

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»

В.о. зав. кафедри комп'ютерних систем
та робототехніки

_____ к.ф.-м.н., доц. М.М. Хруслов
«__» _____ 2025 р.

Пояснювальна записка
до кваліфікаційної роботи
бакалавра

на тему: **Модель прогнозування впливу погодних факторів
на врожайність сільськогосподарських культур**

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
Галузь знань 15 – Автоматизація та приладобудування
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

Захищено на засіданні
Екзаменаційної комісії № 46
протокол № __ від __.06.2025 р.
Оцінка ____ / ____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
студент групи КУ-41
ПОЛЮСКЕВИЧ Данило Станіславович

Керівник: к.е.н, доцент
закладу вищої освіти кафедри
комп'ютерних систем та робототехніки
ЧУБ Ольга Ігорівна

Рецензент: професор кафедри
теоретичної та прикладної інформатики
Харківського національного університету
імені В.Н. Каразіна, д.т.н., професор
ФРОЛОВ В'ячеслав Вікторович

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 67 сторінок, з яких основна частина – 50 сторінок, містить 7 рисунків, 2 таблиці та 3 інфографіки. До списку використаних джерел входить 28 найменувань.

Мета кваліфікаційної роботи – розробити та реалізувати прогнозну модель на основі методів машинного навчання для оцінки впливу погодних умов на врожайність сільськогосподарських культур з метою підвищення точності аграрного планування та підтримки прийняття управлінських рішень.

Об’єкт дослідження – процес прогнозування врожайності сільськогосподарських культур на основі аналізу впливу погодних факторів.

Предмет дослідження – методи машинного навчання для прогнозування впливу погодних умов на врожайність сільськогосподарських культур, з акцентом на використанні нейронних мереж типу LSTM.

Проблема, що вирішується у роботі, полягає у підвищенні точності прогнозування врожайності сільськогосподарських культур шляхом використання методів машинного навчання для автоматизованого аналізу впливу погодних умов і формування обґрунтованих аграрних рішень.

Область застосування – агропромисловий сектор та агроаналітика, зокрема, системи підтримки прийняття рішень в аграрному менеджменті на основі прогнозних моделей. Розроблене рішення може використовуватись для оцінки потенційної врожайності, планування посівних кампаній, оптимізації агротехнічних заходів та управління ризиками, пов’язаними зі зміною погодних умов. Система прогнозування може бути інтегрована у веб-сервіси, мобільні додатки або хмарні платформи, що забезпечує гнучкість і доступність навіть у регіонах з обмеженою ІТ-інфраструктурою.

Ключові слова: ПРОГНОЗУВАННЯ, LSTM, ШТУЧНИЙ ІНТЕЛЕКТ, СІЛЬСЬКЕ ГОСПОДАРСТВО, ПОГОДНІ ДАНІ, АГРОАНАЛІТИКА, PYTHON.

ABSTRACT

The explanatory note for the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices. The total length of the work is 67 pages, of which the main part – 50 pages – contains 7 figures, 2 tables, and 3 infographics. The list of references includes 28 sources.

The aim of the qualification work is to develop and implement a predictive model based on machine learning methods to assess the impact of weather conditions on the yield of agricultural crops, with the goal of improving the accuracy of agricultural planning and supporting decision-making processes.

The object of the study is the process of forecasting agricultural crop yields based on the analysis of weather factors.

The subject of the study is machine learning methods for modeling the impact of weather conditions on crop yield, with a focus on the use of LSTM-type neural networks.

The problem addressed in the work is to improve the accuracy of crop yield forecasting by applying machine learning methods for automated analysis of weather conditions and the formation of data-driven agricultural decisions.

Field of application – the agro-industrial sector and agroanalytics, in particular, decision support systems in agricultural management based on predictive models. The developed solution can be used for assessing potential yields, planning sowing campaigns, optimizing agrotechnical measures, and managing risks associated with changing weather conditions. The forecasting system can be integrated into web services, mobile applications, or cloud platforms, ensuring flexibility and accessibility even in regions with limited IT infrastructure.

Keywords: FORECASTING, LSTM, ARTIFICIAL INTELLIGENCE, AGRICULTURE, WEATHER DATA, AGROANALYTICS, PYTHON.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ	8
1.1. Аналіз проблеми прогнозування впливу погодних факторів на врожайність	8
1.2. Сучасний стан досліджень у сфері прогнозування врожайності.....	9
1.3. Обґрунтування вибору нейронної мережі.....	12
1.4. Цифрові рішення в агросекторі.....	13
1.4.1 Аналіз інструментів та технологій, що використовуються в сільському господарстві.....	14
1.4.2 Аналіз цільової аудиторії програмного забезпечення	15
Висновки за розділом 1	16
РОЗДІЛ 2. КОНСТРУЮВАННЯ ПРОГНОЗНОЇ МОДЕЛІ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ LSTM	17
2.1. Загальна методика побудови прогнозної моделі	17
2.1.1 Постановка задачі прогнозування врожайності	17
2.1.2 Вибір вхідних параметрів.....	18
2.2. Підготовка даних для моделі	19
2.2.1 Агрегація та форматування	19
2.2.2 Заповнення пропусків, обробка аномалій.....	20
2.3. Використання інженерії ознак.....	21
2.3.1 Декомпозиція часового ряду	21
2.3.2 Масштабування та нормалізація даних.....	22
2.3.3 Формування послідовностей та розбиття даних.....	23
2.4. Розробка та навчання моделі	24
2.4.1 Кількість шарів, нейронів, функція активації	24
2.4.2 Вибір функції втрат, оптимізатора, швидкості навчання	26
2.4.3 Налаштування гіперпараметрів	26
Висновки за розділом 2	27

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ПРОГНОЗНОЇ МОДЕЛІ	29
3.1. Структура та основні компоненти.....	29
3.1.1 Модулі, функціональні блоки, розмежування логіки	29
3.1.2 Отримання погодних даних з Meteostat API.....	30
3.2. Первинна обробка та структурування погодних рядів для моделі	32
3.3. Реалізація та навчання нейронної мережі	33
3.3.1 Реалізація моделі LSTM.....	33
3.3.2 Навчання моделі	35
3.3.3 Збереження навченої моделі та повторне використання для прогнозування.....	37
3.4. Генерація прогнозу та взаємодія з користувачем	38
3.4.1 Генерація прогнозу	38
3.4.2. Відображення історичних даних за останні 30 днів.....	40
3.4.3 Формування рекомендацій на основі прогнозу	42
3.5. Аналіз точності та порівняння з базовими підходами	45
3.5.1 Методика оцінки точності	45
3.5.2 Візуалізація результатів прогнозу.....	47
3.5.3 Порівняння з базовими підходами.....	48
Висновки за розділом 3	51
ВИСНОВКИ.....	52
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А	56
ДОДАТОК Б	58
ДОДАТОК В.....	61
ДОДАТОК Г	67

ВСТУП

Сільське господарство є одним із найважливіших секторів економіки, який забезпечує продовольчу безпеку, стабільність на внутрішніх ринках та експортний потенціал країни. В умовах зміни клімату, збільшення частоти екстремальних погодних явищ і нестабільності аграрних сезонів зростає потреба в сучасних інструментах, що дозволяють підвищити точність прогнозування врожайності сільськогосподарських культур. Традиційні методи аналітики часто не враховують складну динаміку погодних факторів та їх вплив на розвиток культур, що призводить до значних похибок у прогнозах і неефективного використання ресурсів.

Інтенсивний розвиток цифрових технологій відкриває нові можливості для аграрного сектору, зокрема завдяки застосуванню штучного інтелекту та машинного навчання для моделювання складних процесів. Одним із перспективних підходів є використання нейронних мереж довготривалої короткочасної пам'яті (LSTM), які добре справляються з обробкою часових рядів, що особливо актуально при аналізі погодних даних. Розробка таких моделей дозволяє не лише прогнозувати врожайність на основі історичних кліматичних показників, а й адаптувати рішення до конкретних умов кожного господарства.

Актуальність дослідження зумовлена потребою аграрного сектору у точних, доступних та масштабованих рішеннях, що можуть працювати як в умовах великих агропідприємств, так і в малих фермерських господарствах. Своєчасне та обґрунтоване планування посівів, вибір культур і агротехнічних заходів значною мірою залежить від якісного прогнозування, яке можна забезпечити лише за рахунок впровадження цифрових інструментів.

Мета кваліфікаційної роботи – розробити та реалізувати прогнозну модель на основі методів машинного навчання для оцінки впливу погодних умов на врожайність сільськогосподарських культур з метою підвищення точності аграрного планування та підтримки прийняття управлінських рішень.

Для досягнення поставленої мети в роботі було визначено такі основні завдання:

1. Огляд сучасних підходів до прогнозування врожайності сільськогосподарських культур із використанням математичних моделей, статистичних методів і машинного навчання.
2. Розробка концептуальної моделі нейронної мережі для прогнозування впливу погодних факторів на врожайність.
3. Тестування розробленої моделі на реальних або симульованих даних для оцінки її точності та ефективності.
4. Аналіз можливих обмежень та проблем прогнозування, таких як нестача даних, вплив непередбачуваних факторів або адаптація моделі до різних регіонів.
5. Підготовка рекомендацій щодо вдосконалення моделі та її практичного застосування в агропромисловому секторі.

У дослідженні застосовано методи аналізу часових рядів та інструменти обробки даних із використанням мовного середовища Python [2]. Для роботи з даними використовувалися бібліотеки *pandas* [3], *statsmodels* [4] та *meteostat* [5], а для створення й навчання нейронної мережі – фреймворк *PyTorch*.

