності .....	<b>Помилка! Закладку не визначено.</b>
4.1 Оцінка моделі математичних рівнянь. ....	<b>Помилка! Закладку не визначено.</b>
4.2 Оцінка моделі нечіткої логіки.....	<b>Помилка! Закладку не визначено.</b>
4.3 Оцінка моделі нейронної мережі .....	<b>Помилка! Закладку не визначено.</b>
Висновки до розділу 4 .....	<b>Помилка! Закладку не визначено.</b>
Розділ 5 Висновки та перспективи подальших досліджень	<b>Помилка! Закладку не визначено.</b>

5.1 Узагальнення отриманих результатів. ....	<b>Помилка! Закладку не визначено.</b>
5.2 Формулювання можливих напрямків для подальших наукових досліджень у даній області. ....	<b>Помилка! Закладку не визначено.</b>
Висновки .....	<b>Помилка! Закладку не визначено.</b>
Список використаних джерел .....	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК А.....	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК Б .....	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК В.....	<b>Помилка! Закладку не визначено.</b>

## ВСТУП

Перші заводи, що з'явилися у XVIII столітті, зробили справжню індустріальну революцію та вивели промисловість по всьому світу на новий рівень. Десятиліття за десятиліттям ручна праця замінювалась машинами, а знаряддя та інструменти ставали станками. Поступово прилади для полегшення або автоматизації певного виду діяльності стали з'являтися і за межами великих виробництв. Маленькі електронні помічники зараз є у кожному будинку, від швейних машинок до автоматизованих годівниць для домашніх тварин. Такі прилади не лише полегшують наше життя, але й можуть відігравати вирішальну роль, якщо необхідне точне та незалежне від людського фактору виконання роботи. Особливо це відчутно у наукових сферах або у виробництвах промислового масштабу. Історія показує, що стрімкий розвиток технологій не лише полегшує наше життя, а й створює нові випробування та підвищує рівень конкуренції. Саме тому впровадження нових технологій та постійне прагнення до вдосконалення є невід'ємною частиною будь-якого успішного проекту.

Але кожному апгрейду мають передувати аналіз та розрахунки, що дозволять зрозуміти процес, виявити його недоліки та знайти способи його покращення. Одним з найкращих інструментів для такого аналізу є математичні моделі. За допомогою них можна описати об'єкт, його стани та дії, зовнішній вплив на нього та його реакцію. Використовуючи математичні моделі можна сформулювати умови для тестування та прогнозування процесу. Важливо розуміти відмінності між ними, вміти порівнювати їхні переваги та недоліки, тому що різні види математичних моделей найкраще підходять для вирішення різних задач.

## РОЗДІЛ 1 ПРОБЛЕМАТИКА ВИРОЩУВАННЯ РОСЛИН

### 1.1 Вивчення предметної області та визначення основних термінів

Рослина – це живий організм, який виростає з насіння чи спори, має коріння для забезпечення живлення та водопостачання, стебло для підтримки та транспортування речовин, листя для фотосинтезу, а також квіти та плоди для розмноження [1].

Рослини здатні до фотосинтезу – процесу, за якого вони виробляють власну їжу, використовуючи сонячне світло, воду та вуглекислий газ. Вони є основою життя на Землі, оскільки забезпечують кисень для дихання та їжу для багатьох живих істот. Рослини дуже різноманітні – від дрібних трав'янистих до величезних дерев, вони можуть існувати у найрізноманітніших умовах, від полюсів до тропіків.

Стан рослини – це опис її фізичного, біологічного та фізіологічного стану в конкретний момент часу. Це включає в себе різні аспекти, такі як здоров'я, ріст, розвиток, функціонування органів (коренів, стебла, листя, квітів), ступінь здатності до фотосинтезу, наявність хвороб, стресу або небажаних впливів навколишнього середовища [2].

Стан рослини може бути оцінений за допомогою різних параметрів, таких як колір листя, його структура, розмір, наявність чи відсутність квітів, плодів, а також швидкість росту та загальний вигляд. Відповідно, визначення стану рослини є важливим для оцінки її здоров'я та потреб у догляді.

Природний стан рослини описує її стан в умовах природного середовища без втручання людини. Це описує фізіологічний та біологічний стан рослини в її природному середовищі, включаючи вплив клімату, ґрунту, доступність води, світла, конкуренцію з іншими рослинами, взаємодію з тваринами тощо.

Природний стан рослин визначає їхню адаптацію до конкретного середовища, їхню здатність до виживання та росту в певних умовах. Він може включати аспекти, такі як розвиток кореневої системи для забезпечення живлення та води,

форму листя для максимального ухвалення світла чи механізми захисту від шкідників та хвороб. Цей стан рослини є ключовим для її екологічної ролі в природному середовищі та взаємодії з іншими живими істотами в екосистемі.

Оптимальний стан рослини відображає умови, за яких вона функціонує на найкращому рівні з точки зору росту, розвитку та продуктивності. Це той стан, коли рослина має найкращі умови для здорового росту, оптимального використання доступних ресурсів (таких як вода, світло, поживні речовини), максимального вироблення кисню (за допомогою фотосинтезу), а також для розмноження. Оптимальний стан рослини може залежати від багатьох факторів, таких як тип ґрунту, вологість, температура, доступність світла, наявність поживних речовин, вплив шкідників та хвороб, а також генетичних особливостей самої рослини. Цей стан є важливим для досягнення максимальної продуктивності рослини, її оптимального здоров'я та розвитку, що впливає на врожайність, якість плодів чи насіння, а також загальний внесок рослини в екосистему [1].

Кімнатна рослина – це рослина, яку зазвичай вирощують та доглядають у приміщенні або кімнаті з метою декорування або прикрашання інтер'єру. Вони можуть бути представлені різноманітними видами: квітучими, зеленими, сукулентами, пальмами та іншими. Кімнатні рослини часто вирощують у горщиках або контейнерах. Їхній розвиток залежить від різних умов середовища: освітлення, вологість, температури та інші.

Ці рослини можуть бути використані для створення природного елемента в приміщенні або поліпшення якості повітря. Вони допомагають створити затишну атмосферу та мають позитивний вплив на самопочуття та здоров'я людини.

Система для вирощування рослин – це комплексна структура, яка створена для культивування рослин, зазвичай у контрольованих умовах, таких як теплиці, теплиці з контрольованим кліматом, гідропонічні установки та інші. Ці системи створюються для оптимізації умов для росту рослин, забезпечуючи їхню потребу

у воді, світлі, поживних речовинах та інших факторах, що сприяють їхньому розвитку.

Системи для вирощування рослин можуть використовувати різні методи, такі як гідропоніка (де рослини ростуть у водних розчинах без ґрунту), аеропоніка (де коріння рослин знаходиться у повітрі, а зволожена краплина живить рослини), системи з контрольованим кліматом тощо.

Ці системи дозволяють оптимізувати умови для росту рослин, забезпечуючи кращу якість врожаю, ефективне використання ресурсів та зменшення впливу на навколишнє середовище. Вони широко використовуються в сільському господарстві, ботанічних садах, фермерських господарствах та дослідницьких установах для ефективного вирощування різних видів рослин [3].

Автоматизовані системи для вирощування кімнатних рослин можуть мати різні рівні автоматизації та функціональності, але основна мета полягає у створенні оптимальних умов для росту рослин без значного втручання людини.

Основні компоненти таких систем включають:

- 1) системи поливу – це автоматичні системи зволоження або системи поливу, які контролюють кількість води, яку отримують рослини;
- 2) світлодіодні системи – автоматизовані системи можуть мати спеціальні освітлювальні системи з програмованими часами світла, що підтримують оптимальне освітлення для кожного виду рослин;
- 3) системи контролю температури та вологості – вони автоматично регулюють температуру та вологість, створюючи комфортні умови для росту;
- 4) дозатори рідких добрив – автоматизовані системи можуть постачати необхідні поживні речовини за розкладом, що дозволяє рослинам отримувати оптимальне харчування;
- 5) системи моніторингу та управління – деякі системи мають сенсори, що вимірюють параметри, такі як вологість ґрунту, рівень CO<sub>2</sub>, рН рівень тощо. Ці дані використовуються для адаптації умов для рослин.

Такі системи можуть бути програмовані для автоматичного виконання певних операцій на основі певних параметрів (наприклад, включення світла або поливу, коли це потрібно), що робить процес догляду за рослинами більш ефективним та менш залежним від постійного контролю людини.

## 1.2 Визначення факторів, що впливають на стан рослини

Ріст та розвиток кімнатних рослин залежать від кількох факторів:

- 1) світло – кімнатні рослини потребують достатньої кількості світла для фотосинтезу. Від правильного освітлення залежить їхня здатність до росту та цвітіння;
- 2) вода – підтримка оптимального рівня вологості ґрунту відіграє важливу роль. Пере- або недолив води може негативно впливати на рослини;
- 3) температура – рослини мають свої власні температурні вимоги. Деякі потребують тепла, інші – прохолоди. Важливо забезпечити оптимальні умови для кожного виду;
- 4) ґрунт – він повинен бути належної якості та забезпечувати необхідну кількість поживних речовин та мінералів рослини;
- 5) вентиляція – рослини потребують свіжого повітря. Недостатність циркуляції повітря може призвести до хвороб та проблем з ростом рослини;
- 6) добрива – правильні добрива надають рослинам необхідні поживні речовини для росту та розвитку;
- 7) контейнери – правильний розмір та дренаж контейнера також важливі. Недостатній простір для кореневої системи або поганий дренаж можуть вплинути на здоров'я рослини.

Збалансоване поєднання цих факторів допоможе забезпечити здоровий та гарний ріст кімнатних рослин, але при цьому важливо враховувати потреби і особливості кожної рослини [4]. У таблиці 1.1 нижче наведено класифікацію рослин за факторами впливу.

Таблиця 1.1 – Залежність росту рослин від низки факторів

Фактори впливу	Види рослин			Поради
Освітлення	Світлолюбні: алоє, каланхое, молочай, товстянка	Тіньо- витривалі: фікуси, драцени, плющі	Помірного освітлення: хвойні, папороті, цикламени	Найбільш сприятливі кімнати із західними і східними вікнами. Деякі потребують додаткової підсвітки
Температура	Невибагливі: алоє, драцена, лимон, пальма, плющ, традесканція, фікус	Теплолюбні: дифенбахія, орхідея, сенполія	Прохолодного утримання: бегонія, каланхое, папороть	Оптимальний температурний режим у межах 18-25 °С. Не можна різко змінювати температурний режим, не слід допускати перенагрів рослин від радіаторів та уникати протягів
Полив	Рясне (одразу після просихання ґрунту): лимон, фікус, плющ	Помірне (поливають через 1 або 2 дні після просихання): хлорофітум, драцена	Нечасте (залишають сухим протягом декількох днів): кактуси	Для поливу використовують відстояну або очищену воду кімнатної температури
Вологість	Більшість рослин потребують 60% вологості	Тропічні потребують 90% вологості		Для підвищення вологості використовують побутові зволожувачі, оприскувачі

Фактори, які впливають на ріст та розвиток кімнатних рослин, можуть взаємодіяти та впливати один на одного у складному способі. Взаємодія цих факторів може мати значний вплив на здоров'я та зростання рослин.

Наприклад:

- 1) світло і вода – брак світла може вплинути на фотосинтез та розвиток рослини, а отже, вона може менше використовувати воду. З іншого боку, надмір води при недостатньому світлі може призвести до розмокання кореневої системи через низьку фотосинтетичну активність;
- 2) температура і вологість – підвищена температура може призвести до випаровування води, що може зробити рослину більш схильною до пересихання. Зниження вологості повітря від підвищеної температури може призвести до спекотного та сухого середовища, що також може

впливати на рослину;

- 3) ґрунт та удобрення – не всі види ґрунту взаємодіють однаково з добривами. Деякі ґрунти можуть тримати більше води та поживних речовин, що впливає на необхідність поливу та удобрення.

Ці взаємодії потребують уважного контролю та балансування умов, щоб забезпечити оптимальне середовище для росту кімнатних рослин. Розуміння та управління цими взаємодіями допомагає створити сприятливі умови для здоров'я та зростання рослин [5].

### 1.3 Опис існуючої системи вирощування рослин

Система вирощування рослин, що є об'єктом дослідження даної дипломної роботи, представляє собою стенд із 5 полицями, на якому може бути розміщено до 10 рослин (по дві рослини на кожній полиці). Система оснащена датчиками та елементами керування, що дозволяють слідкувати за природним станом рослини та підтримувати його оптимальним за допомогою вбудованих систем поливу та освітлення. На разі у системі доступне лише ручне керування. Взаємодія користувача з системою виконується за допомогою мобільного додатку та локального сервера, який розташований на одноплатному комп'ютері на стенді. Цей же сервер відповідає за отримання та збереження показників датчиків.

Датчики, що наявні у системі:

- 1) датчики вологості ґрунту;
- 2) датчики рівня освітленості;
- 3) датчик вологості, температури та тиску повітря.

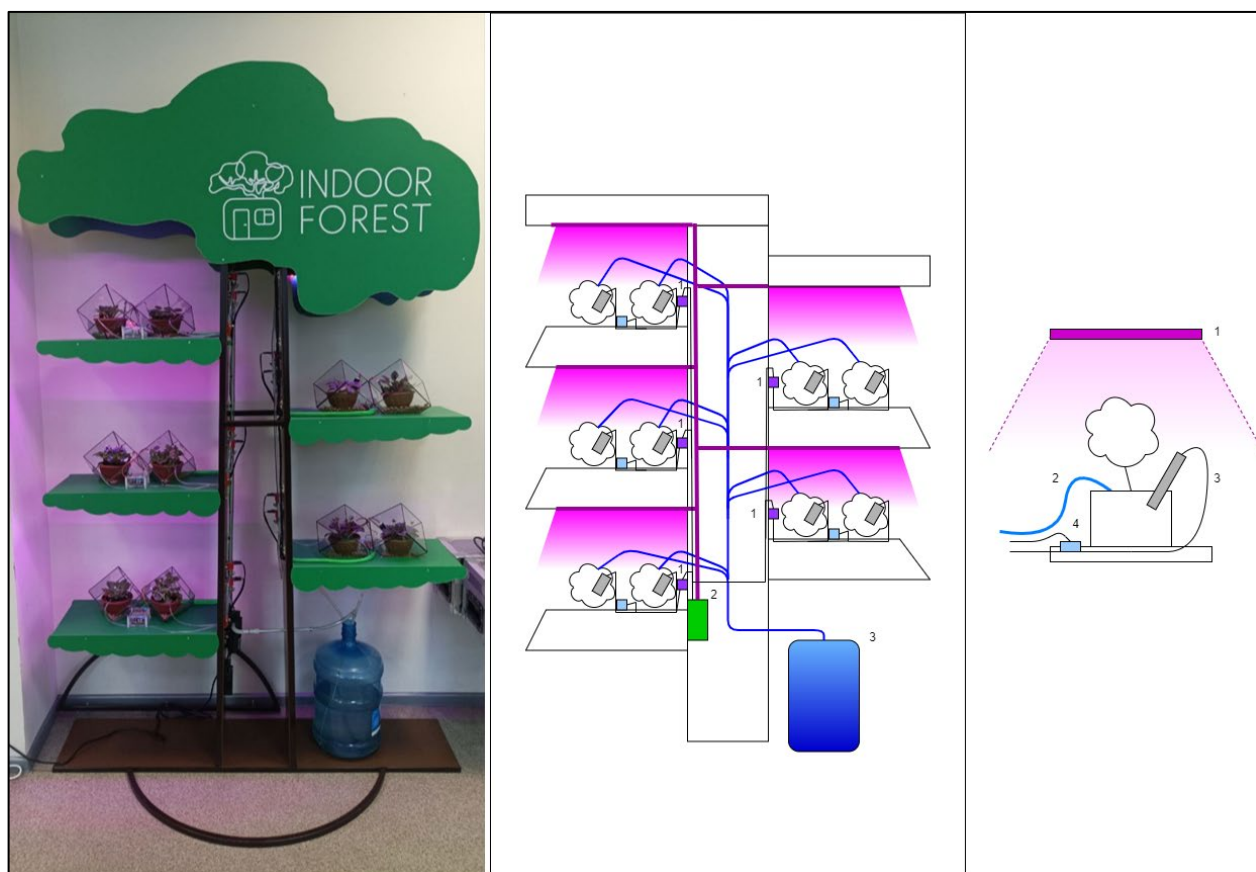
Компоненти керування, що наявні у системі:

- 1) система поливу: бутель з водою, насос, трубки.
- 2) система освітлення: ШІМ (широотно-імпульсна модуляція), фіто-стрічка.

Побудову системи вирощування наведено на рисунку 1.1.

Raspberry Pi (рисунок 1.2) побудований на системі-на-чипі (SoC) Broadcom BCM2835, яка включає в себе процесор ARM із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 чи 256 мегабайтів оперативної пам'яті.

Жорсткий диск відсутній, натомість використовується SD карта. Молодша (А) і старша (В) моделі Raspberry Pi відрізняються об'ємом оперативної пам'яті (256 і 512 мегабайтів відповідно) і кількістю USB-портів (один проти двох). Крім цього, у старшої моделі є роз'єм Ethernet 10/100, а молодша споживає на третину менше енергії. Для Raspberry Pi випущено спеціалізований дистрибутив Linux, Raspbian OS (заснований на дистрибутиві Debian) і магазин застосунків Pi Store, де є як платні, так і безплатні програми. Серед іншого Raspberry Pi використовує веб-браузер Iceweasel та KOffice, які постачаються в комплекті. Офіційна мова програмування для Raspberry Pi — Python.



а

б

в

а – зовнішній вигляд системи

б – архітектура системи (1 – мікроконтролер STM8, 2 – одноплатний комп'ютер Raspberry Pi, бутель з водою)

в – архітектура однієї рослини (1 – фітострічка, 2 – трубка для постачання води, 3 – датчик вологості ґрунту, 4 – датчик рівня освітленості)

Рисунок 1.1 – Будова системи вирощування рослин

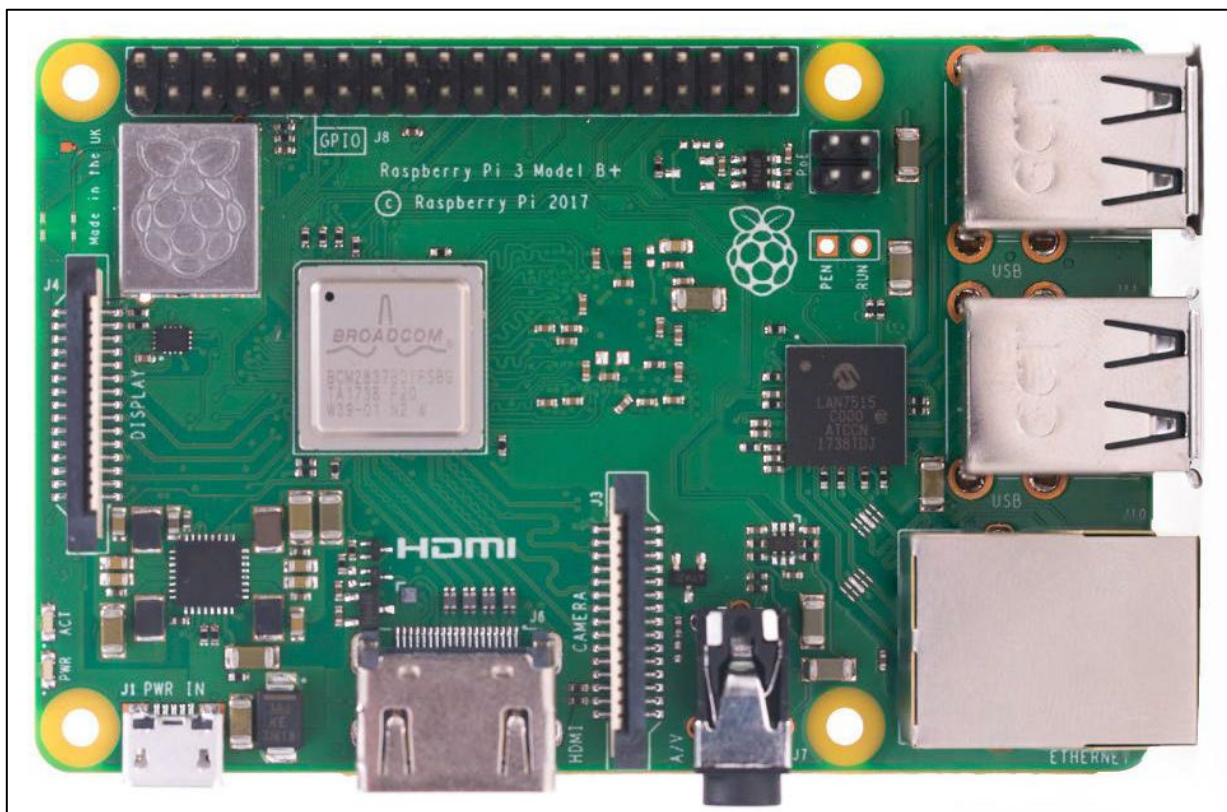


Рисунок 1.2 – Raspberry Pi

Датчик вологості ґрунту Capacitive Soil Moisture Sensor v1.2 (рис. 1.3.а) складається з двох металевих електродів, які вставляються в ґрунт. Провід, який підключений до датчика, дозволяє підключити його до мікроконтролера для зчитування даних. Принцип роботи полягає в тому, що вологий ґрунт є кращим провідником електричного струму, ніж сухий. Коли датчик вноситься у ґрунт, його електроди взаємодіють з вологістю ґрунту, що змінює електричний опір між електродами. Цей змінний опір слугує індикатором рівня вологості ґрунту. Датчик не дає прямого числового вимірювання відсоткового вмісту води в ґрунті, а замість цього він надає аналоговий сигнал, який потребує калібрування та подальшого перетворення у потрібні одиниці вимірювання.

Специфікація:

- 1) підтримує 3-контактний інтерфейс датчика;
- 2) аналоговий вихід;
- 3) робоча напруга: DC 3.3-5.5V;
- 4) вихідна напруга: DC 0-3.0V;

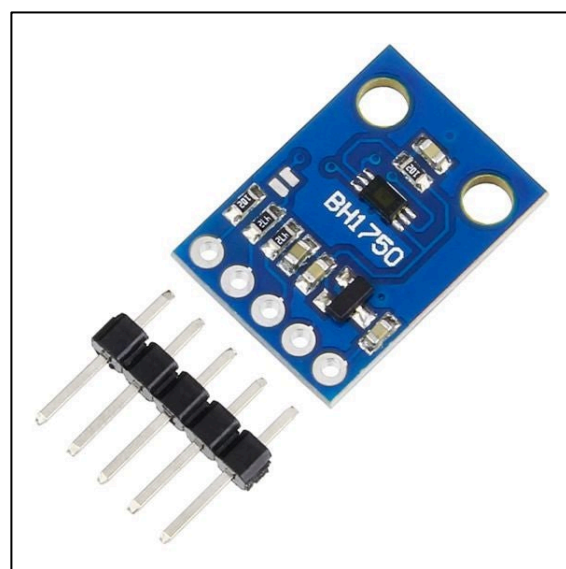
5) інтерфейс: РН2.0-3Р;

6) розмір: 99x16мм.

Датчик освітленості ВН1750 (рис. 1.3.б) – це цифровий датчик, призначений для вимірювання освітленості в навколишньому середовищі. Він здатний точно вимірювати рівень світла навколишнього середовища і видає результат у цифровому форматі, що полегшує його використання з мікроконтролерами. Цей датчик використовує технологію CMOS і здатний працювати в широкому діапазоні освітленості, що дає можливість отримувати значення як при слабкому освітленні, так і при яскравому сонячному світлі. Він має високу точність та низьке споживання енергії, що робить його популярним для використання в проектах, де важлива точність вимірювань та низький рівень споживаної енергії. ВН1750 підключається до мікроконтролера через шину I2C, що спрощує зчитування даних та інтеграцію з різними пристроями. Існують бібліотеки для багатьох мікроконтролерів, які спрощують використання цього датчика та дозволяють швидко отримувати дані про освітленість у потрібному проекті.



а



б

а – датчик вологості ґрунту Capacitive Soil Moisture Sensor v1.2

б – датчик освітленості ВН1750

Рисунок 1.3 – Датчики

Специфікація:

- 1) джерело живлення: 2.4В-3.6В (зазвичай 3.0В);
- 2) менше споживання струму: 0.12 мА;
- 3) вимірювальний діапазон: 1-65535lx;
- 4) зв'язок: шина I2C.

Доступ до даних датчиків, а також елементів керування системи, виконується за допомогою мобільного застосунку, який підключається до одноплатного комп'ютера по локальній Wi-fi мережі. Мобільний застосунок дозволяє:

- 1) переглядати поточні показники вологості ґрунту та рівня освітленості для однієї рослини;
- 2) вмикати полив на задану кількість мілілітрів для однієї рослини;
- 3) вмикати фітострічку на заданий рівень для однієї рослини.

Зв'язок компонентів системи зображено на рисунку 1.4.

Задача автоматизації даної системи полягає у реалізації наступних функцій:

- 1) зчитування показників датчиків один раз у певну кількість часу.
- 2) прийняття рішення щодо необхідного поливу та освітлення.
- 3) виконання прийнятого рішення за допомогою елементів поливу та освітлення.

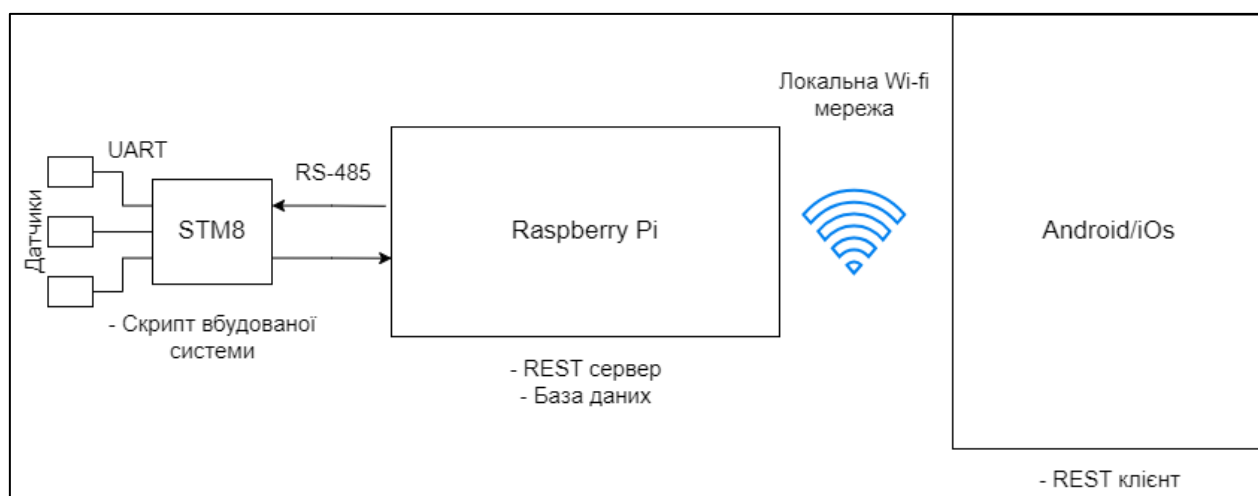


Рисунок 1.4 – Зв'язок компонентів системи

## Висновки до розділу 1

Освітлення та полив є основними факторами, що впливають на ріст та розвиток рослини. Для того, щоб забезпечити оптимальні умови середовища для кімнатної рослини необхідно враховувати її особливості та потреби. Для автоматизації керування поливом та освітленням у системі вирощування рослин потрібно спроектувати та реалізувати спосіб прийняття рішень щодо необхідної кількості поливу та необхідного рівня освітлення.

## РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ

### 2.1 Розгляд різних типів математичних моделей.

Математична модель – це абстрактне представлення реального об'єкта, системи чи явища за допомогою математичних символів, рівнянь, формул, графіків або інших математичних засобів. Вона використовується для опису, аналізу, передбачення та розуміння реальних процесів, що відбуваються в природі, науці, техніці, економіці та інших галузях [6].

Використання математичних моделей відбувалося протягом великої частини історії людства у різних галузях знань. Вже в давньому світі вчені використовували математичні відношення та моделі для розуміння астрономічних явищ [13]. Наприклад, в Стародавньому Єгипті та Месопотамії проводились спостереження небесних тіл, щоб створити календарі та передбачати астрономічні події. Епоха Відродження (XIV-XVI століття) була періодом значного розвитку науки та математики. Тут вчені почали формалізувати закони фізики та розвивати математичні моделі, щоб пояснити природні явища. Наприклад, Галілео Галілей використовував математичні моделі для опису руху тіл та падіння предметів [11]. У XVII столітті розвиток науки та математики прискорився завдяки внескам вчених, таких як Ісаак Ньютон та Готфрід Вільгельм Лейбніц. Ньютон заснував класичну механіку, формалізувавши закони руху та впровадивши диференціальні та інтегральні рівняння, що стали основою для математичних моделей фізичних явищ. У XIX столітті, з появою індустріальної революції, математичні моделі стали використовуватися для оптимізації виробничих процесів, розвитку технологій та побудови машин [6]. Розробка комп'ютерів та поява чисельних методів обчислень дозволила використовувати більш складні математичні моделі для моделювання складних систем, таких як клімат, економіка, біологія, техніка та інші галузі.

Сьогодні математичні моделі використовуються в найрізноманітніших галузях: від прогнозування погоди та економічних процесів до проектування комп'ютерних програм, розвитку медицини та аналізу соціальних систем. Вони є невід'ємною частиною наукового дослідження та технологічного розвитку [7].

Різні види математичних моделей використовуються для вирішення різних задач [8].

Основні види математичних моделей:

- 1) аналітичні моделі – вони використовуються для опису за допомогою аналітичних виразів, рівнянь та функцій. Наприклад, математична модель, яка описує закони руху тіла у фізиці [18]. Приклади аналітичних моделей:
  - математичні рівняння;
  - деякі типи диференціальних рівнянь.
- 2) символічні моделі – використовують символи та спеціальні позначення для представлення системи або процесу. Наприклад, логістична модель в економіці [19]. Приклади символічних моделей:
  - теорія графів;
  - нечітка логіка;
  - нейронні мережі.
- 3) системи диференціальних рівнянь – це моделі, що використовують диференціальні рівняння для опису зміни змінних у часі. Вони широко використовуються в фізиці, біології, економіці та інших галузях [20]. Приклади систем диференціальних рівнянь:
  - модель Реакція-Дифузія;
  - деякі моделі в фізиці та біології.
- 4) симуляційні моделі – вони використовуються для імітації реальних процесів на комп'ютері або інших пристроях. Це може бути моделювання руху транспорту, розповсюдження хвороби тощо [15].

Приклади симуляційних моделей:

- комп'ютерне моделювання;
- методи оптимізації;
- генетичні алгоритми.

5) статистичні моделі – використовуються для аналізу випадкових процесів та взаємозв'язків між даними. Наприклад, регресійні моделі для прогнозування змін змінних на підставі статистичних даних[11].

Приклади статистичних моделей:

- Байєсівська статистика;
- теорія ймовірностей;
- регресійні моделі.

2.2 Визначення переваг та обмежень кожного типу моделі.

Переваги аналітичних моделей:

- 1) точність – аналітичні моделі дозволяють отримати точні або апроксимовані аналітичні рішення для конкретних умов задачі;
- 2) універсальність – деякі класичні аналітичні рішення можуть бути застосовані до широкого спектру задач, що спрощує їх застосування у різних областях;
- 3) розуміння – вони часто допомагають розуміти сутність процесів, які вони моделюють, через аналітичні рішення та математичні властивості.

Обмеження аналітичних моделей:

- 1) складність задач – для деяких складних систем або процесів може бути вкрай складно (а іноді навіть неможливо) отримати аналітичні рішення;
- 2) спрощення – у намаганні знайти аналітичне рішення деякі аспекти можуть бути спрощені або ідеалізовані, що призводить до втрати точності моделі;
- 3) обмежена універсальність – деякі аналітичні рішення можуть бути застосовані лише до обмеженого спектру умов, адже реальні процеси можуть бути більш складними, ніж модель передбачає;

- 4) часові витрати – для складних аналітичних моделей може бути потрібно значно більше часу на розробку та знаходження рішення, порівняно з іншими методами моделювання.

Таким чином, аналітичні моделі є потужним інструментом для розв'язання різних проблем, проте вони не завжди можуть охопити всю складність реальних процесів. Тому в багатьох випадках вони використовуються у поєднанні з іншими методами моделювання для отримання більш повного розуміння [8].

Переваги символічних моделей:

- 1) універсальність – символічні моделі, такі як теорія графів чи нечітка логіка, можуть застосовуватися до різноманітних галузей, що робить їх універсальними і корисними для вирішення різних завдань;
- 2) гнучкість – вони дозволяють працювати з нечіткістю, неоднозначністю та комплексністю даних, що часто важко або неможливо узагальнити у традиційних математичних моделях;
- 3) високий рівень абстракції – ці моделі працюють на рівні абстракції, що дозволяє відокремлювати ключові аспекти проблеми від деталей.

Обмеження символічних моделей:

- 1) неоднозначність і інтерпретація – оскільки символічні моделі працюють з нечіткістю, інтерпретація результатів може бути складною, оскільки рішення часто залежать від контексту;
- 2) складність обробки – деякі символічні моделі, наприклад, нейронні мережі, можуть бути дуже складними у розумінні та налаштуванні, особливо у випадку великої кількості параметрів;
- 3) неуніверсальність – деякі символічні моделі можуть бути менш універсальними у порівнянні з класичними аналітичними моделями та не завжди підходять для розв'язання широкого спектру завдань;
- 4) відсутність точного рішення – оскільки символічні моделі працюють з нечіткістю чи великим обсягом даних, точне аналітичне рішення не завжди можливе.

Символічні моделі використовуються для моделювання складних систем, де точні рішення часто не можливі або де необхідно працювати з нечіткими, неоднозначними даними. Однак їх використання також пов'язане з труднощами у тлумаченні результатів та управлінні складністю моделей [7].

Переваги систем диференціальних рівнянь:

- 1) моделювання зміни в часі – ці моделі дозволяють описувати зміни величин в часі, що особливо важливо для фізичних, біологічних та інших процесів, які залежать від часу;
- 2) гнучкість в описі складних систем – диференціальні рівняння можуть бути використані для моделювання складних систем з багатьма змінними та параметрами, що залежать від часу чи інших змінних;
- 3) узагальненість – диференціальні рівняння є потужним інструментом для опису фізичних законів, які лежать в основі багатьох явищ у природі та інших галузях.

Обмеження систем диференціальних рівнянь:

- 1) складність розв'язання – для багатьох систем диференціальних рівнянь важко (а іноді неможливо) знайти аналітичні рішення, і тому необхідно користуватися чисельними методами для отримання наближених розв'язків;
- 2) чутливість до початкових умов та параметрів -диференціальні рівняння можуть бути дуже чутливими до початкових умов і параметрів, що робить необхідним точне задання цих умов для отримання точних розв'язків;
- 3) неуніверсальність – деякі фізичні системи можуть бути складними для моделювання за допомогою диференціальних рівнянь через їх складність та взаємодії;
- 4) складність у використанні для деяких систем – для деяких систем зі складними взаємодіями або нестандартними умовами моделювання може бути вкрай складним.

Диференціальні рівняння є потужним інструментом для моделювання змін в часі в різних галузях, але їх використання вимагає ретельності та уважності у визначенні початкових умов та параметрів, а також може потребувати чисельних методів для отримання розв'язків [10]. Переваги симуляційних моделей:

- 1) моделювання складних систем – симуляційні моделі дозволяють моделювати поведінку складних систем з великою кількістю взаємопов'язаних компонентів, що робить їх корисними для аналізу складних реальних систем;
- 2) експерименти на віртуальних системах – вони дозволяють виконувати експерименти на віртуальних моделях без необхідності реальних експериментів, що може зберегти час і ресурси;
- 3) тестування гіпотез – симуляційні моделі можуть використовуватися для тестування гіпотез та вивчення впливу різних факторів на систему, допомагаючи в розумінні її поведінки;
- 4) прогнозування – вони можуть допомогти у прогнозуванні майбутньої поведінки системи на основі введених даних та параметрів.

Обмеження симуляційних моделей:

- 1) складність створення – розробка складних симуляційних моделей може бути вкрай складною та витратною за часом, особливо для систем з великою кількістю деталей;
- 2) необхідність точних даних та параметрів – точність результатів симуляцій може залежати від якості введених даних та параметрів моделі, що може бути важко отримати для деяких систем;
- 3) складність валідації – валідація симуляційних моделей - це складний процес, оскільки потрібно переконатися, що модель адекватно відображає реальну систему;
- 4) обмеження точності – деякі симуляційні моделі можуть мати обмежену точність у відображенні реальних систем через спрощення та узагальнення введених даних чи параметрів.

Симуляційні моделі дозволяють аналізувати складні системи та відтворювати їхню поведінку, але їхнє використання також пов'язане з викликами у створенні, валідації та використанні для прогнозування та аналізу складних систем [9].

Переваги статистичних моделей:

- 1) універсальність – вони можуть застосовуватися до широкого спектру даних та відносно різних дисциплін, забезпечуючи універсальний підхід до аналізу;
- 2) стійкість – статистичні моделі можуть пристосовуватися до різноманітних типів даних, включаючи як кількісні, так і якісні показники;
- 3) прогнозування та узагальнення – вони можуть бути використані для прогнозування майбутніх подій на основі наявних даних та для узагальнення результатів на більш широку популяцію;
- 4) виявлення залежностей – статистичні моделі дозволяють виявляти складні залежності та взаємозв'язки між різними змінними у даних.

Обмеження статистичних моделей:

- 1) припущення про дані – результати статистичних моделей можуть бути залежні від припущень про розподіл даних та інших параметрів, які не завжди можна точно визначити;
- 2) недостатня увага до контексту – деякі моделі можуть спрощувати аналіз, ігноруючи важливі контекстуальні аспекти, що може призвести до неправильних висновків;
- 3) обмежена узагальнююча здатність – деякі статистичні моделі можуть бути обмежені у здатності узагальнювати результати на нові дані або на різні контексти;
- 4) чутливість до викидів – деякі статистичні методи можуть бути чутливими до викидів чи відсутності деяких даних, що може спричинити неточні або неправильні результати.

Статистичні моделі є потужним інструментом для аналізу даних, прогнозування та розуміння залежностей. Однак вони також мають свої

обмеження, які важливо враховувати при їхньому застосуванні для аналізу та прийняття рішень [9].

### 2.3 Обґрунтування вибору конкретних математичних моделей для аналізу.

Проаналізуємо перераховані переваги та обмеження різних видів моделей для того щоб визначити, які моделі найкраще підійдуть для вирішення задачі автоматизації керування природним станом рослини у системі вирощування. Для цього перерахуємо характеристики моделей та відмітимо ті, які були б давали перевагу для поставленої задачі. Результати порівнянь моделей наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння характеристик моделей

Тип математичних моделей	Властивість	Є перевагою
Аналітичний	Точність	+
	Універсальність	+
	Розуміння	–
Символьний	Універсальність	+
	Гнучкість	–
	Високий рівень абстракції	+
Системи диференціальних рівнянь	Моделювання змін у часі	–
	Гнучкість в описі складних систем	–
	Узагальненість	–
Симуляційні моделі	Моделювання складних систем	–
	Експерименти на віртуальних системах	+
	Тестування гіпотез	–
Статистичні моделі	Універсальність	+
	Стійкість	–
	Прогнозування та узагальнення	–

Наступні характеристики математичних моделей були визначені як переваги у вирішенні задачі керування природним станом рослини у системі вирощування:

- точність – модель зможе надавати точні прогнози та результати, це дозволить краще контролювати параметри середовища, поливу, інтенсивність світла тощо;
- універсальність – модель зможе бути застосована до різних типів рослин, це дозволить адаптувати її до різноманітних умов вирощування та культурних особливостей кожної рослини;
- високий рівень абстракції – модель зможе абстрагувати складні фізичні процеси, зменшуючи їхню складність до ключових параметрів та взаємодій, це полегшить розуміння та управління системою вирощування рослин;
- експерименти на віртуальних системах – модель дозволить проводити експерименти в віртуальному середовищі, що може бути ефективним та економічним способом тестування різних стратегій та умов вирощування.

Отже, для автоматизованої системи вирощування найкращими будуть математичні моделі аналітичного та символного видів через їхню універсальність, точність та високий рівень абстракції.

Розглянемо конкретні приклади обраних видів математичних моделей більш детально.

Переваги математичних рівнянь:

- аналітичність – можливість отримати точні аналітичні рішення;
- простота інтерпретації – легка інтерпретація та розуміння математичних рівнянь;
- широкий спектр застосувань – застосовується у фізиці, економіці, інженерії та інших галузях.

#### Обмеження математичних рівнянь:

- обмеженість формулювань – математичні рівняння можуть бути недостатніми для опису складних фізичних процесів;
- обмеженість рішень – не всі рівняння мають аналітичні рішення; іноді потрібні чисельні методи для знаходження наближених розв'язків.

#### Переваги диференціальних рівнянь:

- моделювання зміни – дозволяють описувати зміну систем в часі або просторі;
- універсальність – використовуються для опису фізичних, біологічних та економічних процесів.

#### Обмеження диференціальних рівнянь:

- складність розв'язання – деякі диференціальні рівняння складно або навіть неможливо розв'язати аналітично;
- неоднозначність розв'язків – деякі рівняння можуть мати багато різних розв'язків або не мати розв'язку взагалі.

#### Переваги теорії графів:

- універсальність застосування – модель дозволяє моделювати різноманітні взаємозв'язки та системи у вигляді графів;
- ефективність алгоритмів – існує багато ефективних алгоритмів для розв'язання проблем теорії графів, таких як пошук найкоротших шляхів, потоків у мережах, тощо;
- використання у різних галузях – застосовується в комп'ютерних науках, логістиці, соціальних мережах, транспортних системах тощо.

#### Обмеження теорії графів:

- складність масштабування – деякі задачі на великих графах можуть бути складними для обробки через обмежену обчислювальну потужність;
- чутливість до змін структури – додавання або видалення вузлів може суттєво змінити характеристики графа.

#### Переваги нечіткої логіки:

- універсальність у моделюванні нечітких концепцій – дозволяє виразити нечіткість та невизначеність в мові арифметики;
- пристосованість до реальних умов – добре підходить для моделювання реальних ситуацій, де поняття нечіткості важливі;
- потужність у прийнятті рішень – дозволяє приймати рішення в умовах нечіткості та невизначеності.

#### Обмеження нечіткої логіки:

- складність інтерпретації – іноді складно інтерпретувати результати, оскільки модель базується на нечітких концепціях;
- потреба у досвідченості – необхідно враховувати експертні знання для правильного визначення нечітких множин і правил.

#### Переваги нейронних мереж:

- здатність до навчання на основі даних – можливість автоматичного навчання для виявлення складних залежностей у великих обсягах даних;
- універсальність застосування – застосовується у багатьох сферах, включаючи комп'ютерне бачення, природні мови, медицину тощо;
- адаптивність до складних завдань – можливість вирішувати складні задачі, які важко формалізувати традиційними методами.

#### Обмеження нейронних мереж:

- потреба у великій кількості даних – для навчання ефективної моделі може знадобитися велика кількість даних;
- чорний ящик – у складних нейронних мережах може бути складно пояснити, як саме модель приймає рішення (недостатній рівень інтерпретованості).

Перераховані характеристики математичних моделей порівняно у таблиці 2.2.