Практичне значення роботи полягає у можливості впровадження розробленої моделі в агропідприємствах, цифрових аналітичних платформах або мобільних рішеннях, орієнтованих на фермерські господарства. Такий підхід може стати ефективним інструментом для планування сівозміни, вибору культур відповідно до погодних умов та управління аграрними ризиками.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ

1.1. Аналіз проблеми прогнозування впливу погодних факторів на врожайність

Забезпечення стабільної врожайності сільськогосподарських культур значною мірою залежить від погодних умов, які безпосередньо впливають на всі етапи розвитку рослин – від проростання до дозрівання. Коливання температури, нестача або надлишок опадів, зниження вологості ґрунту, зменшення кількості сонячної радіації – всі ці фактори можуть знижувати продуктивність культур, навіть за умов правильної агротехніки.

У зв'язку з посиленням кліматичних змін проблема стає ще більш актуальною. Нестабільність погодних умов змушує аграріїв адаптуватися до нових реалій: змінювати строки сівби, підбирати стійкіші сорти, оперативно реагувати на стресові фактори навколишнього середовища. В таких умовах стає критично важливою здатність вчасно оцінювати погодну ситуацію та приймати рішення на основі прогнозу.

Прогнозування врожайності не є новим напрямом. Упродовж останніх десятиліть у галузі розроблено численні методи – від емпіричних таблиць до складних математичних моделей. Сучасні підходи використовують великий обсяг даних: супутникові знімки, дані з дронів, метеостанцій, ґрунтові сенсори, дистанційне зондування [6]. У поєднанні з ГІС це дає змогу аналізувати просторово-часові патерни впливу клімату на посіви.

Проте більшість існуючих моделей зосереджені на числовому прогнозі врожайності [7]. Такий підхід не завжди є практично доцільним у реальному виробництві, де аграрій часто стикається з іншим завданням: яку культуру краще обрати для вирощування в поточних умовах? У цьому контексті актуальності набувають системи, які замість оцінки врожайності формують рекомендації щодо доцільності вирощування культур з урахуванням поточних погодних умов, динаміки та історичних трендів.

1.2. Сучасний стан досліджень у сфері прогнозування врожайності

Прогнозування врожайності сільськогосподарських культур є важливим напрямом як у фундаментальних дослідженнях, так і в прикладних розробках аграрного сектору. У першу чергу це пов'язано з потребою в точному плануванні виробництва, оптимізації використання ресурсів та гарантуванні продовольчої безпеки на національному і регіональному рівнях.

Упродовж останніх десятиліть дослідники застосовували різні підходи до прогнозування – від класичних статистичних моделей до сучасних алгоритмів машинного та глибокого навчання. Серед традиційних методів найчастіше використовуються регресійні підходи та моделі вирівнювання динамічних рядів. Вони прості в реалізації, проте мають обмежену здатність враховувати складну багатфакторну природу впливу кліматичних умов.

Сучасні методи прогнозування демонструють значне зростання точності завдяки застосуванню алгоритмів машинного навчання, які здатні аналізувати великі обсяги даних і враховувати нелінійні залежності. Наприклад, моделі LightGBM [8], XGBoost [9], Random Forest [10] успішно застосовуються для аналізу просторових та метеорологічних даних. Для аналізу супутникових зображень та зонування територій використовуються архітектури типу U-Net [11], які дають змогу отримувати точну інформацію про стан полів.

Інтеграція кількох підходів – статистичних, аналітичних і нейромережових – дозволяє створювати гібридні моделі, які перевершують класичні за точністю, але потребують якісно підготовлених наборів даних та значних обчислювальних ресурсів.

В українському науковому просторі також ведуться активні дослідження у цьому напрямі. Зокрема, у методичних рекомендаціях вітчизняних агроустанов застосовуються такі підходи, як метод аналогій, експертних оцінок та динамічного вирівнювання, які враховують вплив основних факторів росту на продуктивність. Однак більшість таких методик зосереджені на загальному прогнозі урожайності – у тоннах з гектара чи у відсотках від середнього значення.

З огляду на зростання складності кліматичних умов та потреби в адаптивному управлінні посівними кампаніями, усе більшої актуальності набувають інтелектуальні системи, здатні враховувати не лише абсолютні показники врожайності, а й змінні погодні сценарії. У цьому контексті дослідники дедалі частіше звертаються до моделей, що аналізують часову динаміку метеорологічних параметрів для формування оперативних аграрних рішень. Такий підхід потребує комплексного аналізу існуючих алгоритмів – від класичних статистичних до сучасних нейромережевих.

Прогнозування врожайності сільськогосподарських культур є багатофакторною задачею, для вирішення якої застосовуються різноманітні методи та технології. Вибір конкретної методології залежить від характеру даних, цілей прогнозування та доступних ресурсів. Серед найбільш широко використовуваних підходів можна виділити регресійні моделі [12], статистичні методи [13], а також методи машинного навчання [14], кожен із яких має свої переваги й обмеження.

На рис. 1.1 представлено три основні підходи, що найчастіше застосовуються для задач прогнозування врожайності: регресійні моделі, статистичні методи та методи машинного навчання. Кожен з них має свою сферу застосування та рівень складності обробки даних.

Розглянемо кожен метод окремо:

- Регресійні моделі є одним із основних інструментів для моделювання залежностей між різними факторами, що впливають на врожайність. Ці моделі допомагають виявити лінійні та нелінійні зв'язки між залежними й незалежними змінними, такими як погодні умови, кількість опадів, температура та інші фактори. Використання множинних лінійних регресій дає змогу оцінити важливість кожного з факторів і їх взаємодію. Однак такий підхід обмежений у випадках, коли між змінними існують складні нелінійні залежності.



Рисунок 1.1 – Методи прогнозування врожайності

- Статистичні методи займають важливе місце в аналізі даних. До таких методів відносяться кореляційний аналіз, аналіз часових рядів та факторний аналіз. Кореляція допомагає визначити, наскільки сильно пов'язані між собою різні змінні, що впливають на врожайність. Аналіз часових рядів особливо важливий для вивчення сезонних коливань і прогнозування врожайності на основі історичних даних. Такі методи дають можливість виявити тренди та цикли в погодних умовах і впливі на урожайність.

- Методи машинного навчання все частіше застосовуються в сільському господарстві для прогнозування врожайності, оскільки ці технології здатні обробляти великі обсяги даних і виявляти складні патерни. Використання алгоритмів, таких як нейронні мережі, дерева рішень, методи підтримки векторних машин (SVM) [15] та ансамблеві методи, дозволяє створювати високоточні моделі, які враховують безліч змінних одночасно. Зокрема, глибокі нейронні мережі (deep learning) [16] демонструють гарні результати у випадках, коли існує велика кількість різноманітних змінних, які важко моделювати традиційними методами.

Усе частіше у практиці прогнозування використовуються гібридні методи, які комбінують регресійні підходи з методами машинного навчання.

Наприклад, методи, що використовують глибоке навчання, можуть бути поєднані з традиційними регресійними моделями для підвищення точності прогнозів.

1.3. Обґрунтування вибору нейронної мережі

Задача аналізу погодних даних для прийняття рішень у сільському господарстві вимагає врахування часової структури вхідної інформації. Температура, вологість, опади – всі ці параметри змінюються з часом, і їх вплив не є миттєвим: часто ефект спостерігається із затримкою, наприклад, через фазу розвитку культур.

У такому контексті модель має не просто обробляти окремі значення, а й виявляти закономірності в динаміці. Саме цим вимогам відповідає архітектура з довготривалою короткочасною пам'яттю. Вона забезпечує збереження інформації про попередні стани та дозволяє моделі «пам'ятати» ключові погодні патерни, які впливають на подальший розвиток рослин.

На відміну від звичайних рекурентних мереж, ця архітектура вирішує проблему затухаючого градієнта, що дає змогу моделювати тривалі часові залежності. Це критично важливо для задач, де, наприклад, волога у ґрунті, накопичена в травні, може вплинути на урожай у серпні.

Ще однією важливою причиною вибору цієї моделі є її гнучкість. LSTM-підхід може адаптуватися до нових кліматичних сценаріїв, масштабуватися на різні регіони та працювати з часовими рядами різної довжини. Він сумісний з іншими компонентами машинного навчання – наприклад, з модулями нормалізації, декомпозиції тренду, інтеграцією з супутниковими даними тощо.

Таким чином, застосування цієї архітектури дає змогу не лише підвищити точність прогнозу погодних умов, а й використовувати їх для формування рекомендацій щодо вибору культур. Це робить модель цінним інструментом підтримки аграрних рішень, орієнтованим на реальну практику.

Нейронні мережі, зокрема архітектури з рекурентною пам'яттю, широко застосовуються для задач прогнозування часових рядів. Їх ключова перевага

полягає у здатності моделювати нелінійні залежності та зберігати інформацію про попередні стани вхідних даних. Це забезпечує суттєву перевагу над класичними статистичними моделями, такими як ARIMA [17], які зазвичай розраховані на стаціонарні процеси з фіксованою сезонністю.

На рис. 1.2 узагальнено основні переваги та обмеження нейромережових моделей у задачах прогнозування часових рядів.



Рисунок 1.2 – Переваги та обмеження нейронних мереж у задачах прогнозування часових рядів

1.4. Цифрові рішення в агросекторі

У сучасному сільському господарстві прогнозні моделі відіграють ключову роль у прийнятті обґрунтованих рішень. Успішне застосування таких моделей залежить не лише від обраної архітектури чи точності алгоритмів, а й від технологічного контексту, у якому вони використовуються. Розуміння специфіки галузі, доступних цифрових інструментів та особливостей їх адаптації до українських реалій є необхідним етапом перед розробкою практичного рішення. У підпунктах цього розділу розглянуто ключові технології, що застосовуються в аграрній сфері, а також цільову аудиторію, на яку орієнтовано розроблювану систему.

1.4.1 Аналіз інструментів та технологій, що використовуються в сільському господарстві

Впровадження сучасних цифрових технологій у сільське господарство України значно підвищує ефективність аграрного сектору. Для прогнозування врожайності, аналізу погодних умов та оптимізації процесів вирощування культур активно застосовуються інструменти машинного навчання та обробки великих даних. Найбільш поширеними серед них є мови програмування (Python, R), платформи глибокого навчання (TensorFlow), інструменти для обробки даних (Pandas, NumPy) та сервіси хмарних обчислень (Google Cloud AI, AWS Machine Learning, Microsoft Azure AI).[18].