Таблиця 2.2 – Порівняння характеристик моделей

Модель	Характеристика	Є перевагою
Математичні рівняння	Аналітичність	+
	Простота інтерпретації	+
	Широкий спектр застосувань	+
Диференціальні рівняння	Моделювання зміни	–
	Універсальність	+
Теорія графів	Універсальність	+
	Ефективність алгоритмів	–
	Використання у різних науках	–
Нечітка логіка	Універсальність	+
	Пристосованість до реальних умов	+
	Потужність у прийнятті рішень	+
Нейронні мережі	Здатність до навчання на основі даних	+
	Універсальність	+
	Адаптивність	+

Враховуючи вищенаведені переваги та недоліки конкретних математичних моделей аналітичного та символного типів, найкращими моделями для автоматизованої системи вирощування рослин будуть математичні рівняння, нечітка логіка та нейронні мережі. Математичні рівняння відзначаються високим рівнем точності та здатністю враховувати різноманітні фізичні процеси у вирощуванні рослин. Нечітка логіка дозволяє враховувати невизначеність та нечіткість в умовах вирощування, що є важливим в реальних аграрних умовах. Нейронні мережі, використовуючи свою здатність до навчання, можуть адаптуватися до змін у середовищі та оптимізувати стратегії вирощування[16]. Ці моделі є універсальними, спроможними вирішувати різноманітні завдання, і водночас мають індивідуальні переваги, такі як простота інтерпретації та пристосованість до реальних умов вирощування, що робить їх ефективними інструментами для оптимізації та автоматизації процесів вирощування рослин.

#### 2.4. Опис методів збору даних.

Збір даних для побудови та перевірки математичних моделей проводився на системі вирощування рослин, що є об'єктом дослідження. Збір проводився у будні дні протягом двох місяців з 30 жовтня 2023 року до 22 грудня 2023. У якості рослини використовувалась африканська фіалка (*Saintpaulia*) також відома як сенполія. Сенполія має наступні вимоги до догляду:

- 1) горщик діаметром 6-10 см;
- 2) температурний режим +18..+24 °С;
- 3) яскраве освітлення протягом 13-14 годин на добу;
- 4) рівномірний, але не рясний полив.

Згідно цих умов були визначені наступні вимоги до рівня освітлення та вологості ґрунту:

- 1) вологість ґрунту: 50-60%;
- 2) рівень освітлення: 10 000-15 000 лк.

Протягом збору даних, система вирощування рослин знаходилась у приміщенні, у якому підтримувалась середня температура 24°С. Отримання даних з датчиків здійснювалось кожні 3 години з 9 ранку до 18 вечора з однієї рослини на стенді. Отримані показники освітлення та вологості, а також виконані дії наведені у таблиці 2.3, де Н – показник вологості ґрунту у відсотках, L – показник рівня освітленості у люксах, w – кількість води, що була влита у горщик, у мілілітрах, f – рівень освітлення, що був ввімкнений на фітострічці.

Таблиця 2.3 – Фактичні дані

Дата	9:00				12:00				15:00				18:00			
	H	L	w	f	H	L	w	f	H	L	w	f	H	L	w	f
30.10	47	7204	100	8000	61	10935	0	4000	59	8762	0	6000	58	518	0	14000
31.10	54	7251	100	8000	65	11389	0	3000	64	8903	0	6000	63	524	0	14000
01.11	59	7004	0	8000	57	11035	0	4000	55	8498	0	6000	54	513	0	14000
02.11	50	6999	0	8000	48	10419	100	5000	60	8150	0	7000	59	517	0	14000
03.11	54	7183	0	8000	52	10582	0	5000	51	8076	0	7000	50	520	0	14000
04.11																
05.11																
06.11	37	6802	150	8000	57	10109	0	5000	56	7731	0	7000	54	513	0	14000

## Продовження таблиці 2.3

Дата	9:00				12:00				15:00				18:00			
	H	L	w	f	H	L	w	f	H	L	w	f	H	L	w	f
07.11	49	6784	50	8000	57	10026	0	5000	55	7696	0	7000	54	508	0	14000
08.11	50	6850	0	8000	48	10037	80	5000	59	7708	0	7000	58	519	0	14000
09.11	55	6897	0	8000	54	10569	0	5000	53	7809	0	7000	52	510	0	14000
10.11	49	6738	100	8000	60	10393	0	5000	59	7226	0	8000	58	512	0	14000
11.11																
12.11																
13.11	41	6852	100	8000	52	10199	0	5000	51	6741	0	8000	50	509	0	14000
14.11	47	6730	100	8000	59	10736	0	5000	58	7002	0	8000	56	518	0	14000
15.11	53	6747	0	8000	52	10833	0	5000	51	7264	0	8000	50	506	0	14000
16.11	46	6863	100	8000	57	10467	0	5000	55	7201	0	8000	54	508	0	14000
17.11	51	6779	0	8000	50	10509	0	5000	49	6704	100	8000	60	514	0	14000
18.11																
19.11																
20.11	45	6707	100	8000	54	10531	0	5000	53	6459	0	8000	52	511	0	14000
21.11	48	6818	100	8000	59	10528	0	5000	58	6528	0	8000	57	519	0	14000
22.11	54	6810	0	8000	53	10219	0	5000	52	6817	0	8000	51	514	0	14000
23.11	47	6741	100	8000	58	10074	0	5000	57	6393	0	8000	56	513	0	14000
24.11	52	6758	0	8000	51	10885	0	5000	50	6303	0	8000	49	512	100	14000
25.11																
26.11																
27.11	43	6704	100	8000	53	9327	0	6000	52	6174	0	9000	51	514	0	14000
28.11	47	6792	100	8000	57	9507	0	6000	56	5651	0	9000	55	516	0	14000
29.11	52	6686	0	8000	51	9596	0	6000	50	5644	0	9000	49	501	100	14000
30.11	43	6602	150	8000	59	9727	0	6000	58	6348	0	9000	57	503	0	14000
01.12	54	6601	0	8000	53	9963	0	6000	52	5745	0	9000	51	503	0	14000
02.12																
03.12																
04.12	36	6514	150	8000	55	9406	0	6000	55	6125	0	9000	54	516	0	14000
05.12	51	6627	0	8000	51	9657	0	6000	50	6073	0	9000	49	508	100	14000
06.12	56	6668	0	8000	55	9280	0	6000	54	6481	0	8000	53	506	0	14000
07.12	50	6580	0	8000	49	9776	100	6000	58	6134	0	9000	58	515	0	14000
08.12	54	6524	0	8000	53	9211	0	6000	52	6376	0	8000	52	513	0	14000
09.12																
10.12																
11.12	36	6519	150	8000	56	9225	0	6000	56	5245	0	9000	55	505	0	14000
12.12	52	6455	0	8000	51	9193	0	6000	51	5339	0	9000	50	512	100	14000
13.12	57	6528	0	8000	57	9470	0	6000	56	5843	0	9000	55	515	0	14000
14.12	52	6440	0	8000	51	9185	0	6000	50	5412	0	9000	50	513	100	14000
15.12	56	6410	0	8000	56	9293	0	6000	55	5234	0	9000	54	515	0	14000
16.12																
17.12																
18.12	38	6475	150	8000	59	9569	0	6000	58	5639	0	9000	58	509	0	14000
19.12	54	6377	0	8000	53	9603	0	6000	52	5194	0	9000	52	509	0	14000

## Продовження таблиці 2.3

Дата	9:00				12:00				15:00				18:00			
	H	L	w	f	H	L	w	f	H	L	w	f	H	L	w	f
20.12	49	6478	100	8000	59	9977	0	6000	59	5084	0	9000	58	506	0	14000
21.12	54	6407	0	8000	53	9389	0	6000	52	5311	0	9000	51	517	0	14000
22.12	47	6434	100	8000	56	9818	0	6000	55	5270	0	9000	54	512	0	14000

Більшу частину часу протягом збору даних значення вологості ґрунту (H) вдавалось підтримувати у нормі (50-60%). Найнижче значення вологості ґрунту 36% було зафіксовано 2 рази, найбільше 65% – 1 раз. Кількість води, що використовувалась для поливу становила переважно 100 мл. Якщо ґрунт був дуже сухим (40% і менше) використовувалось 150 мл. Іноді для не дуже сухого ґрунту (~50%) використовувалось 50 мл води.

Отримання показника рівня освітленості здійснювалось при вимкненій фітострічці. Рівень освітлення фітострічкою змінювався від 5000 до 14000 люксів. Найменше фітострічка використовувалась вдень з 12:00 до 15:00, найбільше – ввечері з 18:00 до 20:00. Використання фітострічки дозволило забезпечувати необхідний рівень освітлення для рослини протягом 12 годин.

Отримані дані представлені у графіках на рисунках 2.3 та 2.4 і будуть використані для розробки математичних моделей.

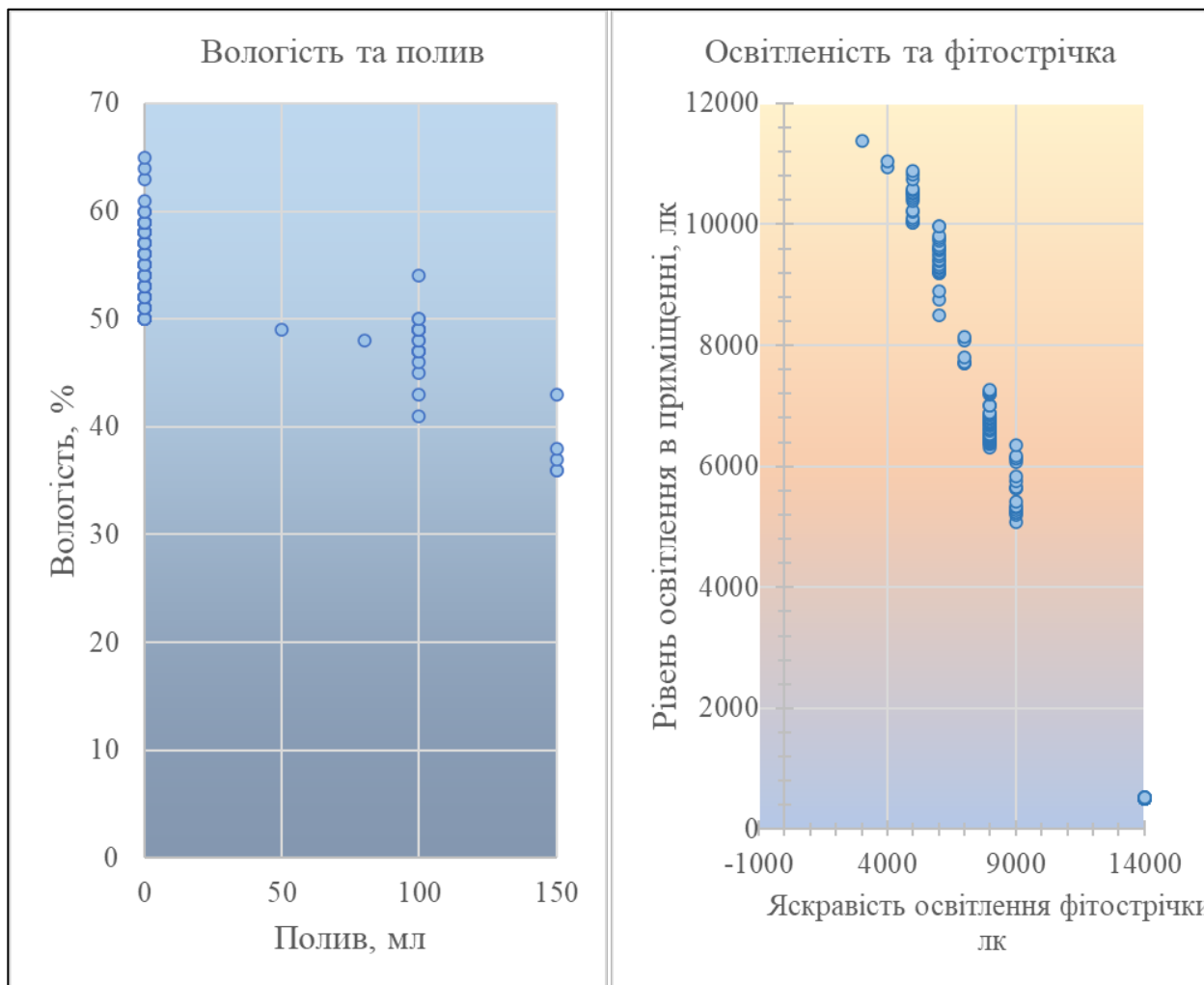


Рисунок 2.3 – Графік кореляції вологості ґрунту та поливу

Рисунок 2.4 – Графік кореляції рівня зовнішнього освітлення та рівня освітлення фітострічки

## 2.5. Визначення критеріїв оцінки ефективності математичних моделей.

Аналіз математичних моделей буде здійснюватися за допомогою оцінки їхньої ефективності у виконанні поставленої задачі. Для цього для кожної моделі буде визначено наступні критерії:

- 1) простота – наскільки модель проста у реалізації та використанні;
- 2) точність – наскільки близькими є результати роботи моделі до очікуваних та наскільки модель стійка до змін;
- 3) витрати ресурсів – наскільки затратним є використання моделі.

Для оцінки точності моделей будуть використовуватись наступні метрики:

- 1) коефіцієнти детермінації та кореляції – вони дозволяють визначити ступінь відповідності між прогнозованими та спостережуваними значеннями;
- 2) функція втрат (Loss Function) середньоквадратична помилка – вона дозволить виміряти середню квадратичну різницю між прогнозованими та фактичними значеннями, що є ключовим показником точності моделі.

Перевірка витрат ресурсів включатиме використання наступних ресурсів:

- 1) час запуску – скільки часу необхідно програмі для того щоб завантажитись та почати робити розрахунки;
- 2) час виконання – як швидко модель зможе виконувати розрахунки;
- 3) використання пам'яті – оцінка обсягу пам'яті, яку модель використовує під час роботи.

## Висновки до розділу 2

Серед розглянутих видів та конкретних прикладів математичних моделей було виділено 3 як найкращі варіанти для задачі автоматизації керування природним станом рослини у системі вирощування, а саме математичні рівняння, нечітка логіка та нейронна мережа. Для моделювання та тестування використовуватимуться фактичні дані, зібрані під час ручного керування системою вирощування. Аналіз та порівняння моделей буде здійснюватися за оцінками їхньої простоти, точності та витрат ресурсів.

## РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ

3.1 Проектування математичних моделей для автоматизованої системи вирощування рослин.

Математичні моделі, що розроблюються, мають вирішувати наступні задачі згідно отриманих фактичних даних:

- 1) визначати кількість води (мл), необхідної для поливу, згідно до заданої вологості ґрунту (%);
- 2) визначати рівень освітлення (лк), який треба ввімкнути на фітострічці, згідно до заданого рівня освітлення у приміщенні (лк).

Існує багато видів математичних рівнянь, і щоб обрати те, яке найбільш точно описуватиме процес, що моделюється, варто розробити декілька рівнянь різних видів і порівняти їхні результати з фактичними. Тож для проектування математичних рівнянь розробимо та порівняємо лінійне алгебраїчне рівняння та квадратичне рівняння.

Лінійні рівняння мають наступний загальний вигляд:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b, \quad (3.1)$$

де:

- $a_1, a_2, \dots, a_n$  – коефіцієнти;
- $x_1, x_2, \dots, x_n$  – змінні;
- $b$  – вільний член.

Для початку розробимо лінійне рівняння для автоматизації поливу. Згідно до поставлених задач, ми маємо одну незалежну змінну (вологість ґрунту) та одну залежну змінну (полив), отже ми можемо виразити їхню залежність за допомогою лінійного рівняння з однією змінною

$$w = m * H + b, \quad (3.2)$$

де:

- $w$  – необхідна кількість води для поливу у мл;
- $H$  – вологість ґрунту у %;
- $m$  – нахил лінії, що показує, наскільки змінюється потреба в поливі залежно від вологості ґрунту;
- $b$  – це кут на вісі  $y$ , який показує необхідну кількість в поливі при вологості ґрунту рівній 0.

Таке рівняння буде мати вигляд прямої, а отже знаючи дві її точки ми зможе знайти коефіцієнти. Для знаходження значень коефіцієнтів  $m$  та  $b$  візьмемо 2 пари граничних фактичних значень  $w$  та  $H$  та на їх основі створимо систему з двох лінійних рівнянь.

$$\begin{cases} 0 = m * 60 + b \\ 150 = m * 36 + b \end{cases} \quad (3.3)$$

Цю систему рівнянь можна розв'язати методом заміщення. Для цього виразимо  $b$  через  $m$ .

$$b = -m * 60 \quad (3.4)$$

Підставимо це значення  $b$  у друге рівняння.

$$150 = m * 36 + (-m * 60) \quad (3.5)$$

Розкриємо дужки та спростимо рівняння.

$$150 = 36m - 60m \quad (3.6)$$

$$150 = -24m \quad (3.7)$$

$$24m = -150 \quad (3.8)$$

$$m = -\frac{150}{24} = -6.25 \quad (3.9)$$

Таким чином отримуємо, що  $m = -6.25$ . Тепер знаючи  $m$  ми можемо знайти  $b$ .

$$b = -(-6.25) * 60 \quad (3.10)$$

$$b = 375 \quad (3.11)$$

Отже, лінійне рівняння для знаходження необхідної кількості води для поливу згідно вологості ґрунту матиме наступний вигляд:

$$w = -6.25 * H + 375 \quad (3.12)$$

Тепер розробимо квадратичне рівняння для тієї задачі поливу. У загальному вигляді квадратичне рівняння матиме наступний вигляд:

$$w = a * H^2 + b * H + c, \quad (3.13)$$

де:

–  $w$  – необхідна кількість води для поливу у мл;

- $H$  – вологість ґрунту у %;
- $a, b$  і  $c$  – коефіцієнти, що визначають кривизну та форму квадратичного зв'язку.

Так само використаємо 2 пари фактичних значень і на їхній основі створимо систему з двох квадратичних рівнянь.

$$\begin{cases} 150 = a * 36^2 + b * 36 + c \\ 100 = a * 46^2 + b * 46 + c \end{cases} \quad (3.14)$$

Отримана система рівнянь не матиме точного розв'язку, тож методом підбору знаходимо коефіцієнти  $a, b$  і  $c$ , які б приблизно її задовольняли.

$$a = -0.068, b = 0, c = 250 \quad (3.15)$$

Графік отриманої квадратичної функції представлено на рисунку 3.1.

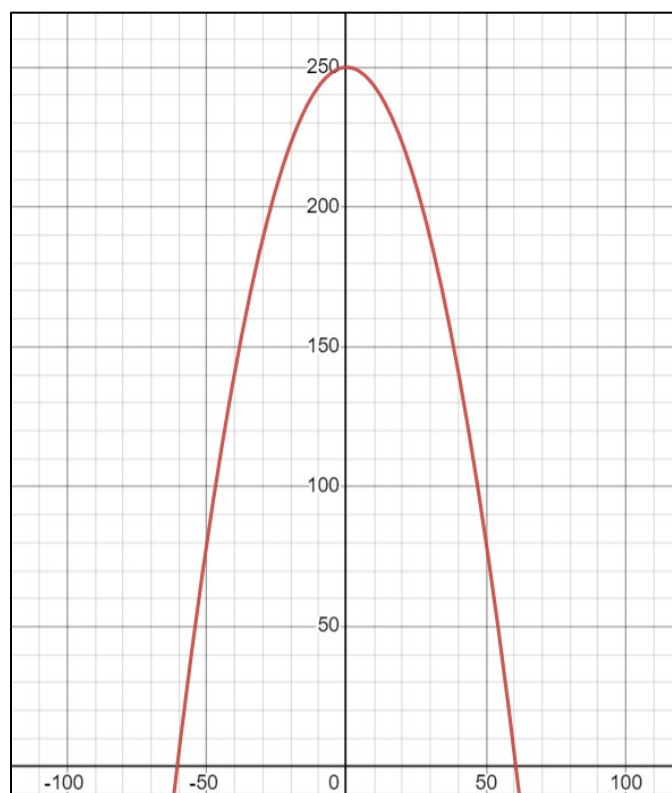


Рисунок 3.1 – Графік квадратичної функції для поливу

Отже, квадратичне рівняння для знаходження необхідної кількості води для поливу згідно вологості ґрунту матиме наступний вигляд:

$$w = -0.068 * H^2 + 250 \quad (3.16)$$

Для підбору математичного рівняння, що найкраще описує взаємозв'язок між вологістю ґрунту та необхідною кількістю поливу необхідно орієнтуватись на реальні експериментальні дані. Перевіримо отримані математичні рівняння з реальними даними (рисунок 3.2).