Аналіз платформ для прогнозування врожайності наведений у табл. 1.1. Однією з основних мов програмування, яка використовується в агротехнологіях, є Python. Він забезпечує широкий набір бібліотек для аналізу даних і побудови моделей машинного навчання. Наприклад, Pandas дозволяє обробляти великі набори аграрних даних, NumPy використовується для виконання математичних розрахунків, а Matplotlib і Seaborn допомагають візуалізувати зміни у врожайності залежно від кліматичних факторів. У деяких агрокомпаніях Python застосовується для розробки моделей прогнозування врожайності на основі багаторічних кліматичних спостережень.

Серед платформ для розробки нейронних мереж у сільському господарстві найбільшого поширення набули TensorFlow та PyTorch. TensorFlow широко використовується в комерційних рішеннях завдяки масштабованості та можливості інтеграції з хмарними сервісами. Наприклад, в агрохолдингах він застосовується для побудови глибоких нейронних мереж, що аналізують супутникові знімки полів, прогножуючи рівень вологості ґрунту та необхідність зрошення. PyTorch, завдяки гнучкості у створенні кастомних моделей, часто використовується в наукових дослідженнях. Його застосовують у моделюванні часових рядів для прогнозування зміни погодних умов та розповсюдження хвороб культур.

Таблиця 1.1

Аналіз платформ для прогнозування врожайності

Регіон	Назва компанії	Технологія	Платформа
Україна	Астарта-Київ	Цифрова система управління агробізнесом	AgriChain (власна розробка)
Україна	AGRAIN	Супутниковий моніторинг, аналітика врожайності	EOSDA Crop Monitoring
Україна	Kernel	Система управління агровиробництвом	Digital AgriBusiness
Україна	Meteo.Farm	Мережа метеостанцій, прогнозування погодних умов	Власна IoT-платформа
Європа	365FarmNet	Автоматичне управління сільськогосподарськими процесами	Веб-платформа для агробізнесу
Європа	Farmonaut	Супутниковий моніторинг, AI-аналітика	Веб та мобільні додатки
Європа	Gamaya	Гіперспектральна зйомка, аналіз стану посівів	Дрони з гіперспектральними камерами

1.4.2 Аналіз цільової аудиторії програмного забезпечення

Основна цільова аудиторія включає великі агрохолдинги, середні та малі фермерські господарства, а також власників тваринницьких підприємств, які зацікавлені у підвищенні ефективності своїх операцій.

Залежно від масштабу господарства, кількість користувачів програмного забезпечення може варіюватися. У великих агрохолдингах до аналізу та прийняття рішень залучені цілі департаменти агрономів, аналітиків та керівників, що працюють із великими обсягами даних. Середні та малі фермери, навпаки, часто покладаються на автоматизовані системи

прогнозування, оскільки не мають можливості утримувати окремий аналітичний відділ. Важливо, щоб програмне забезпечення було інтуїтивно зрозумілим і не вимагало спеціальних технічних навичок.

Одним із ключових факторів, що впливає на доступність технології для фермерів, є її неприв'язаність до конкретного обладнання. Оскільки багато дрібних фермерських господарств не можуть дозволити собі дорогі сервери або спеціалізовані обчислювальні системи, важливою перевагою є можливість використання програмного забезпечення на звичайних ПК або в хмарних сервісах. Веб-інтерфейси або мобільні додатки можуть забезпечити доступність рішення навіть у віддалених регіонах без необхідності значних інвестицій у IT-інфраструктуру.

Цінова політика також є критично важливим фактором для малих та середніх фермерських господарств, які не мають великих фінансових ресурсів. Тому безкоштовні або умовно-безкоштовні версії ПЗ (freemium-модель) можуть значно розширити аудиторію користувачів. Для великих агрохолдингів може бути запропонована модель передплати або платний доступ до розширених аналітичних функцій. Важливим є також мінімізація вартості впровадження, що досягається завдяки відкритим платформам і простій інтеграції з існуючими системами фермерів.

Висновки за розділом 1

У першому розділі було проаналізовано сучасні методи прогнозування врожайності сільськогосподарських культур, із фокусом на їх застосування в умовах нестабільного клімату. Розглянуто класичні статистичні підходи, методи машинного навчання та глибокі нейронні мережі. Обґрунтовано доцільність використання LSTM-архітектури для обробки часових рядів, що відображають погодні умови, з огляду на її здатність зберігати контекст і моделювати довготривалі залежності. Також виконано аналіз цифрових технологій, які вже використовуються в агросекторі України, та виявлено прогалини, що можуть бути заповнені за допомогою запропонованого підходу.

РОЗДІЛ 2.

КОНСТРУЮВАННЯ ПРОГНОЗНОЇ МОДЕЛІ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ LSTM

2.1. Загальна методика побудови прогнозової моделі

Процес створення ефективної системи прогнозування врожайності вимагає чіткого формулювання задачі, вибору відповідної архітектури моделі та налаштування її взаємодії з вхідними даними. Враховуючи складну природу аграрних процесів і високий ступінь впливу погодних факторів, доцільно застосовувати інструменти, здатні моделювати часову динаміку та виявляти приховані закономірності у великих обсягах інформації. У цьому розділі розглянуто етапи побудови нейромережевої моделі, починаючи від постановки задачі, аналізу вхідних даних та формування архітектури, і завершуючи підготовкою програмної реалізації та налаштуванням процесу навчання.

2.1.1 Постановка задачі прогнозування врожайності

Прогнозування врожайності сільськогосподарських культур базується на аналізі факторів, що впливають на розвиток рослин упродовж вегетаційного періоду. Одними з основних таких факторів є погодні умови – температура повітря, кількість опадів, вологість та швидкість вітру. Ці показники мають часову динаміку, і їхній вплив на урожай, як правило, не є миттєвим, а накопичується з часом. Тому при формулюванні задачі прогнозування доцільно використовувати підхід, що враховує саме часові залежності між вхідними змінними та цільовим результатом.

Метою є побудова моделі, яка дозволить оцінити майбутній обсяг урожаю на основі погодних даних за попередній період. Це завдання належить до класу задач регресії, де вхідними даними є багатовимірний часовий ряд метеорологічних параметрів, а вихідною змінною – числове значення, яке описує очікувану продуктивність (наприклад, в центнерах з гектара або умовних індексах).

З огляду на складну структуру впливу погодних чинників, для моделювання доцільно використовувати нейронні мережі типу LSTM, які здатні зберігати інформацію про попередні стани та виявляти довготривалі залежності у даних. Також необхідним є попереднє формування навчальних послідовностей (вікон) з фіксованою довжиною та обробка даних – включаючи заповнення пропусків, декомпозицію, масштабування тощо.

2.1.2 Вибір вхідних параметрів

Ефективність моделі прогнозування врожайності значною мірою залежить від правильного вибору вхідних параметрів – тобто тих змінних, що описують стан зовнішнього середовища і безпосередньо або опосередковано впливають на розвиток сільськогосподарських культур. У цьому дослідженні основну увагу зосереджено на чотирьох типових метеорологічних показниках: середньодобовій температурі повітря, кількості опадів, відносній вологості та швидкості вітру.

Кожен із цих параметрів має своє обґрунтоване значення. Температура визначає швидкість біохімічних процесів у рослинах і впливає на стадії вегетації – від проростання насіння до дозрівання. Опади є основним джерелом вологи, і тривалі періоди засухи або, навпаки, надмірна кількість дощів можуть серйозно знизити врожай. Вологість повітря безпосередньо впливає на транспірацію та випаровування, а також на стійкість рослин до стресу. Швидкість вітру, хоч і менш очевидна, може впливати на висихання ґрунту, температуру на поверхні посівів, запилення, а у випадках сильного вітру – навіть на механічне пошкодження культур.

Окрім біологічної доцільності, обрані параметри мають і практичні переваги. Вони доступні для збору з використанням відкритих метеорологічних сервісів (зокрема, API Meteostat), мають добру історичну глибину, регулярність спостережень і стандартний формат. Це полегшує подальшу агрегацію, очищення та інтеграцію даних у модель.

Вибір саме цих змінних дозволяє досягти балансу між інформативністю,

доступністю та обчислювальною ефективністю. У разі потреби модель може бути розширена іншими параметрами, такими як сонячна радіація або температура ґрунту, але базовий набір із чотирьох факторів уже забезпечує достатню основу для побудови первинної прогностичної системи.

2.2. Підготовка даних для моделі

Після визначення ключових метеорологічних параметрів, які використовуються як вхідні змінні для моделі, наступним кроком є їх якісна підготовка. Необхідно привести дані до уніфікованого формату, адаптованого для обробки нейронною мережею. Цей етап включає агрегацію, вирівнювання, очищення та попередню трансформацію часових рядів, з урахуванням особливостей джерела даних і вимог моделі. Нижче розглянуто конкретні підходи до агрегації та форматування параметрів, зібраних із різною частотою та точністю.

2.2.1 Агрегація та форматування

Після отримання погодних даних ми виконали їх агрегацію та форматування відповідно до вимог вхідного формату для нейронної мережі. Оскільки метеорологічні параметри надходили з різною частотою спостережень, кожен тип даних оброблявся окремо.

Температура повітря, кількість опадів і швидкість вітру були доступні в щоденному форматі, тож ми зберегли ці значення без додаткової агрегації. Добова роздільність є достатньою для відображення загальних погодних умов, тому подальше згладжування на цьому етапі не застосовувалося.

Інша ситуація склалася з вологістю повітря: вона доступна лише у вигляді погодинних вимірювань. Щоб привести цей параметр до єдиного формату, ми згрупували значення за датою та розраховали максимальне значення вологості протягом кожного дня. Такий підхід дозволяє фіксувати потенційно стресові для культур умови – наприклад, коли вологість протягом доби перевищувала критичний рівень навіть протягом нетривалого часу.

Для забезпечення сумісності всіх рядів ми виконали перетворення масивів до єдиного формату з однаковою довжиною та синхронізували їх за календарною шкалою. У разі, якщо обсяги окремих рядів не збігались за кількістю днів, ми скоригували масиви шляхом зрізу до мінімального спільного розміру.

Окрім щоденного формату, ми розглядали можливість формування тижневих агрегованих значень, зокрема для подальшого порівняння або тестування альтернативної структури моделі. Такий підхід передбачає використання ковзного середнього з фіксованим вікном [19], однак для основного сценарію прогнозування ми зупинилися на щоденній роздільності як базовій.

2.2.2 Заповнення пропусків, обробка аномалій

Після агрегації та синхронізації метеорологічних параметрів ми провели попередню перевірку масивів на наявність пропущених значень та аномалій. Цей етап є необхідним для забезпечення стабільності моделі, оскільки пропуски або хибні значення можуть призводити до зниження якості навчання або до помилкових прогнозів.

У процесі перевірки ми виявили, що окремі значення, особливо в масиві вологості, іноді відсутні або представлені як порожні. Це типова ситуація для погодинних даних, особливо якщо спостереження здійснювалися неповний день або були технічні збої на станції. Ми вирішили заповнювати такі пропуски методом зворотного заповнення [20] – тобто використовуючи найближче наявне значення з наступного дня. Якщо ж значення були відсутні декілька днів поспіль, ми поєднували цей підхід із лінійною інтерполяцією [21], яка дозволяє згладити перехід між сусідніми реальними спостереженнями.

Крім пропусків, ми звернули увагу на потенційні аномалії – нетипово високі або низькі значення, які суттєво відхиляються від середніх сезонних коливань. Такі відхилення могли бути пов'язані як з погодними екстремумами,

так і з помилками у вихідних даних. Для їхньої первинної діагностики ми застосували візуалізацію кожного ряду у вигляді графіків із нанесеними щоденними значеннями. Це дозволило нам виявити окремі «викиди», які суттєво не відповідали загальній динаміці. У випадках, коли такі аномалії не мали фізичного обґрунтування (наприклад, опади 999 мм/день), ми замінювали їх найближчим середнім значенням у межах ± 3 днів.

Ми свідомо уникали автоматичного видалення пропусків або аномалій, щоб не втрачати важливу інформацію в часовому ряду. Замість цього ми обрали комбінований підхід: м'яке коригування, інтерполяція та візуальний контроль.

2.3. Використання інженерії ознак

Після усунення пропусків та аномалій, метеорологічні дані стали придатними для глибшої трансформації. Для підвищення ефективності навчання моделі й покращення точності прогнозування ми застосували інженерію ознак – зокрема, декомпозицію часових рядів на окремі компоненти. Це дозволяє моделі краще вловлювати довгострокові закономірності, виділяти сезонні ефекти та знижувати ризик перенавчання. У наступному підпункті наведено реалізований підхід до декомпозиції.

2.3.1 Декомпозиція часового ряду

Після попередньої підготовки даних ми виконали декомпозицію часових рядів для виділення трендової та сезонної складових. Це дозволяє покращити якість моделі, оскільки розділення систематичних змін у даних допомагає нейронній мережі фокусуватися на прогнозуванні тренду, мінімізуючи вплив сезонних коливань.

На цьому етапі ми розглядали кожен часовий ряд окремо. Для температури, вологості, опадів та швидкості вітру ми виділили:

- Трендову компоненту – загальну довгострокову тенденцію зміни показника з часом [22].

- Сезонну компоненту – регулярні коливання, що повторюються в межах певного періоду (наприклад, річний цикл для температури чи опадів).

Ми застосували адитивну модель декомпозиції, яка передбачає, що загальне значення кожного параметра складається з суми тренду, сезонної складової та залишкової варіації. Такий підхід є обґрунтованим для метеорологічних даних, де сезонні коливання зазвичай мають більш-менш постійну амплітуду.

Для виділення тренду ми використали інтерполяцію відсутніх значень та згладжування, що дозволило отримати плавну лінію, яка описує основну зміну показника у часі. Сезонну компоненту ми зберегли окремо для можливого використання при зворотному перетворенні результатів прогнозу на фінальному етапі.

Таким чином, в результаті декомпозиції не було суттєвих викривлень або втрати інформації. Тому ми переходимо до етапу нормалізації трендових даних, що є необхідним для ефективного навчання нейронної мережі.

2.3.2 Масштабування та нормалізація даних

Після виділення трендової складової ми виконали масштабування даних для приведення їх до уніфікованого діапазону. Масштабування є необхідним етапом при підготовці даних для навчання нейронних мереж, оскільки значення різних порядків можуть уповільнити або ускладнити процес оптимізації моделі.

Ми обрали метод мінімаксної нормалізації [23], що полягає у перетворенні значень у межах від 0 до 1. Для кожного параметра ми обчислили мінімальне та максимальне значення тренду за весь період навчальної вибірки та виконали нормалізацію шляхом лінійного перетворення.

Такий підхід дозволяє:

- уникнути домінування змінних із великими абсолютними значеннями над змінними з меншими масштабами;

- забезпечити стабільність градієнтного спуску при навчанні моделі;
- прискорити процес сходження моделі до мінімуму функції втрат.

Важливо зазначити, що для кожного параметра ми зберігали знайдені мінімальні та максимальні значення окремо. Це дозволяло нам після завершення прогнозування виконати зворотне масштабування результатів у реальні одиниці виміру. Оскільки наша модель оперує лише нормалізованими значеннями тренду, правильне відновлення масштабу даних є критичним для подальшої інтерпретації прогнозу та побудови рекомендацій.

Таким чином, етап нормалізації забезпечив єдність масштабу всіх вхідних даних і створив умови для ефективного навчання нейронної мережі.

2.3.3 Формування послідовностей та розбиття даних

Після нормалізації даних ми сформувавши навчальні послідовності для подачі на вхід нейронної мережі типу LSTM. Оскільки такі моделі працюють із послідовними даними, важливо правильно організувати часові ряди таким чином, щоб зберегти часову залежність між спостереженнями.

Вихідні нормалізовані дані охоплювали дворічний період, що забезпечує наявність сезонних циклів і достатній обсяг для навчання. На основі цього суцільного часового ряду ми формували навчальні приклади методом ковзного вікна фіксованої довжини.

Кожна навчальна послідовність включала:

- Вхідні дані (X) – послідовність нормалізованих значень за 30 днів;
- Цільове значення (y) – нормалізоване значення параметра на наступний день після останнього елемента послідовності.

Таким чином, ми встановили розмір вікна у 30 днів, що дозволяє моделі враховувати майже місячний період погодних змін для прогнозування майбутніх значень. Це рішення обґрунтоване тим, що погодні процеси та агрономічні умови часто мають інерційний характер і проявляють стабільні тенденції в межах одного місяця.

Процес формування послідовностей здійснювався ковзанням вікна по часовому ряду з кроком в один день, що дозволило отримати велику кількість навчальних прикладів і максимально ефективно використати доступні дані. Структурування вхідних даних у вигляді пар «послідовність → прогнозоване значення» забезпечує можливість моделі вчитися на часових залежностях та правильно інтерпретувати зміну погодних факторів у динаміці.

Після формування навчальних вибірок ми виконали їх розбиття на тренувальну та тестову частини. Для цього було обрано класичне співвідношення – 80% даних для тренування і 20% для тестування. Поділ здійснювався без перемішування, оскільки для часових рядів важливо зберігати хронологічний порядок спостережень. Такий підхід дозволяє об'єктивно оцінити здатність моделі прогнозувати нові дані, що знаходяться за межами тренувальної вибірки, і уникнути ризику перенавчання.

2.4. Розробка та навчання моделі

На основі підготовлених і структурованих даних наступним кроком стала безпосередня розробка архітектури нейронної мережі та її навчання. На цьому етапі було визначено оптимальні параметри моделі, обрано функції активації, втрат, оптимізатор, а також проведено налаштування гіперпараметрів, необхідних для ефективного навчання прогнозної системи.

2.4.1 Кількість шарів, нейронів, функція активації

Після завершення етапу попередньої обробки ми розробили архітектуру нейронної мережі для прогнозування погодних параметрів на основі послідовних даних. Основними характеристиками моделі є кількість рівнів, розмірність прихованих станів та логіка функціонування шарів.

На рис. 2.1 наведено спрощену архітектуру побудованої LSTM-моделі, яка складається з двох рекурентних шарів та фінального лінійного шару, що генерує прогноз.

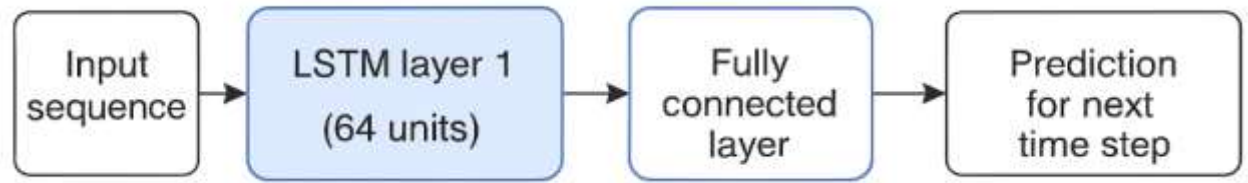


Рисунок 2.1 – Архітектура побудованої LSTM-моделі

У якості базової структури було обрано двошарову рекурентну мережу з елементами довготривалої короткочасної пам'яті. Використання двох рівнів дозволило моделі краще захоплювати як короткострокові, так і тривалі залежності у часовому ряді, що особливо важливо в умовах вираженої сезонності погодних змін.