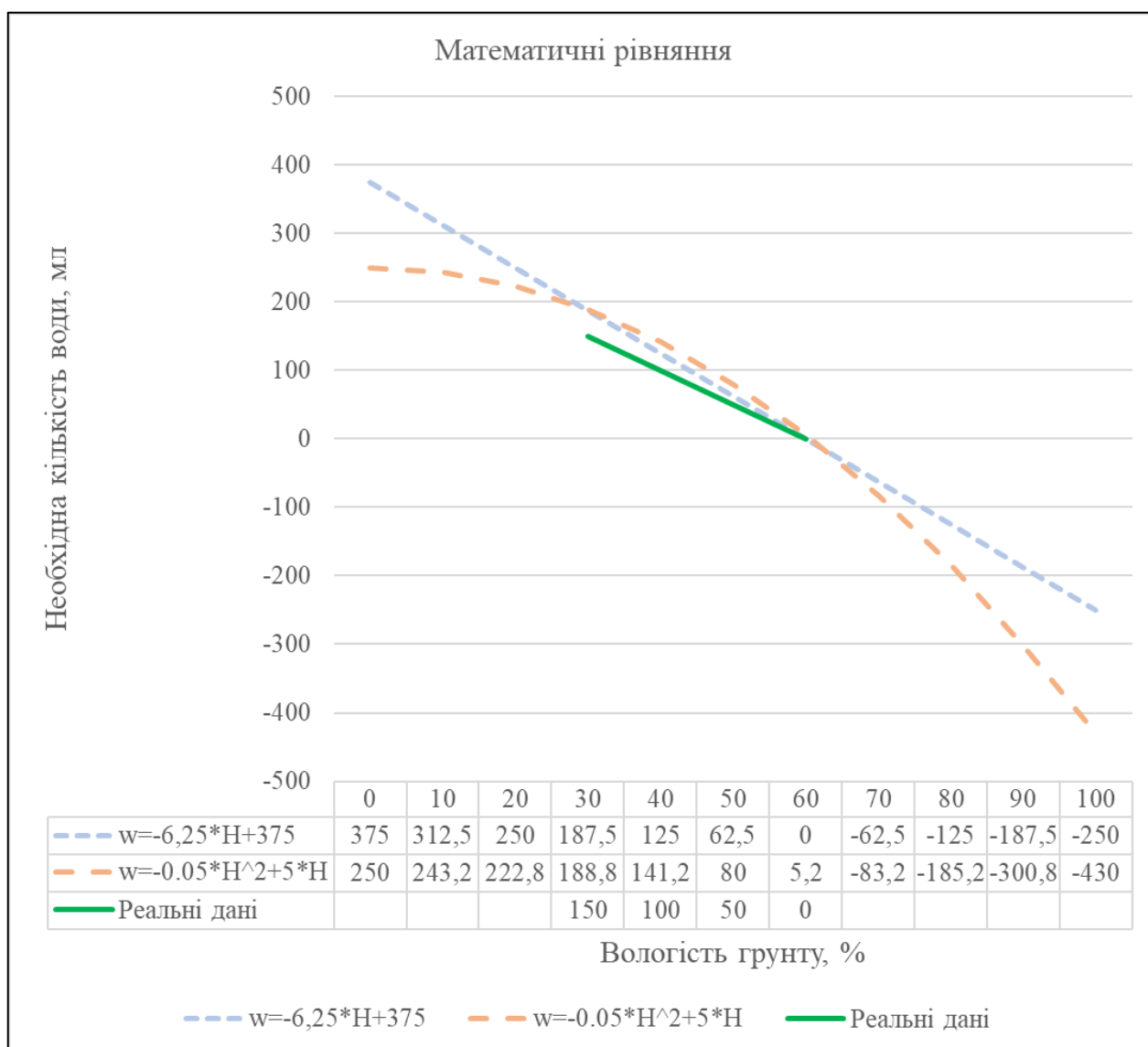


Рисунок 3.2 – Порівняння графіків математичних рівнянь для поливу

Зіставивши графіки обох рівнянь та порівнявши їх з реальними даними

можна помітити, що і лінійне рівняння, і квадратичне рівняння дають наближені результати з похибкою від 5,2 до 41,2 одиниць. Оскільки лінійне рівняння дає більш наближені до тестових результати, було вирішено обрати саме його для програмної реалізації.

Враховуючи особливості датчиків освітлення, залежність необхідного рівня освітлення фітострічки від яскравості зовнішнього освітлення можна описати за допомогою лінійного рівняння з однією змінною, оскільки датчики освітлення показують сумарну кількість світла, що на них падає. Для визначення необхідного рівня освітленості, яке необхідно ввімкнути на фітострічці, достатньо від бажаного рівня освітлення відняти рівень освітлення у приміщенні. Таким чином, враховуючи, що досліджуваній рослині необхідно до 15000 лк, маємо наступне лінійне рівняння:

$$f = 15000 - L, \quad (3.17)$$

де:

- $f$  – рівень освітлення у лк, який необхідно ввімкнути на фітострічці;
- $L$  – рівень зовнішнього освітлення у лк.

Порівняємо графік фактичних даних, отриманих під час досліджень, з графіком даними, що розраховані спроектованою функцією (рисунок 3.3). Як бачимо, спроектована функція наближено повторює реальні дані з похибкою від 6 до 1000 лк. Розроблена функція буде використана у програмній реалізації математичної моделі.

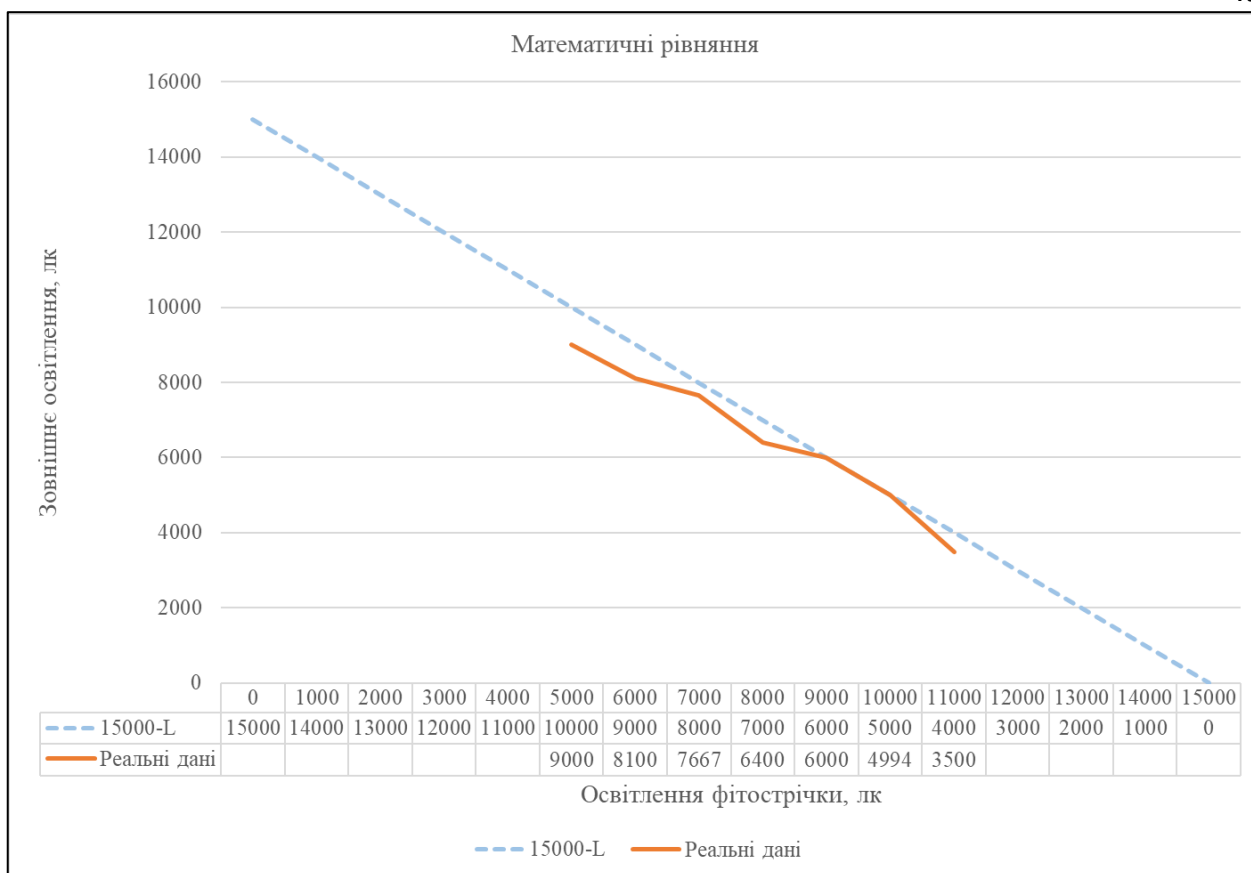


Рисунок 3.3 – Порівняння графіку математичних рівнянь для освітлення

Основні концепції нечіткої логіки відрізняються від класичної булевої логіки, яка працює з чіткими значеннями "правда" або "неправда". В нечіткій логіці використовуються поняття відносних категорій чи ступенів належності до певних груп. Першим кроком визначимо вхідні та вихідні змінні.

Вхідні змінні:

- 1) вологість ґрунту;
- 2) рівень зовнішнього освітлення.

Вихідні змінні:

- 1) необхідна кількість поливу;
- 2) необхідний рівень освітлення фітострічкою.

Далі визначимо лінгвістичні змінні та їхні відповідні значення, орієнтуючись на оптимальні умови для досліджуваної рослини.

Ґрунт:

- 1) дуже сухий – від 0 до 50%;

- 2) сухий – від 40 до 55%;
- 3) оптимальний – від 50 до 60%;
- 4) вологий – від 55 до 100%.

Оптимальною для досліджуваної рослини є вологість ґрунту у межах від 50 до 60%, тому саме таке значення визначено як «нормальне». Від 40 до 50% ґрунт є сухим і потребує помірного поливу, а вологість нижче 40% може бути небезпечною для рослини, тож визначаємо її як «дуже сухий» ґрунт.

Зовнішнє освітлення:

- 1) ніч – від 0 до 5000;
- 2) сутінки – від 0 до 2000;
- 3) схід/захід – від 500 до 7000;
- 4) похмурий день – від 2000 до 10000;
- 5) день – від 5000 до 14000;
- 6) яскравий день – від 8000 до 20000.

Незважаючи на те, що досліджувана рослина є світлолюбною, вона все одно потребує певний час темряви, щоб побути у стані спокою та відновитися, тому варто врахувати «ніч» як окрему лінгвістичну змінну. Сутінки визначають рівень освітлення, який може бути створений у приміщенні за допомогою штучних джерел світла (люмінесцентних ламп, світла від моніторів, тощо). Такий рівень освітлення спостерігається зазвичай ввечері, коли природні джерела світла відсутні.

Полив:

- 1) рясний – 75-200 мл;
- 2) помірний – 50-100 мл;
- 3) маленький – 25-75 мл;
- 4) відсутній – 0-50 мл.

Освітлення фітострічки:

- 1) яскраве – 11000-14000 лк;

- 2) нормальне – 7000-14000 лк;
- 3) помірне – 5000-11000 лк;
- 4) низьке – 0-7000 лк;
- 5) відсутнє – 0-1 лк.

Тепер опишемо нечіткі правила, які визначатимуть залежність між лінгвістичними змінними.

Нечіткі правила для вологості та поливу:

- 1) якщо ґрунт «дуже сухий», то полив має бути «рясним»;
- 2) якщо ґрунт «сухий», то полив має бути «помірним»;
- 3) якщо ґрунт «оптимальний», то полив має бути «маленьким»;
- 4) якщо ґрунт «вологий», то полив має бути «відсутнім»;

Нечіткі правила для освітлення та фітострічки:

- 1) якщо зовнішнє освітлення «ніч», то рівень освітлення фітострічки має бути «відсутнім»;
- 2) якщо зовнішнє освітлення «сутінки», то рівень освітлення фітострічки має бути «яскравим»;
- 3) якщо зовнішнє освітлення «схід/захід», то рівень освітлення фітострічки має бути «нормальним»;
- 4) якщо зовнішнє освітлення «похмурий день», то рівень освітлення фітострічки має бути «помірним»;
- 5) якщо зовнішнє освітлення «день», то рівень освітлення фітострічки має бути «низьким»;
- 6) якщо зовнішнє освітлення «яскравий день», то рівень освітлення фітострічки має бути «відсутнім».

Описані нечіткі правила буде реалізовано програмно.

Нейронна мережа – це математична модель, яка створена на основі роботи нервової системи людини для обробки інформації. Вона складається зі з'єднаних вузлів, які називають нейронами, які обмінюються сигналами між

собою. Для поставленої задачі опису залежності поливу та освітлення від показників датчиків у системі вирощування рослин візьмемо просту зв'язану мережу (MLP - Multilayer Perceptron).

MLP (або зв'язана мережа) – це тип нейронної мережі, де кожен нейрон у попередньому шарі з'єднаний з кожним нейроном у наступному шарі, що робить його повністю зв'язаним. Цей тип не має специфічних структур, характерних для згорткових (CNN) або рекурентних (RNN) мереж.

Архітектури мережі складатиметься з трьох Dense шарів:

- 1) перший Dense шар (32 нейрони) матиме активацію 'relu' (повертає 0 для від'ємних значень і саме значення для позитивних значень) і очікуватиме вхідні дані розмірністю (1,), оскільки у нас є тільки одна функція вхідних даних.
- 2) другий Dense шар (64 нейрони) також матиме активацію 'relu' (повертає 0 для від'ємних значень і саме значення для позитивних значень).
- 3) останній Dense шар (1 нейрон) не має активації, оскільки у нас задача регресії (потрібно передбачити числове значення на основі вхідних даних).

У якості оптимізатора використовуватиметься Adam, а для розрахунку функції витрат – середньоквадратична помилка. Це необхідно для компіляції моделі, щоб визначити спосіб оновлення ваг моделі під час навчання з урахуванням втрати та градієнта функції витрат.

Для навчання моделі буде використано фактичні дані, що були описані у пункті 2.4. Модель навчатиметься протягом 100 епох, використовуючи 80% даних для навчання та 20% даних для валідації.

### 3.2 Обґрунтування обраних засобів для реалізації.

Для програмної реалізації спроектований математичних моделей було обрано мову Python.

Python – це високорівнева, інтерпретована мова програмування загального призначення, яка відома своєю простотою та читабельністю коду. Ось деякі ключові особливості мови програмування Python:

- 1) простота та читабельність – Python пропонує простий і читабельний синтаксис, що робить його легким для вивчення та розуміння. Це дозволяє швидко створювати програми, зменшуючи кількість коду порівняно з іншими мовами;
- 2) інтерпретованість – Python – це інтерпретована мова програмування, що означає, що код виконується рядок за рядком. Це дозволяє розробникам швидко виконувати та тестувати код без необхідності компіляції;
- 3) широке використання – Python використовується для різних завдань, таких як веб-розробка, наукові обчислення, штучний інтелект, обробка даних, розробка ігор, автоматизація та багато іншого;
- 4) багата екосистема – мова має велику кількість бібліотек та фреймворків, які розширюють її функціональність. Наприклад, для наукових обчислень – NumPy, для машинного навчання – TensorFlow, PyTorch, для нечіткої логіки – Skfuzzy, тощо;
- 5) крос-платформеність – Python є крос-платформеним, тобто програми, написані на цій мові, можуть запускатися на різних операційних системах (Windows, macOS, Linux) без значних змін. Це також дозволяє розробити програму у одній ОС (наприклад, Windows), а потім запускати програму у іншій (Raspberry OS);
- 6) динамічна типізація – у Python не потрібно явно вказувати типи даних змінних, оскільки він визначає їх автоматично під час виконання програми.

Для програмування було обрано інтегроване середовище розробки PyCharm.

PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. PyCharm надає

потужні інструменти для редагування коду, такі як автодоповнення, рефакторинг, перевірка правильності синтаксису, швидкі переходи між функціями та файлами. Під час написання коду PyCharm надає автоматичну підказку для методів, функцій та змінних, а також доступ до документації Python. У ньому присутні інструменти для налагодження програм, зокрема можливість встановлення точок зупинки, покрокове виконання коду та відстеження значень змінних. PyCharm дозволяє легко створювати та керувати проектами, використовуючи інтегровані системи керування версіями (наприклад, Git), а також інструменти для управління пакетами Python. Є можливість створювати віртуальні середовища для ізоляції проектів та управління їх залежностями. PyCharm підтримує плагіни, які дозволяють розширювати його функціональність залежно від потреб користувача. На додаток до розробки Python, PyCharm також має підтримку веб-технологій, таких як HTML, CSS, JavaScript, що дозволяє розробникам створювати веб-проекти.

Для програмної реалізації нечіткої логіки буде використовуватись бібліотека Skfuzzy.

«scikit-fuzzy» – це бібліотека для обчислення нечітких систем та нечіткої логіки в середовищі Python. Вона базується на бібліотеці `SciPy` та розширює можливості обробки нечітких даних та роботи з нечіткими множинами.

Основні можливості «scikit-fuzzy» включають:

- 1) моделювання нечітких множин – бібліотека дозволяє створювати та оперувати нечіткими множинами, включаючи трикутні, трапецевидні, гаусівські та інші типи функцій належності;
- 2) нечітко-логічні операції – «scikit-fuzzy» надає можливість виконувати операції нечіткої логіки, такі як кон'юнкція, диз'юнкція, виключне "або" та інші;
- 3) нечітко-логічні моделі та системи – бібліотека дозволяє створювати нечіткі логічні моделі, які можуть бути використані для моделювання

нечітких управляючих систем (Fuzzy Control Systems) та інших нечітких моделей;

- 4) кластеризація з використанням нечітких методів – «scikit-fuzzy» має інструменти для використання нечітких методів у задачах кластеризації даних;
- 5) візуалізація результатів – бібліотека надає засоби для візуалізації нечітких множин та нечітко-логічних моделей.

Для програмної реалізації нейронної мережі будуть використовуватися бібліотеки TensorFlow та Keras.

«Keras» – це високорівневий інтерфейс для побудови та навчання нейронних мереж, що працює поверх бібліотеки глибокого навчання «TensorFlow», «Theano» або «CNTK». Розроблений Франсуа Шоле, Keras спрощує процес створення нейронних мереж, забезпечуючи зручний та інтуїтивний інтерфейс для розробників.

Основні особливості Keras:

- 1) простота використання – Keras надає простий та інтуїтивний API для побудови нейронних мереж. Його код легко зрозуміти, що робить його чудовим вибором для початківців;
- 2) модульність – Keras спроектований на принципі модульності, що дозволяє легко створювати та комбінувати різні шари нейронних мереж для побудови складних моделей;
- 3) підтримка різних архітектур – Keras підтримує створення різних типів мереж, включаючи звичайні шаровані нейронні мережі (MLP), згорткові мережі (CNN), рекурентні мережі (RNN) та багато інших;
- 4) багатошаровість – можливість легко створювати мережі з багатьма шарами та налаштовувати їх параметри;
- 5) підтримка різних фреймворків глибокого навчання – Keras може використовувати TensorFlow, Theano або CNTK як свою базову

бібліотеку глибокого навчання;

- б) широке співтовариство та документація – Keras має активну спільноту користувачів і велику кількість документації та прикладів використання.

Для розрахування коефіцієнту кореляції та середньоквадратичної помилки будуть використовуватись вбудовані функції Microsoft Excel задля точності результатів і уникнення помилок у розрахунках.

Для розрахування використання ресурсів будуть використовуватись наступні функції мови Python:

- 1) `time` з бібліотеки `time` – для визначення часу виконання програми.
- 2) `memory_usage` з бібліотеки `memory_profiler` – для визначення використання пам'яті під час виконання програми.

### 3.3 Розробка програмних застосунків, що реалізують спроектовані математичні моделі.

Для кожної розробленої математичної моделі в рамках проекту, було взято до уваги не лише створення відповідного `.py` файлу, а й забезпечено належну організацію тестування та валідації моделі. У директорії `resource` спеціально введено три піддиректорії: `mathematical_equations`, `fuzzy_logic` та `neural_network` для зберігання файлів з тестами для математичних рівнянь, нечіткої логіки та нейронної мережі відповідно. Кожен `.py` файл тесту був написаний з урахуванням специфіки відповідної математичної моделі, що дозволяє ефективно перевіряти та валідувати її точність та стабільність.

Зручна структура проекту, представлена на рисунку 3.4, дозволяє легко визначати, виокремлювати та розширювати функціонал окремих моделей, а також проводити комплексне тестування всього пакету математичних моделей у системі вирощування рослин.

Файли `mathematical_equations.py`, `fuzzy_logic.py` та `neural_network.py` являють собою виконавчі файли, у яких створюється відповідна математична

модель та виконується прогнозування результатів для заданих вхідних параметрів. Отримані результати виводяться у консоль.

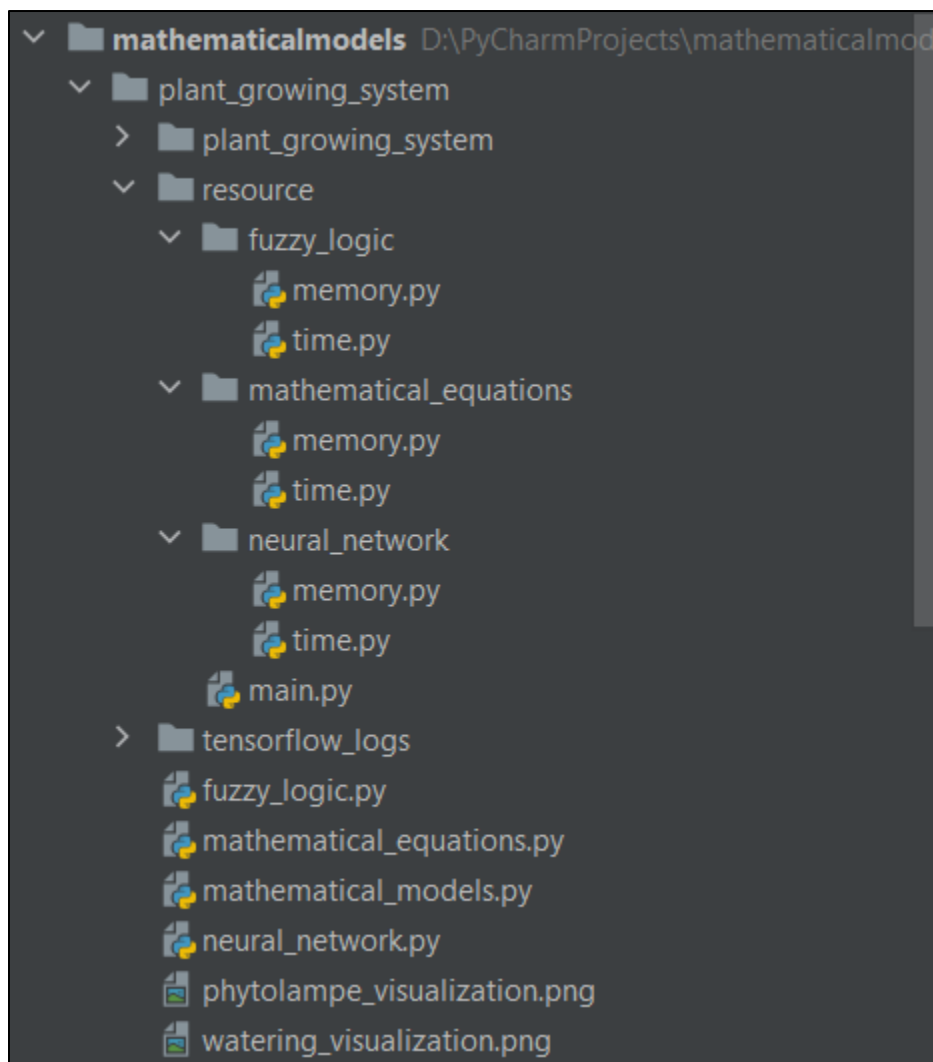
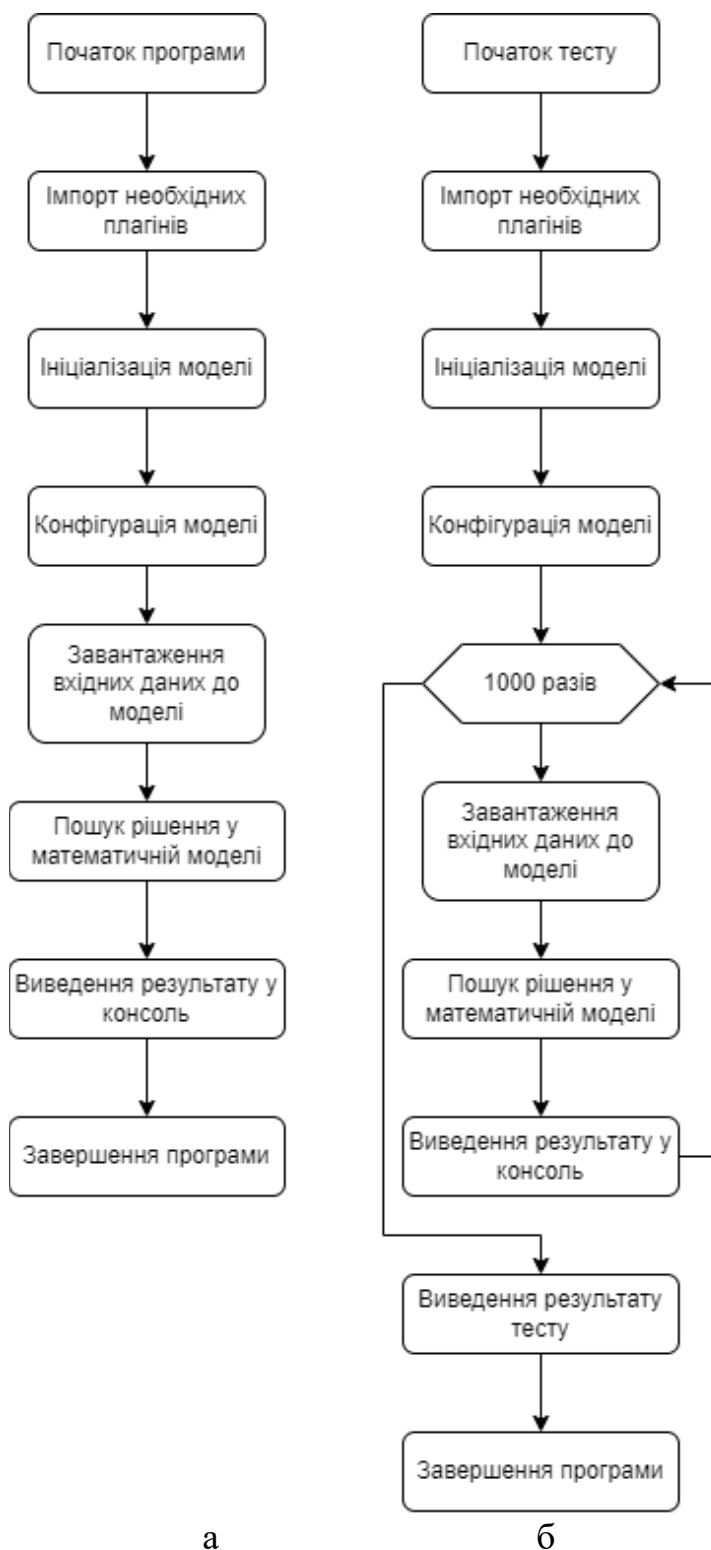


Рисунок 3.4 – Структура проекту

Для зручності тестування функції створення та виконання математичних моделей були зібрані у окремому файлі `mathematical_models`.

Для тестування витрат ресурсів робляться заміри часу та пам'яті під час виконання 1 тисячі запитів до математичних моделей поливу і освітлення. Для імітації передачі отриманих результатів до серверу керування системи вирощування рослин, під час тестування усі отримані результати виводяться у консоль. Блок-схеми виконавчих програм та тестів наведено на рисунку 3.5.

Для знаходження коефіцієнтів кореляції та середньоквадратичної помилки було виконано запуск програм із необхідними вхідними даними.



а – блок-схема виконання програм

б – блок-схема виконання тестів

Рисунок 3.5 – Блок-схеми виконання програм та тестів

### Висновки до розділу 3

На основі фактичних даних було спроектовано, розроблено та відлагоджено визначені математичні моделі. Було проведено заміри показників витрат ресурсів даних моделей, а також зафіксовано їхні результати для подальшого розрахунку коефіцієнтів кореляції та середньоквадратичної помилки.

## РОЗДІЛ 4 ОЦІНКА ЕФЕКТИВНОСТІ

### 4.1 Оцінка моделі математичних рівнянь.

Модель дуже проста у реалізації, програмний код зайняв лише 4 рядки. Ніяких додаткових бібліотек не було використано, адже розроблені рівняння є лінійними з однією змінною, тож не потребували додаткових складних математичних функцій. Для використання програми достатньо запустити виконавчий файл, модель не потребує імпорту даних або додаткової конфігурації. Але модель не враховує множину визначення та значень функції, тому може розраховувати необхідну кількість поливу для вологості більшої 100%, або ж повертати від'ємну кількість мілілітрів, що може викликати неточності у прогнозуванні. Результати розрахунків математичних рівнянь для поливу наведено у таблиці 4.1:

Таблиця 4.1 – Результати роботи математичних рівнянь для поливу

Вологість ґрунту	Середнє значення поливу (реальні дані)	Математичне рівняння
36	150	150
37	150	143,75
38	150	137,5
41	100	118,75
43	125	106,25
45	100	93,75
46	100	87,5
47	100	81,25
48	100	75
49	93,75	68,75
50	15,38	62,5
51	0	56,25
52	0	50
53	0	43,75
54	5,88	37,5
55	0	31,25
56	0	25
57	0	18,75

Продовження таблиці 4.1.

58	0	12,5
59	0	6,25
60	0	0
61	0	-6,25
63	0	-18,75
64	0	-25
65	0	-31,25

Результати розрахунків математичних рівнянь для освітлення наведено у таблиці 4.2:

Таблиця 4.2 – Результати роботи математичних рівнянь для освітлення

Освітлення	Середнє значення освітлення (реальні дані)	Математичне рівняння
0	0	15000
500	14000	14500
5000	9000	10000
5500	9000	9500
6000	8277,777778	9000
6500	8000	8500
7000	8000	8000
7500	7000	7500
8000	6666,666667	7000
8500	6000	6500
9000	6000	6000
9500	6000	5500
10000	5000	5000
10500	4888,888889	4500
11000	3500	4000

Для розрахунку коефіцієнтів точності отримані результати порівнювались із середніми фактичними значеннями. Коефіцієнти точності для поливу наведено у таблиці 4.3:

Таблиця 4.3 – Коефіцієнти точності математичних рівнянь для поливу

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,89	0,80	705

Коефіцієнт кореляції 0,89 близький до 1. Це свідчить про те, що існує високий позитивний лінійний зв'язок між двома змінними. Значення

коефіцієнту детермінації 0,80 означає, що 80% варіації залежної змінної пояснюється варіацією незалежної змінної у моделі, що є гарним показником точності моделі. Значення 705 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Коефіцієнти точності для освітлення наведено у таблиці 4.4:

Таблиця 4.4 – Коефіцієнти точності математичних рівнянь для освітлення

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,99	0,07	147 247

Коефіцієнт кореляції 0,99 вказує на дуже високий позитивний лінійний зв'язок між двома змінними. Значення коефіцієнту детермінації 0,07 означає, що лише 7% варіації залежної змінної можна пояснити варіацією незалежної змінної у моделі, що вказує на її низьку точність. Значення 147247 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Середні витрати ресурсів моделі математичних рівнянь наведено у таблиці 4.5:

Таблиця 4.5 – Витрати ресурсів моделлю математичних рівнянь

Час запуску	Час виконання	Пам'ять
0 с	0.01 с	24 МБ

На запуск математичної моделі витрачалось 0 секунд, оскільки для виконання програми не завантажувалось додаткових бібліотек чи плагін, а також не відбувалось конфігурації моделі. На пошук рішень на 1000 запитів прогнозування необхідного поливу та 1000 запитів необхідного освітлення із виводом отриманих результатів у консоль витрачено 0.01 секунду. На виконання цих запитів було виділено 24 МБ пам'яті.

#### 4.2 Оцінка моделі нечіткої логіки.

На програмування моделі нечіткої логіки була витрачена значна кількість часу. По-перше необхідно було ознайомитись з бібліотекою, що використовувалась, та її документацією. Також модель нечіткої логіки потребувала значного налаштування та відлагодження.

Результати розрахунків нечіткої логіки для поливу наведено у таблиці 4.6:

Таблиця 4.6 – Результати роботи нечіткої логіки для поливу

Вологість ґрунту	Середнє значення поливу (реальні дані)	Нечітка логіка
36	150	143
37	150	143
38	150	142
41	100	118
43	125	93
45	100	78
46	100	72
47	100	67
48	100	62
49	93,75	56
50	15,38	50
51	0	49
52	0	49
53	0	48
54	5,88	47
55	0	0
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0	0
63	0	0
64	0	0
65	0	0

Результати розрахунків нечіткої логіки для освітлення наведено у таблиці 4.7:

Таблиця 4.7 – Результати роботи нечіткої логіки для освітлення

Освітлення	Середнє значення освітлення (реальні дані)	Нечітка логіка
500	14000	12996
5000	9000	9176
5500	9000	7844
6000	8277,777778	6847
6500	8000	6067
7000	8000	5375
7500	7000	5057
8000	6666,666667	4787
8500	6000	4611
9000	6000	4157
9500	6000	3698
10000	5000	3083
10500	4888,888889	3122
11000	3500	3166

Для розрахунку коефіцієнтів точності отримані результати порівнювались із середніми фактичними значеннями. Коефіцієнти точності для поливу наведено у таблиці 4.8:

Таблиця 4.8 – Коефіцієнти точності нечіткої логіки для поливу

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,89	0,80	669

Коефіцієнт кореляції 0,89 близький до 1. Це свідчить про те, що існує високий позитивний лінійний зв'язок між двома змінними. Значення коефіцієнту детермінації 0,80 означає, що 80% варіації залежної змінної пояснюється варіацією незалежної змінної у моделі, що є гарним показником точності моделі. Значення 669 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Коефіцієнти точності для освітлення наведено у таблиці 4.9:

Таблиця 4.9 – Коефіцієнти точності нечіткої логіки для освітлення

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,96	0,92	1 662 144

Коефіцієнт кореляції 0,96 близький до 1. Це свідчить про те, що існує високий позитивний лінійний зв'язок між двома змінними. Значення коефіцієнту детермінації 0,92 означає, що 92% варіації залежної змінної пояснюється варіацією незалежної змінної у моделі, що є гарним показником точності моделі. Значення 1662144 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Середні витрати ресурсів моделі математичних рівнянь наведено у таблиці 4.10:

Таблиця 4.10 – Витрати ресурсів моделлю нечіткої логіки

Час запуску	Час виконання	Пам'ять
0.007 с	9.5 с	102 МБ

На запуск математичної моделі витрачалось 0,007 секунд. На пошук рішень на 1000 запитів прогнозування необхідного поливу та 1000 запитів необхідного освітлення із виводом отриманих результатів у консоль витрачено 9,5 секунд. На виконання цих запитів було виділено 102 МБ пам'яті.

### 4.3 Оцінка моделі нейронної мережі

На програмування моделі нейронної мережі була витрачена значна кількість часу. По-перше необхідно було ознайомитись з бібліотекою, що використовувалась, та її документацією, по-друге необхідно було підготувати масиви з даними для навчання мережі. Також модель нейронної мережі потребувала значного налаштування та відлагодження. Під час тестування була помічена велика неточність між очікуваними даними та результатами, що

повертала нейронна мережа. Причиною цього виявилась недостатня кількість початкових даних для тренування моделі. При збільшенні розміру вибірки з 160 пар даних до 1120 результати роботи нейронної мережі значно покращились, тож для нейронної мережі будуть наведені результати оцінки для різних обсягів навчальних даних.

Результати розрахунків нейронної мережі із 160 парами навчальних даних для поливу наведено у таблиці 4.11:

Таблиця 4.11 – Результати роботи нейронної мережі (160) для поливу

Вологість ґрунту	Середнє значення поливу (реальні дані)	Нейронна мережа
36	150	12
37	150	12
38	150	12
41	100	13
43	125	13
45	100	14
46	100	14
47	100	14
48	100	15
49	93,75	15
50	15,38	15
51	0	15
52	0	16
53	0	16
54	5,88	16
55	0	16
56	0	17
57	0	17
58	0	17
59	0	17
60	0	17
61	0	18
63	0	18
64	0	18
65	0	19

Результати розрахунків нейронної мережі із 1120 парами навчальних даних для поливу наведено у таблиці 4.12:

Таблиця 4.12 – Результати роботи нейронної мережі (1120) для поливу

Вологість ґрунту	Середнє значення поливу (реальні дані)	Нейронна мережа
36	150	174
37	150	166
38	150	158
41	100	134
43	125	119
45	100	103
46	100	95
47	100	88
48	100	79
49	93,75	63
50	15,38	47
51	0	31
52	0	16
53	0	7
54	5,88	6
55	0	4
56	0	3
57	0	1
58	0	0
59	0	-1
60	0	-2
61	0	-4
63	0	-6
64	0	-8
65	0	-9

Результати розрахунків нейронної мережі із 160 парами навчальних даних для освітлення наведено у таблиці 4.13:

Таблиця 4.13 – Результати роботи нейронної мережі (160) для освітлення

Освітлення	Середнє значення освітлення (реальні дані)	Нейронна мережа
500	14000	452
5000	9000	4415
5500	9000	4855

Продовження таблиці 4.13

Освітлення	Середнє значення освітлення (реальні дані)	Нейронна мережа
6000	8277,777778	5296
6500	8000	5746
7000	8000	6176
7500	7000	6617
8000	6666,666667	7057
8500	6000	7497
9000	6000	7938
9500	6000	8378
10000	5000	8818
10500	4888,888889	9259
11000	3500	9699

Результати розрахунків нейронної мережі із 1120 парами навчальних даних для освітлення наведено у таблиці 4.14:

Таблиця 4.14 – Результати роботи нейронної мережі (1120) для освітлення

Освітлення	Середнє значення освітлення (реальні дані)	Нейронна мережа (1120)
500	14000	12482
5000	9000	8204
5500	9000	7728
6000	8277,777778	7252
6500	8000	6776
7000	8000	6300
7500	7000	5824
8000	6666,666667	5348
8500	6000	4872
9000	6000	4581
9500	6000	4360
10000	5000	4138
10500	4888,888889	3951
11000	3500	3791

Для розрахунку коефіцієнтів точності отримані результати порівнювались із середніми фактичними значеннями. Коефіцієнти точності нейронної мережі із 160 парами навчальних даних для поливу наведено у таблиці 4.15:

Таблиця 4.15 – Коефіцієнти точності нейронної мережі (160) для поливу

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
-0,89	0,8	4669

Коефіцієнт кореляції -0,89 – це високий від'ємний коефіцієнт кореляції. Від'ємне значення свідчить про те, що існує обернений (від'ємний) лінійний зв'язок між двома змінними. Значення коефіцієнту детермінації 0,80 означає, що 80% варіації залежної змінної пояснюється варіацією незалежної змінної у моделі, що є гарним показником точності моделі. Значення 4669 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Коефіцієнти точності нейронної мережі із 1120 парами навчальних даних для поливу наведено у таблиці 4.16:

Таблиця 4.16 – Коефіцієнти точності нейронної мережі (1120) для поливу

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,96	0,93	245

Значення коефіцієнту кореляції 0,96 свідчить про дуже сильний лінійний позитивний зв'язок між двома змінними. Значення коефіцієнту детермінації 0,93 означає, що 93% варіації залежної змінної пояснюється варіацією незалежної змінної у моделі, що є гарним показником точності моделі. Значення 245 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Коефіцієнти точності нейронної мережі із 160 парами навчальних даних для освітлення наведено у таблиці 4.17:

Таблиця 4.17 – Коефіцієнти точності нейронної мережі (160) для освітлення

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
-0,99	0,07	13 462 642

-0,99 – це дуже високий від'ємний коефіцієнт кореляції. Високе від'ємне значення свідчить про сильний обернений (від'ємний) лінійний зв'язок між двома змінними. 0,07 – це низький коефіцієнт детермінації, який означає, що лише 7% варіації залежної змінної можна пояснити варіацією незалежної змінної в рамках моделі. Значення 13462642 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Коефіцієнти точності нейронної мережі із 1120 парами навчальних даних для освітлення наведено у таблиці 4.18:

Таблиця 4.18 – Коефіцієнти точності нейронної мережі (1120) для освітлення

Коефіцієнт кореляції	Коефіцієнт детермінації	Середньоквадратична помилка
0,98	0,09	866 711

Значення коефіцієнту кореляції 0,98 свідчить про дуже сильний лінійний позитивний зв'язок між двома змінними. 0,09 – це низький коефіцієнт детермінації, який означає, що лише 7% варіації залежної змінної можна пояснити варіацією незалежної змінної в рамках моделі. Значення 866711 вказує на середню квадратичну помилку між прогнозованими значеннями та спостережуваними значеннями. Чим менше значення MSE, тим краще модель передбачає дані, однак важливо порівняти MSE з іншими моделями.

Середні витрати ресурсів моделі нейронної мережі наведено у таблиці 4.19:

Таблиця 4.19 – Витрати ресурсів моделлю нейронної мережі

Час запуску	Час виконання	Пам'ять
15,6 с	122,6 с	360 МБ

На запуск математичної моделі витрачалось 15,6 секунд. На пошук рішень на 1000 запитів прогнозування необхідного поливу та 1000 запитів необхідного освітлення із виводом отриманих результатів у консоль витрачено 122,6 секунд. На виконання цих запитів було виділено 360 МБ пам'яті.

#### Висновки до розділу 4

Кожна розроблена математична модель була оцінена за критеріями простоти, точності за витрат ресурсів. Результати оцінювання наведені у відповідних таблицях.

## РОЗДІЛ 5 ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

### 5.1 Узагальнення отриманих результатів.

З усіх розроблених математичних моделей найпростішою у реалізації та використанні була модель математичних рівнянь. На відміну від інших моделей, вона не потребувала попереднього вивчення додаткових бібліотек та їхніх документацій. Також для її використання не потрібно було попередньо завантажувати початкові дані.

Порівняємо графіки результатів роботи математичних моделей для визначення необхідного поливу згідно заданої вологості ґрунту (рисунок 5.1).

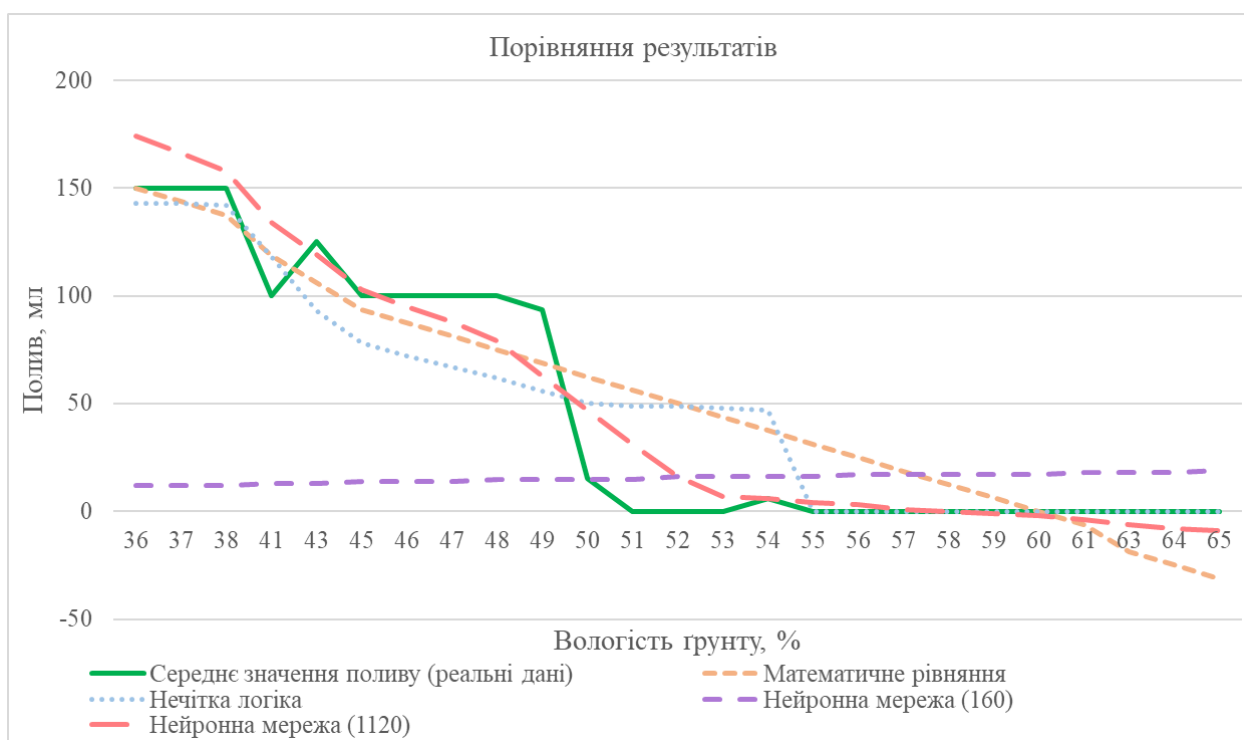


Рисунок 5.1 – Графіки результатів роботи математичних моделей для визначення необхідного поливу

Зеленою непереривною лінією помічено фактичні дані, отримані під час збору даних. Найбільш наближеними до форми графіку фактичних є графіки нечіткої логіки та нейронної мережі із 1120 парами навчальних даних.

Порівняємо графіки результатів роботи математичних моделей для визначення необхідного рівня освітлення фітострічки згідно рівня зовнішнього освітлення (рисунок 5.2).