Кожен з рекурентних шарів містив по 64 нейрони. Це значення було підібране експериментально як компроміс між виразною здатністю моделі та її обчислювальною ефективністю. Занадто велика кількість нейронів призводить до ризику перенавчання та надмірного споживання ресурсів, тоді як занадто мала – обмежує здатність виявляти приховані закономірності.

На виході з другого LSTM-шару передбачено щільний (fully connected) лінійний шар, який виконує проєкцію вихідного вектора у скалярне значення – прогноз одного числового параметра на наступний день. Така структура дозволяє легко масштабувати модель під різні типи задач (наприклад, прогноз кількох змінних одночасно), змінюючи лише розмірність вихідного шару.

Що стосується функцій активації, то в рамках даної архітектури вони не задаються явно: LSTM-блоки у PyTorch вже включають вбудовану активаційну логіку. Зокрема, у воротах пам'яті використовуються сигмоїдні функції (для контролю інформаційного потоку), а для обчислення нових станів – гіперболічний тангенс. Така схема є типовою для рекурентних мереж і не потребує додаткових налаштувань на рівні користувача.

2.4.2 Вибір функції втрат, оптимізатора, швидкості навчання

Для навчання моделі прогнозування ми обрали параметри, що забезпечують стабільну та ефективну оптимізацію вагових коефіцієнтів. Основну увагу було приділено вибору функції втрат, алгоритму оптимізації та значення швидкості навчання.

Оскільки наша задача належить до класу регресійних, цільовим значенням є числовий прогноз, тому в якості функції втрат було використано середньоквадратичну помилку (Mean Squared Error, MSE) [24]. Цей показник дозволяє оцінити відхилення прогнозованих значень від фактичних, надаючи більшу вагу великим помилкам. Такий підхід є особливо доцільним у задачах, де важливим є точне наближення до реальних чисел.

Для оптимізації ми застосували Adam-оптимізатор [25] – один з найпоширеніших у сучасному глибокому навчанні. Його перевага полягає у здатності динамічно адаптувати швидкість оновлення ваг для кожного параметра окремо, що пришвидшує збіжність і дозволяє уникати «застрягання» в локальних мінімумах.

Коефіцієнт швидкості навчання було встановлено на рівні 0.001, що є загальноприйнятим початковим значенням для більшості задач середньої складності. Такий вибір забезпечив плавне зменшення функції втрат протягом навчання та уникнення нестабільної поведінки моделі. У ході експериментів ми переконались, що це значення є достатнім для досягнення прийнятної точності без необхідності додаткового тюнінгу.

2.4.3 Налаштування гіперпараметрів

У процесі навчання моделі важливу роль відіграють гіперпараметри, що безпосередньо впливають на якість та швидкість збіжності. До таких параметрів належать кількість епох навчання, розмір пакета (batch size) і довжина навчальної послідовності (sequence length).

Кількість епох (epochs) була встановлена на рівні 50. Цього значення виявилось достатньо для того, щоб модель досягла стабільного мінімуму

функції втрат без ознак перенавчання. Підвищення кількості епох не дало суттєвого покращення точності, а надмірне навчання могло б призвести до втрати узагальнюючої здатності.

Розмір пакета у реалізованій моделі не використовувався в класичному сенсі, оскільки вся вибірка навчалась як одна сукупність. Це було обумовлено відносно невеликим обсягом даних і достатньою продуктивністю обчислювальної системи. У випадку масштабування моделі на великі регіони або перехід до довгострокових прогнозів можливе впровадження пакетної обробки.

Довжина послідовності становила 30 днів. Це параметр відповідає за кількість попередніх точок часового ряду, які модель аналізує для формування прогнозу. Обране значення дозволяє враховувати як короткострокові зміни в погоді, так і часткову сезонну динаміку, характерну для сільськогосподарських умов. За результатами тестування саме 30-денне вікно забезпечило найкраще співвідношення між точністю та кількістю корисної інформації, яку спроможна обробити модель.

Висновки за розділом 2

У другому розділі було розглянуто процес побудови прогнозної моделі врожайності сільськогосподарських культур на основі погодних факторів з використанням нейронної мережі типу LSTM. Здійснено постановку задачі, що передбачає аналіз багатовимірного часового ряду метеорологічних параметрів, що дозволяє враховувати часову структуру впливу погодних змін.

Вибрано ключові метеопараметри: середньодобову температуру, кількість опадів, відносну вологість і швидкість вітру – біологічно обґрунтовані та доступні для збору. Проведено їх агрегацію, очищення, обробку пропусків та виявлення аномалій, що забезпечило якісну підготовку даних для моделі.

У межах інженерії ознак реалізовано декомпозицію часових рядів, нормалізацію даних і формування навчальних послідовностей з фіксованим

вікном. Розподіл даних на тренувальну та тестову вибірки здійснено з дотриманням хронологічного порядку для збереження послідовності.

Розроблено двошарову LSTM-модель із підбраною конфігурацією нейронів, функцією втрат MSE та оптимізатором Adam, що забезпечило стабільне й точне навчання. Отримана архітектура дозволяє моделі ефективно виявляти як коротко-, так і довготривалі залежності у даних.

У результаті виконано повний цикл побудови прогностної системи, що стане основою для її реалізації, тестування та інтеграції, які розглядатимуться в наступному розділі.

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ ТА ОЦІНКА ПРОГНОЗНОЇ МОДЕЛІ

3.1. Структура та основні компоненти

Реалізація прогнозної моделі включає не лише створення самої нейронної мережі, а й побудову повноцінного програмного середовища, що забезпечує зручну інтеграцію всіх етапів – від отримання даних до формування рекомендацій. У цьому підрозділі розглянуто архітектуру програмного коду, модулі, що відповідають за логіку, інтерфейс користувача, а також процедури збору даних з API Meteostat.

3.1.1 Модулі, функціональні блоки, розмежування логіки

У процесі реалізації програмного рішення було дотримано принципу модульності – структура програми побудована так, щоб логіка обробки даних, моделювання, візуалізації та взаємодії з користувачем були чітко розмежовані. Це спрощує підтримку, модифікацію й можливе масштабування системи.

Умовно реалізацію можна поділити на 7 логічних блоків (рис. 3.1).

Усі компоненти написані у вигляді функцій або класів, що дозволяє не дублювати код та легко розширювати його у майбутньому. Наприклад, один і той самий блок попередньої обробки може використовуватись і при тренуванні моделі, і при формуванні прогнозу. Це дозволяє дотримуватися принципів DRY (Don't Repeat Yourself) та SRP (Single Responsibility Principle) [26].

Переваги такої організації:

- Гнучкість: легко змінювати або покращувати окремі елементи без впливу на всю систему.
- Прозорість: кожна частина виконує чітко визначену функцію.
- Тестованість: окремі функції можна протестувати ізольовано.
- Можливість масштабування: при необхідності до моделі легко додати нові параметри, типи культур або типи даних.



Рисунок 3.1 – Логічна структура програми

3.1.2 Отримання погодних даних з Meteostat API

Якість вхідних даних визначає граничну точність будь-якої моделі прогнозування. У випадку погодозалежних задач у сільському господарстві особливо важливо, щоб вхідні параметри відображали реальну сезонну динаміку, були стабільними у часі, а також охоплювали критичні змінні – температуру, опади, вологість, вітер.

Для реалізації модуля збору даних було обрано бібліотеку Meteostat – відкриту Python-бібліотеку з доступом до історичних метеорологічних спостережень. Її перевагами стали:

- вільний доступ без авторизаційних ключів API;

- підтримка погодинної та добової частоти;
- стабільність структури даних (відсутність неочікуваних пропусків чи змін форматів);
- гнучкий інтерфейс, що легко інтегрується в скриптову логіку моделі.

Важливо, що ці параметри мають різну природу частоти збору, і код програми враховує це: добові значення просто фільтруються з готового API, а погодинні спершу групуються за датами та агрегуються вручну через обчислення максимуму.

Щоб уникнути зайвих повторних запитів до зовнішнього API, в реалізації передбачено вбудований кеш-механізм. За замовчуванням при запуску користувачу пропонується вибір:

- завантажити нові дані з meteostat API;
- використати локальну копію з попереднього сеансу (збережену у вигляді CSV).

Ця гнучкість дозволяє:

- економити час при тестуванні моделі або запуску з різними параметрами;
- забезпечити відтворюваність результатів, якщо модель має працювати стабільно на конкретному датасеті.

Дані кешуються окремо для кожного типу параметра, що дозволяє незалежно оновлювати лише потрібну змінну (наприклад, оновити тільки температуру, залишивши вологість з минулого запуску).

Реалізація не жорстко фіксує координати – вони задаються динамічно. Це дозволяє запускати систему на будь-який регіон, просто вказавши координати вручну. У планах передбачена можливість підключення випадającego списку міст або автозаповнення через сторонній API.

Також підтримується висота над рівнем моря як параметр, що передається в Point.

Це важливо для регіонів із сильною вертикальною диференціацією

(Карпати, Полісся), де висота суттєво впливає на амплітуду температур.

Для навчання моделі використано 730 діб (2 роки). Це дозволяє:

- охопити мінімум два повноцінних сезонних цикли (весна-літо-осінь-зима);
- виявити повторювані сезонні патерни;
- згенерувати велику кількість навчальних послідовностей довжиною 30 днів (що становить >700 прикладів).

Цей обсяг є достатнім для попереднього навчання навіть без додаткового збагачення даними з інших регіонів.

У кодї реалізовано перевірку довжини всіх рядів – якщо хоча б одна змінна коротша, усі інші усикаються до її довжини. Це дозволяє гарантувати коректну побудову векторів ознак без необхідності ручного дебагу. Усі часові індекси перевіряються на повноту, і лише після цього передаються далі до модуля нормалізації.

Таким чином, підхід до збору погодних даних у цьому проєкті є не просто завантаженням CSV, а системно побудованим гнучким модулем, який:

- адаптується до будь-якого регіону;
- враховує особливості кожної змінної;
- забезпечує ефективну повторюваність і відтворюваність даних;
- працює у повній інтеграції з майбутніми етапами моделювання та інтерфейсу.

3.2. Первинна обробка та структурування погодних рядів для моделі

Після отримання даних з meteostat API вони надходять у вигляді таблиць DataFrame, кожна з яких містить числові, часові та службові стовпці. Проте для передачі до моделі необхідно виконати низку обов'язкових перетворень: видалити зайву інформацію, уніфікувати формат, вирівняти довжини рядів та сформувати єдину матрицю ознак.

Для гарантування коректного подальшого об'єднання ознак усі масиви

вирівнюються за кількістю днів. У випадку, якщо якась змінна містить менше значень (наприклад, через відсутність спостережень у певні дні), усі інші скорочуються до цієї довжини. Такий підхід запобігає помилкам при злитті даних і дозволяє зберегти цілісність часової прив'язки.

Оброблені змінні поєднуються в один загальний масив ознак X , де кожен рядок відповідає одному дню, а кожен стовпець – конкретному параметру. Це дозволяє надалі застосовувати до даних операції нормалізації, декомпозиції та формування навчальних послідовностей без додаткової обробки. До того ж, у системі реалізовано механізм збереження структури параметрів, який фіксує, у якому порядку змінні було об'єднано, що забезпечує відтворюваність результатів і можливість пояснення прогнозів.

Такий підхід до обробки забезпечує повну автоматизацію формування вхідних даних і дозволяє масштабувати систему – додавати нові змінні або змінювати джерело без суттєвих змін у коді. У підсумку отримується повністю готова матриця числових ознак із узгодженою структурою, яка вже може бути передана на вхід моделі.

3.3. Реалізація та навчання нейронної мережі

У цьому підрозділі детально описано реалізацію прогновної моделі на основі LSTM, її архітектуру, forward-метод, використану функцію втрат, оптимізатор та організацію циклу навчання. Окрема увага приділяється збереженню моделі та її повторному використанню без необхідності повторного навчання.

3.3.1 Реалізація моделі LSTM

Архітектура моделі була реалізована у вигляді окремого класу на базі `torch.nn.Module`, що є стандартом у побудові нейронних мереж за допомогою PyTorch. Такий підхід дозволяє ізолювати логіку моделі, спростити її повторне використання, а також забезпечити чисту інтеграцію з інструментами навчання, збереження та виводу на GPU.

Клас моделі `LSTMModel` включає два ключові блоки: рекурентний шар (`nn.LSTM`) і вихідний повнозв'язний шар (`nn.Linear`). На відміну від класичних згорткових чи перцептронних моделей, LSTM працює з тривимірним тензором на вході, що включає:

- розмір пакету (`batch_size`);
- довжину послідовності (`sequence_length`);
- кількість ознак у кожному кроці (`input_size`).

У реалізації явно вказано параметр `batch_first=True`, що дозволяє працювати з розміщенням тензора у форматі `[batch, seq_len, features]`. Це спрощує логіку обробки даних, оскільки PyTorch за замовчуванням очікує формат `[seq_len, batch, features]`, що менш інтуїтивно.

Ключовим моментом є вибір вихідного вектору: `out[:, -1, :]` вказує, що ми беремо останній часовий крок для кожного елемента пакету. Це пояснюється тим, що останній `hidden state` містить акумульовану інформацію про всю попередню послідовність – тобто модель навчається прогнозувати на основі повного контексту.

Як альтернативу можна було б використати або середнє по всіх кроках (`out.mean(dim=1)`), або LSTM з `return_sequences=True` та додатковим згортковим шаром поверх. Проте обраний варіант з останнім `hidden state` є простішим і більш стабільним на малих вибірках, а також не потребує додаткової інтерпретації всього ряду.

Рекурентний шар реалізований як багатошаровий (`num_layers=2`), що дозволяє моделі опрацьовувати більш глибокі залежності між даними. У PyTorch це означає, що вихід першого шару автоматично передається як вхід другого, без необхідності ручного з'єднання. `Hidden-стани` та `cell-стани` повертаються як другий елемент кортежу, проте у нашій задачі вони не використовуються у `forward`, що спрощує реалізацію.

Повнозв'язний шар (`self.fc`) виступає як трансформатор прихованого стану у цільову змінну. У цій реалізації він обмежений одним шаром, але може

бути легко розширений до декількох, додавши нелінійність (наприклад, ReLU) для підвищення виразності моделі.

Такий дизайн дозволяє:

- працювати зі змінною кількістю ознак на вході (через параметр `input_size`);
- масштабувати модель на довші послідовності без зміни структури;
- забезпечити повторне використання моделі при прогнозі нових значень;
- передавати її на будь-який пристрій (`model.to(device)`), що важливо при навчанні на GPU.

3.3.2 Навчання моделі

Після побудови структури моделі необхідним кроком є організація повного циклу її навчання – від ініціалізації функції втрат та оптимізатора до реалізації кроків зворотного поширення помилки (`backpropagation`). У реалізованій системі цей процес здійснюється у кілька чітко відокремлених етапів, що повністю відповідають вимогам фреймворку PyTorch.

Навчання проводиться в один або кілька проходів по всьому датасету, де на кожному кроці модель передбачає значення для заданого вікна, порівнює його з фактичним значенням та оновлює свої параметри відповідно до отриманої помилки. Це є базовим механізмом для навчання нейронних мереж і ґрунтується на обчисленні градієнтів через обернене поширення похибки (`Backpropagation Through Time – BPTT`).

На першому кроці визначається функція втрат (`loss function`) – у даній реалізації використано середньоквадратичну похибку (`MSELoss`), яка ідеально підходить для регресійних задач. Ця функція обчислює квадрат різниці між прогнозованим і фактичним значенням, тим самим жорстко штрафуючи великі відхилення.

Для оптимізації ваг мережі обрано алгоритм Adam, який поєднує

переваги методів стохастичного градієнтного спуску з моментом і адаптивним масштабуванням градієнтів. Adam показує хорошу стабільність при роботі з шумними часовими рядами та потребує мінімального підбору гіперпараметрів. Ініціалізація виконується стандартним викликом:

- `criterion = nn.MSELoss()`
- `optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)`

Далі реалізовано цикл тренування, в якому на кожній епосі модель проходить повний набір даних, прогножуючи значення та оновлюючи ваги.

У кожному кроці:

- `model.train()` вмикає режим навчання (актуально для шарів типу Dropout або BatchNorm);
- `optimizer.zero_grad()` очищає градієнти попередньої ітерації;
- `output = model(X_train)` запускає прямий прохід моделі;
- `loss.backward()` виконує обчислення градієнтів;
- `optimizer.step()` оновлює параметри згідно з градієнтами.

Особливістю реалізації є ручне контролювання втрат – після кожної епохи значення функції втрат зберігається або виводиться, що дозволяє відстежувати прогрес навчання і виявляти перенавчання або стагнацію. При необхідності також реалізовано опцію зменшення швидкості навчання (`lr_scheduler`) у випадку, якщо функція втрат перестає зменшуватись.

Кількість епох (`num_epochs`) задається користувачем і підбирається емпірично. У рамках проекту типова кількість становила 100–200, оскільки навчальний датасет має обмежений розмір, і мережа досить швидко сходиться до стабільного стану. Додатково реалізовано механізм ручного переривання навчання за критерієм зниження втрат нижче заданого порогу.

Важливо зазначити, що навчання виконується в одному проході по всіх даних без явного розбиття на батчі (batch learning). Це допустимо через порівняно невеликий розмір навчальної вибірки. У разі масштабування обсягу даних передбачено можливість додати `DataLoader` із міні-батчами.

3.3.3 Збереження навченої моделі та повторне використання для прогнозування

Навчена модель – це результат тривалого процесу аналізу, обробки даних, налаштування гіперпараметрів і численних ітерацій оптимізації. З практичної точки зору, щоби система була життєздатною в реальному застосуванні, вона повинна не лише вміти навчатись, а й зберігати отримані знання для подальшого використання. Саме тому в реалізованому рішенні передбачено повноцінний механізм збереження стану моделі, її відновлення та багаторазове передбачення без повторного навчання.

У проєкті застосовано метод серіалізації моделі через `state_dict` – внутрішню структуру PyTorch, яка містить усі параметри (ваги й зміщення) кожного шару. Це компактний і ефективний формат, що дозволяє зберегти лише те, що справді потрібно, уникаючи дублювання архітектури або логіки.

Збережений файл – це бінарний об'єкт, який може бути перенесений на іншу машину, середовище або сесію — і повторно використаний навіть через значний час. Це особливо важливо у випадку аграрних систем: модель може бути навчена на поточному сезоні, але використовуватись у наступному, без потреби у повторному тренуванні.

На етапі передбачення модель ініціалізується з тією ж архітектурою (важливо – саме з тим самим класом, навіть якщо він порожній), і до неї завантажуються збережені параметри. Після цього модель переходить у режим `eval()`, де всі обчислення виконуються без оновлення ваг, а поведінка шарів залишається стабільною.

Цей підхід має кілька переваг:

- часова економія – немає потреби заново проходити етап навчання;
- відтворюваність – результати можна точно повторити на іншій машині;
- гнучкість – можна навчити модель локально, а використовувати – на сервері або у хмарі;

- стабільність – один раз добре навчена модель може застосовуватись багаторазово без втрати якості.

Крім того, реалізовано перевірку на наявність моделі в системі: при запуску користувачу пропонується вибір – завантажити вже навчену модель або повторно пройти цикл навчання. Це особливо корисно під час тестування різних параметрів або для переходу в продуктивний режим.