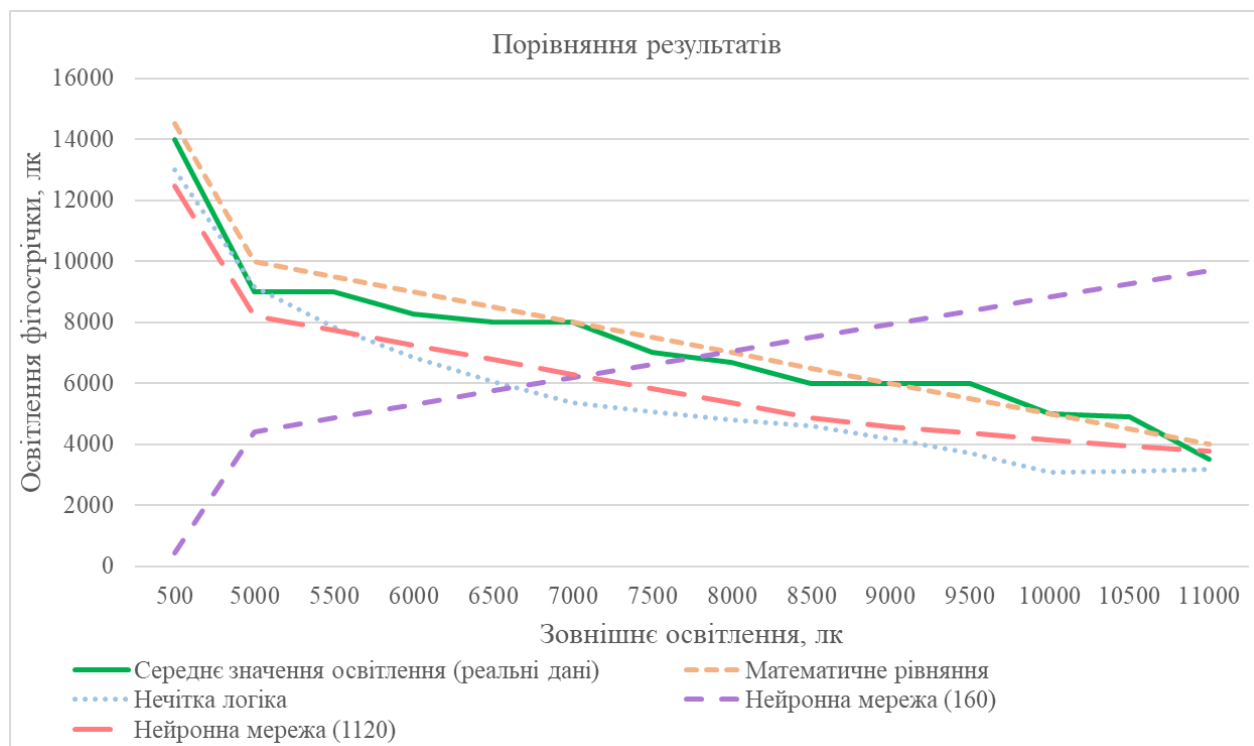


Рисунок 5.2 – Графіки результатів роботи математичних моделей для визначення необхідного освітлення

Найбільш наближеними до форми графіку фактичних є графіки нечіткої логіки та нейронної мережі із 1120 парами навчальних даних.

Порівняємо отримані коефіцієнти точності для усіх розроблених моделей (таблиця 5.1). Найкращі показники виділено жирним шрифтом.

Значення коефіцієнта кореляції 0,89 вказує на досить сильну позитивну лінійну залежність між двома змінними. Коли коефіцієнт кореляції наближається до 1, це свідчить про те, що змінні мають сильну позитивну лінійну зв'язок: коли одна змінна збільшується, інша також схильна збільшуватися пропорційно.

Таблиця 5.1 – Порівняння коефіцієнтів точності для розроблених математичних моделей

Показник \ Модель	Математичні рівняння	Нечітка логіка	Нейронна мережа (160)	Нейронна мережа (1120)
Коефіцієнт кореляції поливу	0,89	0,89	-0,89	<b>0,96</b>
Коефіцієнт кореляції фітострічки	<b>0,99</b>	0,96	-0,99	0,98
Коефіцієнт детермінації поливу	0,80	0,80	0,79	<b>0,93</b>
Коефіцієнт детермінації фітострічки	0,07	<b>0,92</b>	0,07	0,09
Середньоквадратична помилка поливу	705	669	4669	<b>245</b>
Середньоквадратична помилка фітострічки	<b>147 247</b>	1 662 144	13 462 642	866 711

Точне значення "великості" MSE може варіюватися в залежності від контексту конкретної задачі, тому важливо порівнювати MSE між різними моделями або використовувати його для порівняння різних параметрів моделі в рамках однієї задачі. Зазвичай, низьке значення MSE свідчить про кращу точність моделі, а велике значення - про менш точні прогнози.

Серед значень коефіцієнтів кореляції для поливу найкращим є коефіцієнт нейронної мережі із 1120 парами навчальних даних – 0,96. Найкращий коефіцієнт кореляції для фітострічки у математичних рівнянь – 0,99. Найближчий до 1 коефіцієнт детермінації для поливу у нейронної мережі із 1120 парами навчальних даних – 0,93. Для фітострічки найкращий коефіцієнт детермінації у нечіткої логіки – 0,92. Найменша середньоквадратична помилка для поливу у нейронної мережі з 1120 парами навчальних даних – 245. Найменша середньоквадратична помилка для фітострічки у математичних рівнянь – 146 247. Отже ми можемо зробити висновок, що найбільш точними були моделі математичних рівнянь та нейронної мережі із 1120 парами навчальних даних, при чому математичні рівняння були більш точними у

визначенні необхідного рівня освітлення фітострічки, а нейронна мережа була більш точною для керування поливом. Найменш точною виявилась нейронна мережа з 160 парами навчальних даних.

Порівняємо отримані коефіцієнти витрат ресурсів для усіх розроблених моделей (таблиця 5.2). Найкращі показники виділено жирним шрифтом.

Таблиця 5.2 – Порівняння коефіцієнтів витрат ресурсів для розроблених математичних моделей

Показник \ Модель	Математичні рівняння	Нечітка логіка	Нейронна мережа
Час запуску	<b>0 с</b>	0,07 с	15,6 с
Час виконання	<b>0,01 с</b>	9,5 с	122,6 с
Використана пам'ять	<b>24 МБ</b>	102 МБ	360 МБ

Найкращі показники витрат ресурсів отримано для математичних рівнянь. Вони виявились у 950 разів швидшими за нечітку логіку та у 12 260 разів швидшими за нейронну мережу. Це також єдина модель, яка не витратила час на запуск програми. Найбільш ресурсів витратила нейронна мережа.

Таким чином, можемо зробити висновок, що найкращою математичною моделлю для вирішення поставлених задач є модель реалізована за допомогою математичних рівнянь.

## 5.2 Формулювання можливих напрямків для подальших наукових досліджень у даній області.

Не дивлячись на те, що модель математичних рівнянь була кращою за інші моделі по переважній кількості показників, її не можна назвати найбільш ефективною у вирішенні подібних задач. Слід врахувати, що поставлені задачі є доволі простими, особливо у контексті моделювання підтримки природного стану рослини. Як було зазначено у розділі 1, на природний стан рослин впливає безліч факторів і більш того, ці фактори можуть впливати один на одного. Для подальшого покращення існуючої системи вирощування рослин

варто розглянути можливість більш комплексного підходу до керування природним станом, а саме:

- 1) враховувати температуру і вологість повітря;
- 2) враховувати склад та рівень рН ґрунту;
- 3) враховувати вік, розмір та етап життєвого циклу рослини;
- 4) враховувати пору року та час доби.

Точно описати залежність усіх цих факторів за допомогою математичних рівнянь може бути дуже складно чи навіть неможливо, тож для подальшого розвитку існуючої системи вирощування рослин варто розглянути інші види математичних моделей, а саме нейронну мережу. Нейронна мережа дозволить знаходити доволі точні результати на основі декількох вхідних змінних, що значно полегшить задачу пошуку залежностей між факторами. Але слід також врахувати те, що нейронна мережа потребує великої кількості початкових даних, на яких вона зможе навчатись, а також вона використовує значно більше ресурсів, ніж математичні моделі, тож для її розробки варто попередньо забезпечити систему вирощування рослин потужним обладнанням.

#### Висновки до розділу 5

Найкращою математичною моделлю для поставлених задач визначення необхідної кількості поливу згідно вологості ґрунту та необхідного рівня освітлення фітострічки згідно рівня зовнішнього освітлення є математичні рівняння. Однак для більш складних задач, які описуватимуть комплексний зв'язок між декількома факторами та їхній вплив на природний стан рослини, варто розглянути інший вид математичних моделей, а саме нейронну мережу.

## ВИСНОВКИ

Було здійснено дослідження предметної області, дано визначення основним термінам, а також розглянуто фактори, що впливають на природний стан рослини. Було описано існуючу систему поливу та визначено задачі її автоматизації.

Проаналізовано математичні моделі та методології дослідження. Розглянуто різні типи математичних моделей, визначено їхні переваги та обмеження. Обґрунтовано вибір конкретних математичних моделей для аналізу, описано метод збору даних та визначено критерії оцінки ефективності моделей.

Проведено практичну реалізацію розроблених математичних моделей для поставлених задач автоматизації існуючої системи вирощування рослин. Визначені моделі були спроектовані, для них був обґрунтований вибір засобів реалізації.

Виконано оцінку та порівняння ефективності розроблених математичних моделей за критеріями простоти, точності та витрат ресурсів.

За отриманими показниками найкращою моделлю є математичні рівняння, але для подальшого розвитку автоматизованої системи вирощування рослин варто розглянути нейронну мережу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Садовод, І. В. "Основні проблеми та тенденції вирощування кімнатних рослин" [Текст] / І. В. Садовод. – Київ : Видавництво "Зелений світ", 2018. – 120 с.
2. Ліснікова, О. М. "Екологічні аспекти вирощування кімнатних рослин у стільницьких умовах" [Текст] / О. М. Ліснікова. – Харків : Наукове видавництво "Екопростір", 2020. – 185 с.
3. Грунтова, Т. С. "Вплив штучного освітлення на здоров'я кімнатних рослин" [Текст] / Т. С. Грунтова. – Львів : Видавництво "Здорова рослина", 2019. – 90 с.
4. Інститут біології рослин та екології, "Наукові аспекти проблем вирощування кімнатних рослин" [Електронний ресурс]. – Доступно на: <http://www.ibre.org.ua/research/plant-cultivation-issues>. – Дата доступу: 10 січня 2024 р.
5. Журнал "Рослинництво і декоративне садівництво", "Сучасні виклики вирощування кімнатних рослин" [Текст] / Ред. кол.: І. П. Озеров (голова), Т. В. Литвиненко, С. М. Чернявський та ін. – Київ : Видавництво "Флора", 2022. – № 3. – С. 45-56.
6. Сміт, Дж. "Введення в математичне моделювання: засоби, методи, приклади" [Текст] / Дж. Сміт. – Київ : Видавництво "Наука", 2015. – 320 с.
7. Росенблум, М. "Математичні моделі у соціальних науках" [Текст] / М. Росенблум. – Львів : Видавництво ЛНУ ім. Івана Франка, 2018. – 240 с.
8. Айзексон, Г. "Математика та її застосування в природничих науках" [Текст] / Г. Айзексон. – Харків : Видавництво "Основи", 2020. – 176 с.
9. Інститут математики та статистики, "Методологія математичного моделювання в наукових дослідженнях" [Електронний ресурс]. – Доступно на: <http://www.mathinstitute.org/methodology>. – Дата доступу: 15 січня 2024 р.
10. Журнал "Математичні дослідження", "Сучасні підходи до

математичного моделювання в науці" [Текст] / Ред. кол.: О. В. Петров (голова), Н. І. Сидоренко, М. П. Коваленко та ін. – Київ : Видавництво "Математика і наука", 2023. – № 2. – С. 78-91.

11. Джонсон, Р. "Практичні аспекти використання математичних моделей в економіці" [Текст] / Р. Джонсон. – Нью-Йорк : Видавництво "Економічна література", 2016. – 280 с.

12. Сміт, А. "Математичне моделювання в інженерії: від теорії до практики" [Текст] / А. Сміт. – Лондон : Видавництво "Технічна література", 2019. – 320 с.

13. Браун, М. "Реалізація математичних моделей у виробництві: досвід і виклики" [Текст] / М. Браун. – Берлін : Видавництво "Техніка та виробництво", 2020. – 220 с.

14. Інститут досліджень та розвитку, "Математичні моделі в практиці: від ідеї до реалізації" [Електронний ресурс]. – Доступно на: <http://www.instituteofresearch.org/mathematical-models-in-practice>. – Дата доступу: 20 січня 2024 р.

15. Журнал "Прикладне моделювання і обчислення", "Практичні випробування математичних моделей у реальних умовах" [Текст] / Ред. кол.: Л. О. Петров (голова), В. М. Сидоренко, І. П. Коваленко та ін. – Сідней : Видавництво "Прикладне моделювання", 2023. – № 4. – С. 112-125.

16. Андерсон, Д. "Оцінка ефективності математичних моделей в економіці: підходи та методологія" [Текст] / Д. Андерсон. – Нью-Йорк : Видавництво "Економічні дослідження", 2017. – 310 с.

17. Сміт, К. "Метрики для оцінки точності математичних моделей у науці та техніці" [Текст] / К. Сміт. – Лондон : Видавництво "Технічна література", 2020. – 240 с.

18. Янг, Р. "Статистичні методи для оцінки надійності математичних моделей" [Текст] / Р. Янг. – Берлін : Видавництво "Математична статистика", 2018. – 180 с.

19. Інститут оцінки та аналізу, "Методи оцінки ефективності математичних моделей в соціальних науках" [Електронний ресурс]. – Доступно на: <http://www.iae.org.ua/methods-of-evaluation>. – Дата доступу: 25 січня 2024 р.

20. Журнал "Моделювання та аналіз даних", "Сучасні підходи до оцінки ефективності математичних моделей" [Текст] / Ред. кол.: О. І. Петров (голова), Н. В. Сидоренко, М. С. Коваленко та ін. – Київ : Видавництво "Моделі та аналіз", 2023. – № 1. – С. 45-58.

ДОДАТОК А

Технічне завдання

ЗАТВЕРДЖУЮ  
Проректор Українського державного  
університету науки і технологій  
Анатолій РАДКЕВИЧ

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ  
СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.01336 – 01 – ЛЗ

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН  
Керівник розробки  
\_\_\_\_\_Олександр ІВАНОВ  
Виконавець  
\_\_\_\_\_Марина СТЕШЕНКО  
Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО

44165850.01336 – 01

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ  
СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Технічне завдання

Листів 11

2024

**ЗМІСТ**

Вступ.....	3
1. Підстави для розробки.....	4
2. Призначення розробки.....	5
3. Вимоги до програмного продукту.....	6
4. Вимоги до програмної документації.....	8
5. Стадії та етапи розробки.....	9
6. Порядок і контроль приймання.....	10
7. Економічно – технічні показники.....	11

## **ВСТУП**

Перші заводи XVIII ст. викликали індустріальну революцію, замінюючи ручну працю машинами та автоматизуючи виробництво. Електронні помічники тепер знаходяться в кожному будинку, полегшуючи життя та виконуючи різноманітні завдання. Застосування нових технологій є невід'ємною частиною успішних проєктів, але перед кожним апгрейдом важливий аналіз та розрахунки.

Математичні моделі виступають як ефективний інструмент для оцінки, прогнозування та вдосконалення процесів, але важливо розуміти їхні відмінності та відповідність конкретним завданням.

## **1 ПІДСТАВА ДЛЯ РОЗРОБКИ**

Основою для розробки є наказ проректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проєктів» №1196 ст від 05.12. 2022 року.

Тема проєкту: «Аналіз математичних моделей для керування природним станом рослин в автоматизованій системі вирощування».

Керівник дипломного проєкту: Іванов О. П.

## **2 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Функціональне призначення дослідження полягає у вивченні та аналізі математичних моделей для автоматизації існуючої системи вирощування рослин. Необхідно розробити програмний продукт, що впроваджує ці моделі для досягнення ефективності та покращення керування природним станом рослин. Математичні моделі мають враховувати вологість ґрунту та рівень зовнішнього освітлення для визначення необхідної кількості поливу та яскравості освітлення фітострічки.

Експлуатаційне призначення дослідження передбачає впровадження розроблених математичних моделей та програмного продукту у існуючу систему вирощування рослин. Результати нашого дослідження можуть служити основою для вдосконалення сучасних систем вирощування рослин та сприяти розвитку сільськогосподарської та наукових галузей.

### **3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ**

Функціональні вимоги до створення програмного продукту для системи вирощування рослин:

- визначення необхідної кількості поливу (мл) для заданої вологості ґрунту (%);
- визначення необхідного рівня освітлення фітострічки (лк) для заданого рівня зовнішнього освітлення.

#### **4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

До складу документації мають входити:

- текст програми;
- керівництво користувача для користування мобільними додатками.

Вся документація програмного додатку повинна задовольняти вимоги до програмної документації.

**5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ**

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, виріб та обґрунтування критеріїв дослідження.	01.09.23 – 14.09.23
	Попередній вибір методів рішення задач.	14.09.23
	Визначення вимог до технічних засобів.	15.09.23 – 17.09.23
	Узгодження і затвердження технічного завдання.	04.10.23

## **6 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ**

Контроль за виконанням роботи здійснює керівник дипломного проекту: доц. Іванов О. П.

## **7 ЕКОНОМІЧНО – ТЕХНІЧНІ ПОКАЗНИКИ**

Економічно – технічні показники та їх обчислення не були визначені у документах через те, що розробка та дослідження спрямовані на освітні цілі, а не має комерційного характеру.

## ДОДАТОК Б

Технічне завдання

ЗАТВЕРДЖУЮ  
Проректор Українського державного  
університету науки і технологій  
Анатолій РАДКЕВИЧ

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ  
СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.01336 – 01 12 01

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН  
Керівник розробки  
\_\_\_\_\_Олександр ІВАНОВ  
Виконавець  
\_\_\_\_\_Марина СТЕШЕНКО  
Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО

44165850.01336 – 01 12 01

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ  
СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Текст програми

Листів 37

## **АНОТАЦІЯ**

Документ 44165850.01336 – 01 12 01 «Аналіз математичних моделей для керування природним станом рослин в автоматизованій системі вирощування. Текст програми» входить до складу програмної документації на дослідження, мета якого проаналізувати математичні моделі для керування природним станом рослин у автоматизованій системі вирощування.

У даному документі представлений текст програм. Програми написані на мові Python з використанням бібліотек NumPy, skfuzzy, TensorFlow, Keras у програмному середовищі PyCharm.

**3MICT**

1.1 mathematical_equations.py.....	4
1.2 fuzzy_logic.py.....	4
1.3 neural_network.py.....	5
1.4 mathematical_models.py.....	9
1.5 resource/mathematical equations/time.py.....	30
1.6 resource/mathematical equations/memory.py.....	30
1.7 resource/fuzzy logic/time.py.....	31
1.8 resource/fuzzy logic/memory.py.....	31
1.9 resource/neural network/time.py.....	32
1.10 resource/neural network/memory.py.....	33

## 1 ТЕКСТ ПРОГРАМИ

### 1.1 mathematical\_equations.py

```
H = 25
L = 5000
w = -6.25 * H + 375
f = 15000 - L

print("Необхідний полив: " + str(w) + " мл")
print("Необхідне освітлення " + str(f) + " лк")
```

### 1.2 fuzzy\_logic.py

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

# Define the variables
soil_humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'soil humidity')
watering = ctrl.Consequent(np.arange(0, 201, 1), 'watering')
light = ctrl.Antecedent(np.arange(0, 20_001, 1), 'light')
phytolampe = ctrl.Consequent(np.arange(0, 14_001, 1), 'phytolampe')

# Define fuzzy sets for humidity
soil_humidity['very dry'] = fuzz.trimf(soil_humidity.universe, [0, 0, 50])
soil_humidity['dry'] = fuzz.trimf(soil_humidity.universe, [40, 48, 55])
soil_humidity['optimal'] = fuzz.trimf(soil_humidity.universe, [50, 55, 60])
soil_humidity['wet'] = fuzz.trimf(soil_humidity.universe, [55, 100, 100])

# Define fuzzy sets for light
light['night'] = fuzz.trimf(light.universe, [0, 0, 500])
light['twilight'] = fuzz.trimf(light.universe, [0, 400, 2000])
light['sunset/sunrise'] = fuzz.trimf(light.universe, [500, 1000, 7000])
light['cloudy day'] = fuzz.trimf(light.universe, [2000, 5000, 10_000])
light['day'] = fuzz.trimf(light.universe, [5000, 8000, 14_000])
light['bright day'] = fuzz.trimf(light.universe, [8000, 20_000, 20_000])

# Define fuzzy sets for speed
watering['none'] = fuzz.trimf(watering.universe, [0, 0, 25])
watering['low'] = fuzz.trimf(watering.universe, [10, 25, 50])
watering['moderate'] = fuzz.trimf(watering.universe, [40, 70, 100])
watering['heavy'] = fuzz.trimf(watering.universe, [75, 150, 200])

phytolampe['none'] = fuzz.trimf(phytolampe.universe, [0, 0, 1])
phytolampe['low'] = fuzz.trimf(phytolampe.universe, [0, 2000, 7000])
phytolampe['moderate'] = fuzz.trimf(phytolampe.universe, [5000, 9000, 11_000])
phytolampe['normal'] = fuzz.trimf(phytolampe.universe, [7000, 10_000, 14_000])
```

```

phytolampe['bright'] = fuzz.trimf(phytolampe.universe, [11_000, 14_000,
14_000])

# Define fuzzy rules
watering_rule1 = ctrl.Rule(soil_humidity['very dry'], watering['heavy'])
watering_rule2 = ctrl.Rule(soil_humidity['dry'], watering['moderate'])
watering_rule3 = ctrl.Rule(soil_humidity['optimal'], watering['none'])
watering_rule4 = ctrl.Rule(soil_humidity['wet'], watering['none'])

# Define fuzzy rules
phytolampe_rule1 = ctrl.Rule(light['night'], phytolampe['none'])
phytolampe_rule2 = ctrl.Rule(light['twilight'], phytolampe['bright'])
phytolampe_rule3 = ctrl.Rule(light['sunset/sunrise'], phytolampe['normal'])
phytolampe_rule4 = ctrl.Rule(light['cloudy day'], phytolampe['moderate'])
phytolampe_rule5 = ctrl.Rule(light['day'], phytolampe['low'])
phytolampe_rule6 = ctrl.Rule(light['bright day'], phytolampe['none'])

# Create a control system
watering_ctrl = ctrl.ControlSystem([watering_rule1, watering_rule2,
watering_rule3, watering_rule4])
watering_level = ctrl.ControlSystemSimulation(watering_ctrl)

phytolampe_ctrl = ctrl.ControlSystem(
    [phytolampe_rule1, phytolampe_rule2, phytolampe_rule3, phytolampe_rule4,
phytolampe_rule5, phytolampe_rule6])
phytolampe_level = ctrl.ControlSystemSimulation(phytolampe_ctrl)

# Input the parameters (distance and proximity)
watering_level.input['soil humidity'] = 55 # Change here
# Compute the result
watering_level.compute()
# Output the result
print("Необіхдний полив:", watering_level.output['watering'])

phytolampe_level.input['light'] = 5000 # Change here
# Compute the result
phytolampe_level.compute()
# Output the result
print("Необіхдне освітлення:", phytolampe_level.output['phytolampe'])

```

### 1.3 neural\_network.py

```

import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

humidity = np.array([[47], [54], [59], [50], [54],
                    [37], [49], [50], [55], [49],
                    [41], [47], [53], [46], [51],
                    [45], [48], [54], [47], [52],

```

[43], [47], [52], [43], [54],  
 [36], [51], [56], [50], [54],  
 [36], [52], [57], [52], [56],  
 [38], [54], [49], [54], [47],

[61], [65], [57], [48], [52],  
 [57], [57], [48], [54], [60],  
 [52], [59], [52], [57], [50],  
 [54], [59], [53], [58], [51],  
 [53], [57], [51], [59], [53],  
 [55], [51], [55], [49], [53],  
 [56], [51], [57], [51], [56],  
 [59], [53], [59], [53], [56],

[59], [64], [55], [60], [51],  
 [56], [55], [59], [53], [59],  
 [51], [58], [51], [55], [49],  
 [53], [58], [52], [57], [50],  
 [52], [56], [50], [58], [52],  
 [55], [50], [54], [58], [52],  
 [56], [51], [56], [50], [55],  
 [58], [52], [59], [52], [55],

[58], [63], [54], [59], [50],  
 [54], [54], [58], [52], [58],  
 [50], [56], [50], [54], [60],  
 [52], [57], [51], [56], [49],  
 [51], [55], [49], [57], [51],  
 [54], [49], [53], [58], [52],  
 [55], [50], [55], [50], [54],  
 [58], [52], [58], [51], [54]  
 ])

```
light = np.array([[7204], [7251], [7004], [6999], [7183],  

  [6802], [6784], [6850], [6897], [6738],  

  [6852], [6730], [6747], [6863], [6779],  

  [6707], [6818], [6810], [6741], [6758],  

  [6704], [6792], [6686], [6602], [6601],  

  [6514], [6627], [6668], [6580], [6524],  

  [6519], [6455], [6528], [6440], [6410],  

  [6475], [6377], [6478], [6407], [6434],
```

```

  [10935], [11389], [11035], [10419], [10582],  

  [10109], [10026], [10037], [10569], [10393],  

  [10199], [10736], [10833], [10467], [10509],  

  [10531], [10528], [10219], [10074], [10885],  

  [9327], [9507], [9596], [9727], [9963],  

  [9406], [9657], [9280], [9776], [9211],  

  [9225], [9193], [9470], [9185], [9293],  

  [9569], [9603], [9977], [9389], [9818],
```

```

  [8762], [8903], [8498], [8150], [8076],
```

```
[7731], [7696], [7708], [7809], [7226],
[6741], [7002], [7264], [7201], [6704],
[6459], [6528], [6817], [6393], [6303],
[6174], [5651], [5644], [6348], [5745],
[6125], [6073], [6481], [6134], [6376],
[5245], [5339], [5843], [5412], [5234],
[5639], [5194], [5084], [5311], [5270],
```

```
[518], [524], [513], [517], [520],
[513], [508], [519], [510], [512],
[509], [518], [506], [508], [514],
[511], [519], [514], [513], [512],
[514], [516], [501], [503], [503],
[516], [508], [506], [515], [513],
[505], [512], [515], [513], [515],
[509], [509], [506], [517], [512]
])
```

```
watering_needs = np.array([100, 100, 0, 0, 0,
                             150, 50, 0, 0, 100,
                             100, 100, 0, 100, 0,
                             100, 100, 0, 100, 0,
                             100, 100, 0, 150, 0,
                             150, 0, 0, 0, 0,
                             150, 0, 0, 0, 0,
                             150, 0, 100, 0, 100,

                             0, 0, 0, 100, 0,
                             0, 0, 80, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 100, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,

                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 100,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,

                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0,
                             0, 0, 0, 0, 100,
                             0, 0, 100, 0, 0,
                             0, 100, 0, 0, 0,
```

```

        0, 100, 0, 100, 0,
        0, 0, 0, 0, 0
    ])
phytolampe = np.array([8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,

                        4000, 3000, 4000, 5000, 5000,
                        5000, 5000, 5000, 5000, 5000,
                        5000, 5000, 5000, 5000, 5000,
                        5000, 5000, 5000, 5000, 5000,
                        6000, 6000, 6000, 6000, 6000,
                        6000, 6000, 6000, 6000, 6000,
                        6000, 6000, 6000, 6000, 6000,
                        6000, 6000, 6000, 6000, 6000,

                        6000, 6000, 6000, 7000, 7000,
                        7000, 7000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        8000, 8000, 8000, 8000, 8000,
                        9000, 9000, 9000, 9000, 9000,
                        9000, 9000, 8000, 9000, 8000,
                        9000, 9000, 9000, 9000, 9000,
                        9000, 9000, 9000, 9000, 9000,

                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000,
                        14000, 14000, 14000, 14000, 14000
    ])

# Define a neural network model
watering_model = Sequential([
    Dense(32, activation='relu', input_shape=(1,)),
    Dense(64, activation='relu'),
    Dense(1) # No activation for regression task
])

light_model = Sequential([
    Dense(32, activation='relu', input_shape=(1,)),
    Dense(64, activation='relu'),
    Dense(1) # No activation for regression task
])

```

```

])

# Compile the model for regression
watering_model.compile(optimizer='adam', loss='mean_squared_error')
light_model.compile(optimizer='adam', loss='mean_squared_error')

tensorboard_callback = tensorflow.keras.callbacks.TensorBoard(

log_dir="D:/PyCharmProjects/mathematicalmodels/plant_growing_system/tensorflo
w_logs", histogram_freq=1)

# Train the model
watering_model.fit(humidity, watering_needs, epochs=10, batch_size=5,
validation_split=0.2,
                    callbacks=[tensorboard_callback])
light_model.fit(light, phytolampe, epochs=10, batch_size=5,
validation_split=0.2, callbacks=[tensorboard_callback])

humidity_to_predict = np.array([35, 45, 55]) # Replace this with your own
humidity values
light_to_predict = np.array([500, 8000, 11_000]) # Replace this with your
own humidity values

predicted_needs = watering_model.predict(humidity_to_predict)
print("Необхідний полив:")
print(predicted_needs)

predicted_light = light_model.predict(light_to_predict)
print("Необхідне освітлення:")
print(predicted_light)

```

#### 1.4 mathematical\_models.py

```

import skfuzzy as fuzz
from skfuzzy import control as ctrl
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

def mathematical_equations_watering_function(H):
    return -6.25 * H + 375

def mathematical_equations_phytolampe_function(L):
    return 15000 - L

def define_fuzzy_logic_watering_control():
    soil_humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'soil humidity')
    watering = ctrl.Consequent(np.arange(0, 201, 1), 'watering')

```

```

soil_humidity['very dry'] = fuzz.trimf(soil_humidity.universe, [0, 0,
50])
soil_humidity['dry'] = fuzz.trimf(soil_humidity.universe, [40, 48, 55])
soil_humidity['optimal'] = fuzz.trimf(soil_humidity.universe, [50, 55,
60])
soil_humidity['wet'] = fuzz.trimf(soil_humidity.universe, [55, 100, 100])

watering['none'] = fuzz.trimf(watering.universe, [0, 0, 1])
watering['low'] = fuzz.trimf(watering.universe, [0, 20, 50])
watering['moderate'] = fuzz.trimf(watering.universe, [20, 50, 80])
watering['heavy'] = fuzz.trimf(watering.universe, [70, 200, 200])

watering_rule1 = ctrl.Rule(soil_humidity['very dry'], watering['heavy'])
watering_rule2 = ctrl.Rule(soil_humidity['dry'], watering['moderate'])
watering_rule3 = ctrl.Rule(soil_humidity['optimal'], watering['none'])
watering_rule4 = ctrl.Rule(soil_humidity['wet'], watering['none'])

watering_ctrl = ctrl.ControlSystem([watering_rule1, watering_rule2,
watering_rule3, watering_rule4])
return ctrl.ControlSystemSimulation(watering_ctrl)

def define_fuzzy_logic_phytolampe_control():
    light = ctrl.Antecedent(np.arange(0, 20_001, 1), 'light')
    phytolampe = ctrl.Consequent(np.arange(0, 14_001, 1), 'phytolampe')

    light['night'] = fuzz.trimf(light.universe, [0, 0, 500])
    light['twilight'] = fuzz.trimf(light.universe, [0, 400, 2000])
    light['sunset/sunrise'] = fuzz.trimf(light.universe, [500, 1000, 7000])
    light['cloudy day'] = fuzz.trimf(light.universe, [2000, 5000, 10_000])
    light['day'] = fuzz.trimf(light.universe, [5000, 8000, 14_000])
    light['bright day'] = fuzz.trimf(light.universe, [8000, 20_000, 20_000])

    phytolampe['none'] = fuzz.trimf(phytolampe.universe, [0, 0, 1])
    phytolampe['low'] = fuzz.trimf(phytolampe.universe, [0, 2000, 7000])
    phytolampe['moderate'] = fuzz.trimf(phytolampe.universe, [5000, 9000,
11_000])
    phytolampe['normal'] = fuzz.trimf(phytolampe.universe, [7000, 10_000,
14_000])
    phytolampe['bright'] = fuzz.trimf(phytolampe.universe, [11_000, 14_000,
14_000])

    phytolampe_rule1 = ctrl.Rule(light['night'], phytolampe['none'])
    phytolampe_rule2 = ctrl.Rule(light['twilight'], phytolampe['bright'])
    phytolampe_rule3 = ctrl.Rule(light['sunset/sunrise'],
phytolampe['normal'])
    phytolampe_rule4 = ctrl.Rule(light['cloudy day'], phytolampe['moderate'])
    phytolampe_rule5 = ctrl.Rule(light['day'], phytolampe['low'])
    phytolampe_rule6 = ctrl.Rule(light['bright day'], phytolampe['none'])

```



[54], [49], [53], [58], [52],  
[55], [50], [55], [50], [54],  
[58], [52], [58], [51], [54],

# =====  
# [47], [54], [59], [50], [54],  
# [37], [49], [50], [55], [49],  
# [41], [47], [53], [46], [51],  
# [45], [48], [54], [47], [52],  
# [43], [47], [52], [43], [54],  
# [36], [51], [56], [50], [54],  
# [36], [52], [57], [52], [56],  
# [38], [54], [49], [54], [47],  
# [61], [65], [57], [48], [52],  
# [57], [57], [48], [54], [60],  
# [52], [59], [52], [57], [50],  
# [54], [59], [53], [58], [51],  
# [53], [57], [51], [59], [53],  
# [55], [51], [55], [49], [53],  
# [56], [51], [57], [51], [56],  
# [59], [53], [59], [53], [56],  
# [59], [64], [55], [60], [51],  
# [56], [55], [59], [53], [59],  
# [51], [58], [51], [55], [49],  
# [53], [58], [52], [57], [50],  
# [52], [56], [50], [58], [52],  
# [55], [50], [54], [58], [52],  
# [56], [51], [56], [50], [55],  
# [58], [52], [59], [52], [55],  
# [58], [63], [54], [59], [50],  
# [54], [54], [58], [52], [58],  
# [50], [56], [50], [54], [60],  
# [52], [57], [51], [56], [49],  
# [51], [55], [49], [57], [51],  
# [54], [49], [53], [58], [52],  
# [55], [50], [55], [50], [54],  
# [58], [52], [58], [51], [54],  
# [47], [54], [59], [50], [54],  
# [37], [49], [50], [55], [49],  
# [41], [47], [53], [46], [51],  
# [45], [48], [54], [47], [52],  
# [43], [47], [52], [43], [54],  
# [36], [51], [56], [50], [54],  
# [36], [52], [57], [52], [56],  
# [38], [54], [49], [54], [47],  
# [61], [65], [57], [48], [52],  
# [57], [57], [48], [54], [60],  
# [52], [59], [52], [57], [50],  
# [54], [59], [53], [58], [51],  
# [53], [57], [51], [59], [53],  
# [55], [51], [55], [49], [53],

# [56], [51], [57], [51], [56],  
# [59], [53], [59], [53], [56],  
# [59], [64], [55], [60], [51],  
# [56], [55], [59], [53], [59],  
# [51], [58], [51], [55], [49],  
# [53], [58], [52], [57], [50],  
# [52], [56], [50], [58], [52],  
# [55], [50], [54], [58], [52],  
# [56], [51], [56], [50], [55],  
# [58], [52], [59], [52], [55],  
# [58], [63], [54], [59], [50],  
# [54], [54], [58], [52], [58],  
# [50], [56], [50], [54], [60],  
# [52], [57], [51], [56], [49],  
# [51], [55], [49], [57], [51],  
# [54], [49], [53], [58], [52],  
# [55], [50], [55], [50], [54],  
# [58], [52], [58], [51], [54],  
# [47], [54], [59], [50], [54],  
# [37], [49], [50], [55], [49],  
# [41], [47], [53], [46], [51],  
# [45], [48], [54], [47], [52],  
# [43], [47], [52], [43], [54],  
# [36], [51], [56], [50], [54],  
# [36], [52], [57], [52], [56],  
# [38], [54], [49], [54], [47],  
# [61], [65], [57], [48], [52],  
# [57], [57], [48], [54], [60],  
# [52], [59], [52], [57], [50],  
# [54], [59], [53], [58], [51],  
# [53], [57], [51], [59], [53],  
# [55], [51], [55], [49], [53],  
# [56], [51], [57], [51], [56],  
# [59], [53], [59], [53], [56],  
# [59], [64], [55], [60], [51],  
# [56], [55], [59], [53], [59],  
# [51], [58], [51], [55], [49],  
# [53], [58], [52], [57], [50],  
# [52], [56], [50], [58], [52],  
# [55], [50], [54], [58], [52],  
# [56], [51], [56], [50], [55],  
# [58], [52], [59], [52], [55],  
# [58], [63], [54], [59], [50],  
# [54], [54], [58], [52], [58],  
# [50], [56], [50], [54], [60],  
# [52], [57], [51], [56], [49],  
# [51], [55], [49], [57], [51],  
# [54], [49], [53], [58], [52],  
# [55], [50], [55], [50], [54],  
# [58], [52], [58], [51], [54],  
# [47], [54], [59], [50], [54],

# [37], [49], [50], [55], [49],  
# [41], [47], [53], [46], [51],  
# [45], [48], [54], [47], [52],  
# [43], [47], [52], [43], [54],  
# [36], [51], [56], [50], [54],  
# [36], [52], [57], [52], [56],  
# [38], [54], [49], [54], [47],  
# [61], [65], [57], [48], [52],  
# [57], [57], [48], [54], [60],  
# [52], [59], [52], [57], [50],  
# [54], [59], [53], [58], [51],  
# [53], [57], [51], [59], [53],  
# [55], [51], [55], [49], [53],  
# [56], [51], [57], [51], [56],  
# [59], [53], [59], [53], [56],  
# [59], [64], [55], [60], [51],  
# [56], [55], [59], [53], [59],  
# [51], [58], [51], [55], [49],  
# [53], [58], [52], [57], [50],  
# [52], [56], [50], [58], [52],  
# [55], [50], [54], [58], [52],  
# [56], [51], [56], [50], [55],  
# [58], [52], [59], [52], [55],  
# [58], [63], [54], [59], [50],  
# [54], [54], [58], [52], [58],  
# [50], [56], [50], [54], [60],  
# [52], [57], [51], [56], [49],  
# [51], [55], [49], [57], [51],  
# [54], [49], [53], [58], [52],  
# [55], [50], [55], [50], [54],  
# [58], [52], [58], [51], [54],  
# [47], [54], [59], [50], [54],  
# [37], [49], [50], [55], [49],  
# [41], [47], [53], [46], [51],  
# [45], [48], [54], [47], [52],  
# [43], [47], [52], [43], [54],  
# [36], [51], [56], [50], [54],  
# [36], [52], [57], [52], [56],  
# [38], [54], [49], [54], [47],  
# [61], [65], [57], [48], [52],  
# [57], [57], [48], [54], [60],  
# [52], [59], [52], [57], [50],  
# [54], [59], [53], [58], [51],  
# [53], [57], [51], [59], [53],  
# [55], [51], [55], [49], [53],  
# [56], [51], [57], [51], [56],  
# [59], [53], [59], [53], [56],  
# [59], [64], [55], [60], [51],  
# [56], [55], [59], [53], [59],  
# [51], [58], [51], [55], [49],  
# [53], [58], [52], [57], [50],

```

# [52], [56], [50], [58], [52],
# [55], [50], [54], [58], [52],
# [56], [51], [56], [50], [55],
# [58], [52], [59], [52], [55],
# [58], [63], [54], [59], [50],
# [54], [54], [58], [52], [58],
# [50], [56], [50], [54], [60],
# [52], [57], [51], [56], [49],
# [51], [55], [49], [57], [51],
# [54], [49], [53], [58], [52],
# [55], [50], [55], [50], [54],
# [58], [52], [58], [51], [54],
# [47], [54], [59], [50], [54],
# [37], [49], [50], [55], [49],
# [41], [47], [53], [46], [51],
# [45], [48], [54], [47], [52],
# [43], [47], [52], [43], [54],
# [36], [51], [56], [50], [54],
# [36], [52], [57], [52], [56],
# [38], [54], [49], [54], [47],
# [61], [65], [57], [48], [52],
# [57], [57], [48], [54], [60],
# [52], [59], [52], [57], [50],
# [54], [59], [53], [58], [51],
# [53], [57], [51], [59], [53],
# [55], [51], [55], [49], [53],
# [56], [51], [57], [51], [56],
# [59], [53], [59], [53], [56],
# [59], [64], [55], [60], [51],
# [56], [55], [59], [53], [59],
# [51], [58], [51], [55], [49],
# [53], [58], [52], [57], [50],
# [52], [56], [50], [58], [52],
# [55], [50], [54], [58], [52],
# [56], [51], [56], [50], [55],
# [58], [52], [59], [52], [55],
# [58], [63], [54], [59], [50],
# [54], [54], [58], [52], [58],
# [50], [56], [50], [54], [60],
# [52], [57], [51], [56], [49],
# [51], [55], [49], [57], [51],
# [54], [49], [53], [58], [52],
# [55], [50], [55], [50], [54],
# [58], [52], [58], [51], [54]
])
watering_needs = np.array([100, 100, 0, 0, 0,
                             150, 50, 0, 0, 100,
                             100, 100, 0, 100, 0,
                             100, 100, 0, 100, 0,
                             100, 100, 0, 150, 0,
                             150, 0, 0, 0, 0,

```



# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 100,  
# 0, 0, 100, 0, 0,  
# 0, 100, 0, 0, 0,  
# 0, 100, 0, 100, 0,  
# 0, 0, 0, 0, 0,  
# 100, 100, 0, 0, 0,  
# 150, 50, 0, 0, 100,  
# 100, 100, 0, 100, 0,  
# 100, 100, 0, 100, 0,  
# 100, 100, 0, 150, 0,  
# 150, 0, 0, 0, 0,  
# 150, 0, 0, 0, 0,  
# 150, 0, 100, 0, 100,  
# 0, 0, 0, 100, 0,  
# 0, 0, 80, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 100, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 0,  
# 0, 0, 0, 0, 100,  
# 0, 0, 100, 0, 0,  
# 0, 100, 0, 0, 0,  
# 0, 100, 0, 100, 0,  
# 0, 0, 0, 0, 0,  
# 100, 100, 0, 0, 0,  
# 150, 50, 0, 0, 100,  
# 100, 100, 0, 100, 0,  
# 100, 100, 0, 100, 0,  
# 100, 100, 0, 150, 0,  
# 150, 0, 0, 0, 0,  
# 150, 0, 0, 0, 0,





```

# 0, 0, 0, 100, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 100,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 0,
# 0, 0, 0, 0, 100,
# 0, 0, 100, 0, 0,
# 0, 100, 0, 0, 0,
# 0, 100, 0, 100, 0,
# 0, 0, 0, 0, 0
])

model = Sequential([
    Dense(32, activation='relu', input_shape=(1,)),
    Dense(64, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(humidity, watering_needs, epochs=100, batch_size=5,
validation_split=0.2)

return model

def neural_network_watering(model, H_array):
    return model.predict(H_array)

def define_neural_network_light():
    light = np.array([[7204], [7251], [7004], [6999], [7183],
    [6802], [6784], [6850], [6897], [6738],
    [6852], [6730], [6747], [6863], [6779],
    [6707], [6818], [6810], [6741], [6758],
    [6704], [6792], [6686], [6602], [6601],
    [6514], [6627], [6668], [6580], [6524],
    [6519], [6455], [6528], [6440], [6410],
    [6475], [6377], [6478], [6407], [6434],