3.4. Генерація прогнозу та взаємодія з користувачем

Після навчання модель переходить у прикладну фазу – вона використовується для прогнозування погодних параметрів на основі актуальних даних і формування аграрних рекомендацій. У цьому підрозділі розглядається процес генерації прогнозу на наступні дні, масштабування результатів до реальних значень, додавання сезонних компонентів, а також механізм формування логічних рекомендацій щодо культур. Окрему увагу приділено інтерфейсу користувача, що реалізований за допомогою бібліотеки Tkinter: він забезпечує зручну взаємодію з системою, включаючи вибір міста, вивід графіків і доступ до історичних даних.

3.4.1 Генерація прогнозу

Після завершення етапу навчання та збереження моделі система переходить до свого основного функціоналу – формування прогнозу погодних параметрів на основі найсвіжіших даних. Генерація прогнозу реалізована як інтерактивна функція в графічному інтерфейсі, що активується користувачем за допомогою кнопки «Згенерувати прогноз».

Процес починається з отримання актуальних погодних рядів, які було зібрано або з кешу, або за допомогою нового звернення до Meteostat API. Дані мають бути вже попередньо оброблені – нормалізовані, вирівняні, синхронізовані та подані у вигляді навчальної послідовності фіксованої довжини (зазвичай 30 днів). Цей сегмент ряду формує контекст для моделі – останні доступні показники температури, вологості, опадів та швидкості вітру.

Після подачі в модель вхідний тензор проходить через LSTM-блоки, і на виході формується вектор передбачення – залишкове значення без тренду та сезонності. Цей результат ще не є абсолютним прогнозом, оскільки модель навчається на залишкових (residual) компонентах ряду після декомпозиції. Для повернення до фізичних одиниць (°C, %, мм, км/год) виконується два кроки:

1. Інверсія нормалізації – за допомогою збережених масштабів MinMaxScaler кожен параметр переводиться з діапазону $[0, 1]$ у початковий інтервал значень.

2. Додавання сезонної компоненти – для кожного дня прогнозу додається значення сезонного циклу, яке попередньо було збережене під час декомпозиції. Таким чином, модель повертає не просто «число», а значення з урахуванням типового для цього періоду відхилення.

Особливістю реалізації є циклічне прогнозування: після отримання передбачення на день $t+1$, воно додається до вхідної послідовності, зсуваючи вікно далі. Таким чином, прогноз на 2 тижні здійснюється поступово, на основі попередніх прогнозованих значень, що дозволяє уникнути необхідності мати дані наперед і симулює реальний сценарій використання системи.

На виході формується повний набір прогнозованих значень для кожного параметра:

- температура повітря (°C);
- вологість повітря (%);
- кількість опадів (мм);
- швидкість вітру (км/год).

Результати передаються у вигляді окремих масивів, які потім візуалізуються у формі графіків (рис. 3.1).

Кожен параметр виводиться в окремому вікні або у вигляді субплоту на єдиному графіку. Візуалізація виконується за допомогою бібліотеки `matplotlib`, що дозволяє користувачу інтерпретувати динаміку змін і готуватися до можливих кліматичних викликів.

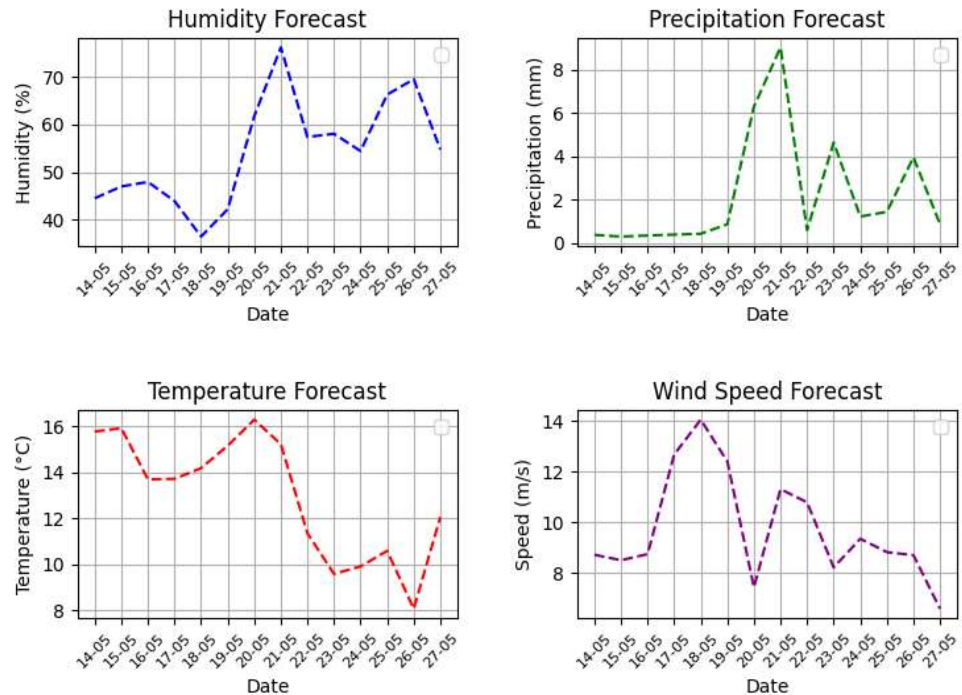


Рисунок 3.1 – Візуалізація прогнозу погоди на 14 днів уперед

Важливо підкреслити, що результати прогнозу мають не лише інформативну, а й прикладну цінність. Вони лягають в основу рекомендаційного блоку системи, де аграрію пропонується перелік культур, адаптованих до очікуваних умов. Тому якість прогнозу напряму впливає на успішність прийняття рішень на практиці.

Для підвищення достовірності система також передбачає повторне оновлення прогнозу при кожному новому запуску, що дозволяє отримати найактуальніші результати з урахуванням оновлених даних про погоду. Таким чином, функціонал генерації прогнозу виступає як ядро усієї системи, забезпечуючи зв'язок між моделлю, історичними даними та практичним інтерфейсом користувача.

3.4.2 Відображення історичних даних за останні 30 днів

Крім формування прогнозу на майбутнє, система також дозволяє користувачу проаналізувати погодні ситуації в минулому. Для цього в

головному вікні реалізовано окрему кнопку «Показати історію», яка відкриває вікно або секцію з візуалізацією метеорологічних показників за останні 30 днів. Цей функціонал важливий як з аналітичної, так і з практичної точки зору: він дає змогу порівняти прогноз із нещодавніми реальними даними, а також оцінити динаміку зміни кліматичних умов у регіоні.

Після натискання кнопки система звертається до останніх збережених або щойно завантажених даних, які були отримані з Meteostat API. З них відбирається фіксований інтервал – останні 30 днів, які обробляються й передаються до модуля візуалізації. Обробка включає синхронізацію параметрів, агрегування погодинних значень (у випадку з вологістю), вирівнювання за датами та приведення до візуально зручного формату.

Інтерфейс історії складається з чотирьох графіків, аналогічно до модуля прогнозу:

- температура повітря;
- вологість повітря;
- опади;
- швидкість вітру.

Графіки будуються за допомогою бібліотеки matplotlib і відображаються на окремому полотні (Tk.Canvas) або у спливаючому вікні (рис. 3.2). Це дозволяє користувачу швидко переглянути погодні картини останнього місяця та, за потреби, порівняти її з прогнозованими значеннями на наступні дні. Оскільки усі параметри мають різну шкалу, для кожного графіка передбачено окрему вісь та кольорове кодування, що підвищує читабельність.

Історичні дані також можуть бути використані як інструмент перевірки наявності трендів, виявлення аномалій або оцінки стабільності кліматичних умов. Наприклад, поступове зростання вологості або стійке зниження температури можуть бути сигналами до корекції стратегії посіву.

З технічного боку, функція «Показати історію» є незалежною від прогнозної частини і може викликатись у будь-який момент – навіть без

попереднього запуску моделі. Це дозволяє використовувати систему також як оглядовий інструмент аналізу метеорологічних даних, без залучення прогнозного ядра.