    [10935], [11389], [11035], [10419], [10582],
    [10109], [10026], [10037], [10569], [10393],
    [10199], [10736], [10833], [10467], [10509],

```

[10531], [10528], [10219], [10074], [10885],  
 [9327], [9507], [9596], [9727], [9963],  
 [9406], [9657], [9280], [9776], [9211],  
 [9225], [9193], [9470], [9185], [9293],  
 [9569], [9603], [9977], [9389], [9818],

[8762], [8903], [8498], [8150], [8076],  
 [7731], [7696], [7708], [7809], [7226],  
 [6741], [7002], [7264], [7201], [6704],  
 [6459], [6528], [6817], [6393], [6303],  
 [6174], [5651], [5644], [6348], [5745],  
 [6125], [6073], [6481], [6134], [6376],  
 [5245], [5339], [5843], [5412], [5234],  
 [5639], [5194], [5084], [5311], [5270],

[518], [524], [513], [517], [520],  
 [513], [508], [519], [510], [512],  
 [509], [518], [506], [508], [514],  
 [511], [519], [514], [513], [512],  
 [514], [516], [501], [503], [503],  
 [516], [508], [506], [515], [513],  
 [505], [512], [515], [513], [515],  
 [509], [509], [506], [517], [512],

#=====

# [7204], [7251], [7004], [6999], [7183],  
 # [6802], [6784], [6850], [6897], [6738],  
 # [6852], [6730], [6747], [6863], [6779],  
 # [6707], [6818], [6810], [6741], [6758],  
 # [6704], [6792], [6686], [6602], [6601],  
 # [6514], [6627], [6668], [6580], [6524],  
 # [6519], [6455], [6528], [6440], [6410],  
 # [6475], [6377], [6478], [6407], [6434],  
 # [10935], [11389], [11035], [10419], [10582],  
 # [10109], [10026], [10037], [10569], [10393],  
 # [10199], [10736], [10833], [10467], [10509],  
 # [10531], [10528], [10219], [10074], [10885],  
 # [9327], [9507], [9596], [9727], [9963],  
 # [9406], [9657], [9280], [9776], [9211],  
 # [9225], [9193], [9470], [9185], [9293],  
 # [9569], [9603], [9977], [9389], [9818],  
 # [8762], [8903], [8498], [8150], [8076],  
 # [7731], [7696], [7708], [7809], [7226],  
 # [6741], [7002], [7264], [7201], [6704],  
 # [6459], [6528], [6817], [6393], [6303],  
 # [6174], [5651], [5644], [6348], [5745],  
 # [6125], [6073], [6481], [6134], [6376],  
 # [5245], [5339], [5843], [5412], [5234],  
 # [5639], [5194], [5084], [5311], [5270],  
 # [518], [524], [513], [517], [520],  
 # [513], [508], [519], [510], [512],

# [509], [518], [506], [508], [514],  
# [511], [519], [514], [513], [512],  
# [514], [516], [501], [503], [503],  
# [516], [508], [506], [515], [513],  
# [505], [512], [515], [513], [515],  
# [509], [509], [506], [517], [512],  
# [7204], [7251], [7004], [6999], [7183],  
# [6802], [6784], [6850], [6897], [6738],  
# [6852], [6730], [6747], [6863], [6779],  
# [6707], [6818], [6810], [6741], [6758],  
# [6704], [6792], [6686], [6602], [6601],  
# [6514], [6627], [6668], [6580], [6524],  
# [6519], [6455], [6528], [6440], [6410],  
# [6475], [6377], [6478], [6407], [6434],  
# [10935], [11389], [11035], [10419], [10582],  
# [10109], [10026], [10037], [10569], [10393],  
# [10199], [10736], [10833], [10467], [10509],  
# [10531], [10528], [10219], [10074], [10885],  
# [9327], [9507], [9596], [9727], [9963],  
# [9406], [9657], [9280], [9776], [9211],  
# [9225], [9193], [9470], [9185], [9293],  
# [9569], [9603], [9977], [9389], [9818],  
# [8762], [8903], [8498], [8150], [8076],  
# [7731], [7696], [7708], [7809], [7226],  
# [6741], [7002], [7264], [7201], [6704],  
# [6459], [6528], [6817], [6393], [6303],  
# [6174], [5651], [5644], [6348], [5745],  
# [6125], [6073], [6481], [6134], [6376],  
# [5245], [5339], [5843], [5412], [5234],  
# [5639], [5194], [5084], [5311], [5270],  
# [518], [524], [513], [517], [520],  
# [513], [508], [519], [510], [512],  
# [509], [518], [506], [508], [514],  
# [511], [519], [514], [513], [512],  
# [514], [516], [501], [503], [503],  
# [516], [508], [506], [515], [513],  
# [505], [512], [515], [513], [515],  
# [509], [509], [506], [517], [512],  
# [7204], [7251], [7004], [6999], [7183],  
# [6802], [6784], [6850], [6897], [6738],  
# [6852], [6730], [6747], [6863], [6779],  
# [6707], [6818], [6810], [6741], [6758],  
# [6704], [6792], [6686], [6602], [6601],  
# [6514], [6627], [6668], [6580], [6524],  
# [6519], [6455], [6528], [6440], [6410],  
# [6475], [6377], [6478], [6407], [6434],  
# [10935], [11389], [11035], [10419], [10582],  
# [10109], [10026], [10037], [10569], [10393],  
# [10199], [10736], [10833], [10467], [10509],  
# [10531], [10528], [10219], [10074], [10885],  
# [9327], [9507], [9596], [9727], [9963],

# [9406], [9657], [9280], [9776], [9211],  
 # [9225], [9193], [9470], [9185], [9293],  
 # [9569], [9603], [9977], [9389], [9818],  
 # [8762], [8903], [8498], [8150], [8076],  
 # [7731], [7696], [7708], [7809], [7226],  
 # [6741], [7002], [7264], [7201], [6704],  
 # [6459], [6528], [6817], [6393], [6303],  
 # [6174], [5651], [5644], [6348], [5745],  
 # [6125], [6073], [6481], [6134], [6376],  
 # [5245], [5339], [5843], [5412], [5234],  
 # [5639], [5194], [5084], [5311], [5270],  
 # [518], [524], [513], [517], [520],  
 # [513], [508], [519], [510], [512],  
 # [509], [518], [506], [508], [514],  
 # [511], [519], [514], [513], [512],  
 # [514], [516], [501], [503], [503],  
 # [516], [508], [506], [515], [513],  
 # [505], [512], [515], [513], [515],  
 # [509], [509], [506], [517], [512],  
 # [7204], [7251], [7004], [6999], [7183],  
 # [6802], [6784], [6850], [6897], [6738],  
 # [6852], [6730], [6747], [6863], [6779],  
 # [6707], [6818], [6810], [6741], [6758],  
 # [6704], [6792], [6686], [6602], [6601],  
 # [6514], [6627], [6668], [6580], [6524],  
 # [6519], [6455], [6528], [6440], [6410],  
 # [6475], [6377], [6478], [6407], [6434],  
 # [10935], [11389], [11035], [10419], [10582],  
 # [10109], [10026], [10037], [10569], [10393],  
 # [10199], [10736], [10833], [10467], [10509],  
 # [10531], [10528], [10219], [10074], [10885],  
 # [9327], [9507], [9596], [9727], [9963],  
 # [9406], [9657], [9280], [9776], [9211],  
 # [9225], [9193], [9470], [9185], [9293],  
 # [9569], [9603], [9977], [9389], [9818],  
 # [8762], [8903], [8498], [8150], [8076],  
 # [7731], [7696], [7708], [7809], [7226],  
 # [6741], [7002], [7264], [7201], [6704],  
 # [6459], [6528], [6817], [6393], [6303],  
 # [6174], [5651], [5644], [6348], [5745],  
 # [6125], [6073], [6481], [6134], [6376],  
 # [5245], [5339], [5843], [5412], [5234],  
 # [5639], [5194], [5084], [5311], [5270],  
 # [518], [524], [513], [517], [520],  
 # [513], [508], [519], [510], [512],  
 # [509], [518], [506], [508], [514],  
 # [511], [519], [514], [513], [512],  
 # [514], [516], [501], [503], [503],  
 # [516], [508], [506], [515], [513],  
 # [505], [512], [515], [513], [515],  
 # [509], [509], [506], [517], [512],

# [7204], [7251], [7004], [6999], [7183],  
# [6802], [6784], [6850], [6897], [6738],  
# [6852], [6730], [6747], [6863], [6779],  
# [6707], [6818], [6810], [6741], [6758],  
# [6704], [6792], [6686], [6602], [6601],  
# [6514], [6627], [6668], [6580], [6524],  
# [6519], [6455], [6528], [6440], [6410],  
# [6475], [6377], [6478], [6407], [6434],  
# [10935], [11389], [11035], [10419], [10582],  
# [10109], [10026], [10037], [10569], [10393],  
# [10199], [10736], [10833], [10467], [10509],  
# [10531], [10528], [10219], [10074], [10885],  
# [9327], [9507], [9596], [9727], [9963],  
# [9406], [9657], [9280], [9776], [9211],  
# [9225], [9193], [9470], [9185], [9293],  
# [9569], [9603], [9977], [9389], [9818],  
# [8762], [8903], [8498], [8150], [8076],  
# [7731], [7696], [7708], [7809], [7226],  
# [6741], [7002], [7264], [7201], [6704],  
# [6459], [6528], [6817], [6393], [6303],  
# [6174], [5651], [5644], [6348], [5745],  
# [6125], [6073], [6481], [6134], [6376],  
# [5245], [5339], [5843], [5412], [5234],  
# [5639], [5194], [5084], [5311], [5270],  
# [518], [524], [513], [517], [520],  
# [513], [508], [519], [510], [512],  
# [509], [518], [506], [508], [514],  
# [511], [519], [514], [513], [512],  
# [514], [516], [501], [503], [503],  
# [516], [508], [506], [515], [513],  
# [505], [512], [515], [513], [515],  
# [509], [509], [506], [517], [512],  
# [7204], [7251], [7004], [6999], [7183],  
# [6802], [6784], [6850], [6897], [6738],  
# [6852], [6730], [6747], [6863], [6779],  
# [6707], [6818], [6810], [6741], [6758],  
# [6704], [6792], [6686], [6602], [6601],  
# [6514], [6627], [6668], [6580], [6524],  
# [6519], [6455], [6528], [6440], [6410],  
# [6475], [6377], [6478], [6407], [6434],  
# [10935], [11389], [11035], [10419], [10582],  
# [10109], [10026], [10037], [10569], [10393],  
# [10199], [10736], [10833], [10467], [10509],  
# [10531], [10528], [10219], [10074], [10885],  
# [9327], [9507], [9596], [9727], [9963],  
# [9406], [9657], [9280], [9776], [9211],  
# [9225], [9193], [9470], [9185], [9293],  
# [9569], [9603], [9977], [9389], [9818],  
# [8762], [8903], [8498], [8150], [8076],  
# [7731], [7696], [7708], [7809], [7226],  
# [6741], [7002], [7264], [7201], [6704],



# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,

# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,

# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,

# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
#  
# 4000, 3000, 4000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 5000, 5000, 5000, 5000, 5000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
# 6000, 6000, 6000, 6000, 6000,  
#  
# 6000, 6000, 6000, 7000, 7000,  
# 7000, 7000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 8000, 8000, 8000, 8000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 8000, 9000, 8000,  
# 9000, 9000, 9000, 9000, 9000,  
# 9000, 9000, 9000, 9000, 9000,  
#  
# 14000, 14000, 14000, 14000, 14000,  
# 14000, 14000, 14000, 14000, 14000,

```

        # 14000, 14000, 14000, 14000, 14000,
        # 14000, 14000, 14000, 14000, 14000,
        # 14000, 14000, 14000, 14000, 14000,
        # 14000, 14000, 14000, 14000, 14000,
        # 14000, 14000, 14000, 14000, 14000,
        # 14000, 14000, 14000, 14000, 14000,
    ])
    model = Sequential([
        Dense(32, activation='relu', input_shape=(1,)),
        Dense(64, activation='relu'),
        Dense(1)
    ])

    model.compile(optimizer='adam', loss='mean_squared_error')
    model.fit(light, phytolampe, epochs=100, batch_size=5,
validation_split=0.2)

    return model

def neural_network_phytolampe(model, L_array):
    return model.predict(L_array)

```

### 1.5 resource/mathematical\_equations/time.py

```

from plant_growing_system.mathematical_models import
mathematical_equations_watering_function
from plant_growing_system.mathematical_models import
mathematical_equations_phytolampe_function
import random
import time

def run_100_000_times():
    for r in [random.randint(0, 101) for _ in range(1_000)]:
        print(mathematical_equations_watering_function(r))

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(mathematical_equations_phytolampe_function(r))

start_time = time.time()

run_100_000_times()

end_time = time.time()
execution_time = end_time - start_time
print(f"Час виконання: {execution_time} секунд")

```

### 1.6 resource/mathematical\_equations/memory.py

```

from plant_growing_system.mathematical_models import
mathematical_equations_watering_function

```

```

from plant_growing_system.mathematical_models import
mathematical_equations_phytolampe_function
import random
from memory_profiler import memory_usage

def run_100_000_times():
    for r in [random.randint(0, 101) for _ in range(100_000)]:
        mathematical_equations_watering_function(r)

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(mathematical_equations_phytolampe_function(r))

if __name__ == '__main__':
    memory_used = memory_usage((run_100_000_times, ()), {})
    print(f"Використання пам'яті: {max(memory_used)} MB")

```

### 1.7 resource/fuzzy\_logic/time.py

```

from plant_growing_system.mathematical_models import
define_fuzzy_logic_watering_control
from plant_growing_system.mathematical_models import
define_fuzzy_logic_phytolampe_control
from plant_growing_system.mathematical_models import fuzzy_logic_watering
from plant_growing_system.mathematical_models import fuzzy_logic_phytolampe
import random
import time

def run_100_000_times(w_level, ph_level):
    for r in [random.randint(0, 101) for _ in range(1_000)]:
        print(fuzzy_logic_watering(w_level, r))

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(fuzzy_logic_phytolampe(ph_level, r))

init_time = time.time()

watering_level = define_fuzzy_logic_watering_control()
phytolampe_level = define_fuzzy_logic_phytolampe_control()

start_time = time.time()

run_100_000_times(watering_level, phytolampe_level)

end_time = time.time()
initiation_time = start_time - init_time
execution_time = end_time - start_time
print(f"Час запуску: {initiation_time} секунд")
print(f"Час виконання: {execution_time} секунд")

```

### 1.8 resource/fuzzy\_logic/memory.py

```

from plant_growing_system.mathematical_models import
define_fuzzy_logic_watering_control
from plant_growing_system.mathematical_models import
define_fuzzy_logic_phytolampe_control
from plant_growing_system.mathematical_models import fuzzy_logic_watering
from plant_growing_system.mathematical_models import fuzzy_logic_phytolampe
import random
from memory_profiler import memory_usage

def run_100_000_times(w_level, ph_level):
    for r in [random.randint(0, 101) for _ in range(1_000)]:
        print(fuzzy_logic_watering(w_level, r))

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(fuzzy_logic_phytolampe(ph_level, r))

watering_level = define_fuzzy_logic_watering_control()
phytolampe_level = define_fuzzy_logic_phytolampe_control()

if __name__ == '__main__':
    memory_used = memory_usage((run_100_000_times, (watering_level,
phytolampe_level), {}))
    print(f"Використання пам'яті: {max(memory_used)} MB")

```

### 1.9 resource/neural\_network/time.py

```

from plant_growing_system.mathematical_models import
define_neural_network_watering
from plant_growing_system.mathematical_models import
define_neural_network_light
from plant_growing_system.mathematical_models import neural_network_watering
from plant_growing_system.mathematical_models import
neural_network_phytolampe
import random
import time
import numpy as np

def run_100_000_times(w_model, l_model):
    for r in [random.randint(0, 101) for _ in range(1_000)]:
        print(neural_network_watering(w_model, np.array([r])))

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(neural_network_phytolampe(l_model, np.array([r])))

init_time = time.time()

watering_model = define_neural_network_watering()
light_model = define_neural_network_light()

```

```

start_time = time.time()

run_100_000_times(watering_model, light_model)

end_time = time.time()
initiation_time = start_time - init_time
execution_time = end_time - start_time
print(f"Час запуску: {initiation_time} секунд")
print(f"Час виконання: {execution_time} секунд")

```

### 1.10 resource/neural\_network/memory.py

```

from plant_growing_system.mathematical_models import
define_neural_network_watering
from plant_growing_system.mathematical_models import
define_neural_network_light
from plant_growing_system.mathematical_models import neural_network_watering
from plant_growing_system.mathematical_models import
neural_network_phytolampe
import random
from memory_profiler import memory_usage
import numpy as np

def run_100_000_times(w_model, l_model):
    for r in [random.randint(0, 101) for _ in range(1_000)]:
        print(neural_network_watering(w_model, np.array([r])))

    for r in [random.randint(0, 20_001) for _ in range(1_000)]:
        print(neural_network_phytolampe(l_model, np.array([r])))

watering_model = define_neural_network_watering()
light_model = define_neural_network_light()

if __name__ == '__main__':
    memory_used = memory_usage((run_100_000_times, ([watering_model,
light_model]), {}))
    print(f"Використання пам'яті: {max(memory_used)} МВ")

```

## ДОДАТОК В

Технічне завдання

ЗАТВЕРДЖУЮ  
Проректор Українського державного  
університету науки і технологій  
Анатолій РАДКЕВИЧ

# АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Керівництво користувача

ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.01336 – 01 ІЗ 01

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН  
Керівник розробки  
\_\_\_\_\_Олександр ІВАНОВ  
Виконавець  
\_\_\_\_\_Марина СТЕШЕНКО  
Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО

44165850.01336 – 01 ІЗ 01

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ КЕРУВАННЯ ПРИРОДНИМ  
СТАНОМ РОСЛИН В АВТОМАТИЗОВАНІЙ СИСТЕМІ ВИРОЩУВАННЯ

Керівництво користувача

Листів 9

## А Н О Т А Ц І Я

Документ 1116130.01269 – 01 ІЗ 01 «Аналіз математичних моделей для керування природним станом рослин в автоматизованій системі вирощування. Керівництво користувача» входить до складу програмної документації на розроблені програмні продукти, що реалізують наступні математичні моделі для системи вирощування рослин:

- математичні рівняння для поливу та освітлення;
- нечітка логіка для поливу та освітлення;
- нейронна мережа для поливу та освітлення.

Програми написані на мові Python з використанням бібліотек NumPy, skfuzzy, TensorFlow, Keras у програмному середовищі PyCharm.

З М І С Т

1 Введення.....	4
2 Призначення та умови застосування.....	5
3 Підготовка до роботи.....	6
4 Опис операцій.....	7
5 Аварійні ситуації.....	8
6 Рекомендації щодо застосування.....	9

## 1 ВВЕДЕННЯ

Програмні продукти реалізують математичні моделі видів математичні рівняння, нечітка логіка та нейронна мережа. Ці математичні моделі дозволяють знаходити необхідну кількість поливу у мл відповідно до заданої вологості ґрунту та необхідний рівень яскравості фітострічки відповідно до рівня зовнішнього освітлення.

Головною метою розробки є надання достатнього інструментарію визначення необхідної кількості поливу та освітлення за допомогою математичних моделей для проведення дослідження та розрахунку оціночних показників, на основі яких буде проводитися дослідження.

Даний продукт призначений для використання науковцями для автоматизації систем вирощування рослин та визначення найбільш ефективної моделі.

Не вимагає ознайомлення із будь-якою додатковою експлуатаційною документацією.

## 2 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Функціональним призначенням програмних продуктів є реалізація різних видів математичних моделей задля їхнього використання у системах вирощування рослин.

Експлуатаційне призначення програмних продуктів – визначення необхідної кількості поливу та освітлення фітострічки згідно до заданих вологості ґрунту та рівня зовнішнього освітлення .

Для забезпечення сталого функціонування програми користувачеві і програмісту необхідно дотримуватися таких умов:

- підтримка пристроєм на котрому будуть запускатися програмні продукти мови Python;
- наявність бібліотек NumPy, skfuzzy, TensorFlow та Keras.

Програмний продукт, що розробляється, розрахований на використання на пристроях, що мають:

- процесор з тактовою частотою не нижче 2 ГГц;
- не менше 250 МБ або 2ГБ оперативної пам'яті.

Для функціонування програмних продуктів необхідні:

- підтримка мови Python;
- наявність бібліотек NumPy, skfuzzy, TensorFlow та Keras.

### 3 ПІДГОТОВКА ДО РОБОТИ

Для початку роботи програмних продуктів необхідно завантажити виконавчі файли на пристрій, впевнитися, що на пристрої встановлені всі необхідні мови та бібліотеки, та запустити програму з терміналу за допомогою команди *python*.

#### 4 ОПИС ОПЕРАЦІЙ

Після завантаження виконавчих файлів на пристрій доступні наступні операції:

- визначення необхідної кількості поливу за допомогою математичних рівнянь;
- визначення необхідної кількості поливу за допомогою нечіткої логіки;
- визначення необхідної кількості поливу за допомогою нейронної мережі;
- визначення необхідного рівня освітлення фітострічки за допомогою математичних рівнянь;
- визначення необхідного рівня освітлення фітострічки за допомогою нечіткої логіки;
- визначення необхідного рівня освітлення фітострічки за допомогою нейронної мережі.

## 5 АВАРІЙНІ СИТУАЦІЇ

Якщо програмний засіб буде запускатися з пристрою який не відповідає необхідним функціональним вимогам, програма виведе на екран повідомлення про помилку та завершить роботу.

Якщо програмний засіб під час роботи буде поводити некоректно чи із помилкою, необхідно перезапустити програму для продовження роботи.

## 6 РЕКОМЕНДАЦІЇ ЩОДО ЗАСТОСУВАННЯ

Після завантаження виконавчих файлів на пристрій їх необхідно запустити через термінал за допомогою команди *python*.

Після запуску виконавчих файлів результати їхньої роботи буде виведено у консоль.

Після ознайомлення із функціями програмних продуктів, їхня роботу завершується і їх можна запустити заново.