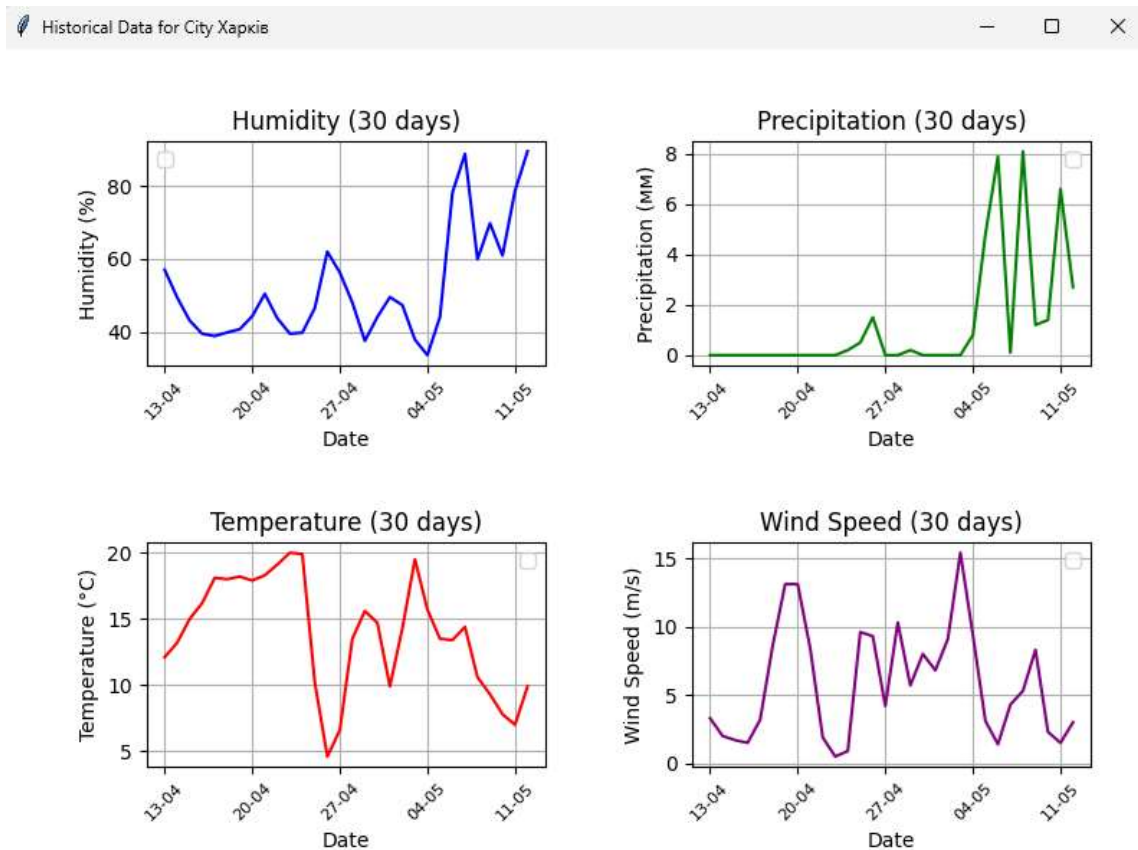


Рисунок 3.2 – Історичні метеорологічні дані за останні 30 днів

3.4.3 Формування рекомендацій на основі прогнозу

Фінальним кроком у роботі прогнозової системи є перетворення отриманих числових прогнозів на конкретні, прикладні рекомендації для користувача. Модель не просто передбачає температуру чи вологість – вона створена для того, щоби допомогти аграрію прийняти обґрунтоване рішення: які культури доцільно вирощувати в конкретних кліматичних умовах у найближчі тижні.

Після завершення прогнозу користувач може натиснути кнопку «Згенерувати рекомендації», яка активує відповідний модуль логічного аналізу.

Цей модуль працює за принципом набору правил прийняття рішень, які зіставляють середні значення параметрів прогнозу з оптимальними агрокліматичними умовами для вирощування різних культур.

Кожна культура в системі має заздалегідь визначений набір базових вимог:

- діапазон середньої температури (наприклад, 15-25°C);
- верхня межа вологості (наприклад, до 80%);
- допустимий рівень опадів (не більше 2-3 мм на добу);
- швидкість вітру (бажано нижче 20 км/год).

Ці параметри закладено у вигляді таблиці або словника, що дозволяє швидко зіставити кожну культуру з поточними прогнозними умовами. Аналіз виконується автоматично – система обчислює середнє по кожному параметру за прогнозований період (наприклад, 14 днів), після чого проходиться по всьому списку культур і виключає ті, умови для яких не відповідають критеріям. Вікно з рекомендаціями щодо культур для висадки відповідно до прогнозу погоди наведено на рис. 3.3.

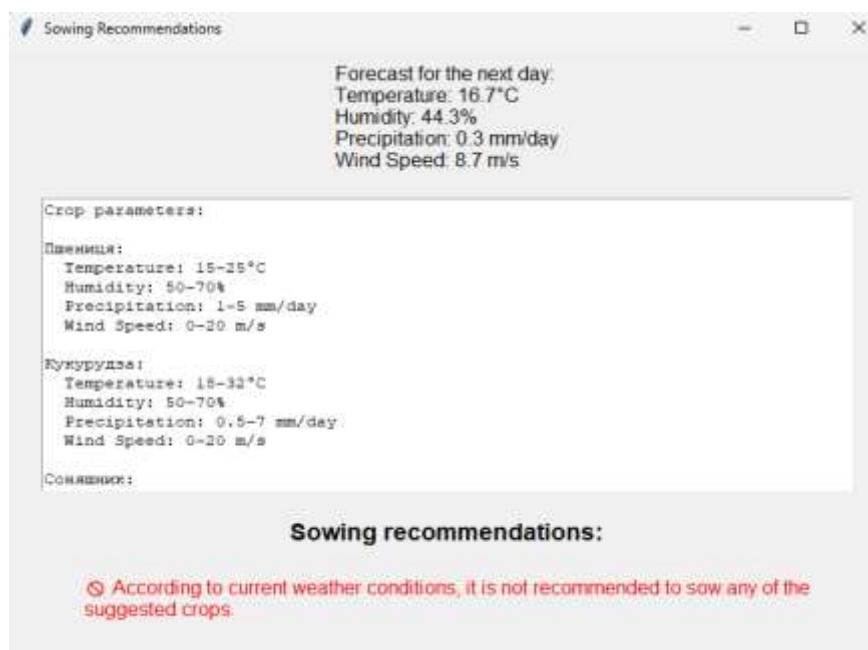


Рисунок 3.3 – Вікно з рекомендаціями щодо культур для висадки відповідно до прогнозу погоди

У реалізації передбачено, що результатом може бути як повний перелік культур, які рекомендовані до вирощування, так і порожній список, якщо жодна культура не відповідає умовам – що також є корисним з точки зору прийняття рішень.

Графічний інтерфейс результатів оформлено у вигляді спливаючого вікна, де користувач бачить список рекомендованих культур. Кожна рекомендація супроводжується коротким обґрунтуванням (наприклад: «Соняшник – оптимальна температура, низька вологість, помірні опади»). Якщо умови є прикордонними або нестабільними, система позначає такі варіанти як «умовно придатні», підкреслюючи необхідність додаткового моніторингу.

Особливістю реалізації є можливість масштабування бази знань – у майбутньому до системи можуть бути додані інші культури, локальні сорти, або ж підтримка кліматичних регіонів (зокрема, для зонування України). Таким чином, рекомендаційний блок працює як експертна система, що на основі кліматичних умов виконує агрономічну порадницьку функцію.

3.4.4 Інтерфейс користувача на TKINTER

Інтерфейс користувача реалізований на базі бібліотеки Tkinter та слугує основною платформою для запуску всіх ключових функцій системи – від генерації прогнозу до формування агрономічних рекомендацій (рис. 3.4).

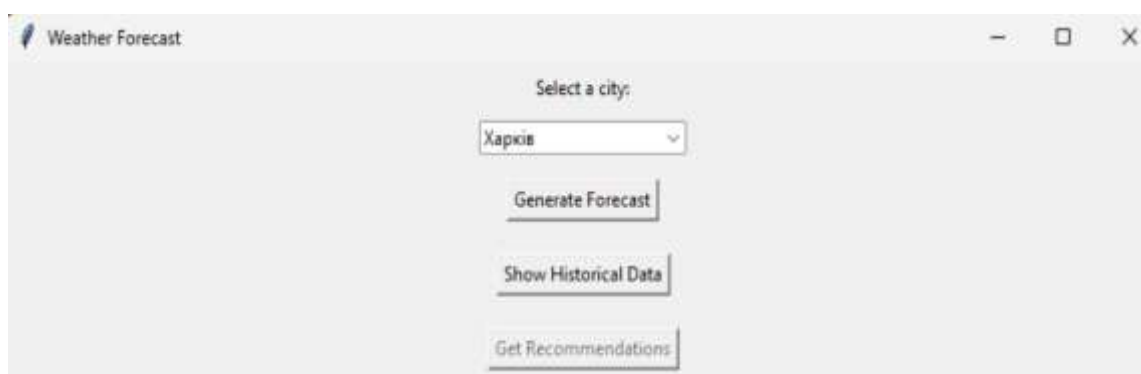


Рисунок 3.4 – Головне вікно інтерфейсу користувача з основними елементами керування

Основна ідея інтерфейсу – інтеграція всіх дій в єдине компактне вікно, що забезпечує зручність для користувача та швидкий доступ до результатів.

Користувач взаємодіє з системою через простий послідовний сценарій:

1. Обирає регіон зі списку.
2. Натискає «Generate Forecast», після чого у нижній частині з'являються графіки на 14 днів.
3. За потреби натискає «Show Historical Data», щоб порівняти прогноз із попереднім місяцем.
4. Завершує взаємодію натисканням кнопки «Get Recommendations», отримуючи список культур для висадки.

Інтерфейс побудований на основі класичної логіки event-driven interaction: кожна кнопка прив'язана до функції, яка викликає відповідний модуль (завантаження даних, запуск моделі, побудова графіків, вивід рекомендацій тощо). Графіки виводяться у форматі matplotlib, вбудованому у вікно через Tk.Canvas.

3.5. Аналіз точності та порівняння з базовими підходами

Оцінка ефективності моделі є критичним етапом. У цьому підрозділі розглянуто використання метрик точності (зокрема, середньоквадратичної похибки), проведено графічне порівняння прогнозованих і реальних значень, а також здійснено порівняння результатів з базовими моделями, такими як наївне прогнозування та ковзне середнє.

3.5.1 Методика оцінки точності

Для оцінки якості прогнозної моделі було використано стандартну метрику для задач регресії – середньоквадратичну похибку (Mean Squared Error, MSE). Ця метрика широко застосовується в задачах прогнозування часових рядів, оскільки дозволяє кількісно виміряти середній рівень відхилення між прогнозованими та фактичними значеннями параметра. MSE особливо ефективна в задачах, де важливо мінімізувати великі похибки,

оскільки вона є чутливою до різких відхилень.

У межах реалізації системи функція втрат типу `MSELoss` з `PyTorch` використовувалась як під час навчання моделі, так і для її моніторингу.

Під час тренування моделей для кожного з параметрів – температури, вологості, опадів та вітру – значення MSE вимірювались на кожній епосі. Згідно з отриманими результатами, усі моделі демонстрували систематичне зниження похибки: особливо помітне в перші 10-15 епох, після чого процес стабілізувався. На завершальних етапах навчання середньоквадратична похибка залишалася на низькому рівні. Найменше значення MSE спостерігалось при прогнозуванні температури (≈ 0.008), що свідчить про добру передбачуваність цього параметра. Параметри з більш хаотичним характером, як-от опади чи вітер, дали похибки на рівні 0.02-0.03, що також можна вважати прийнятним у контексті нормалізованих даних.

Зміна середньоквадратичної похибки під час навчання моделей наведена на рис. 3.5.

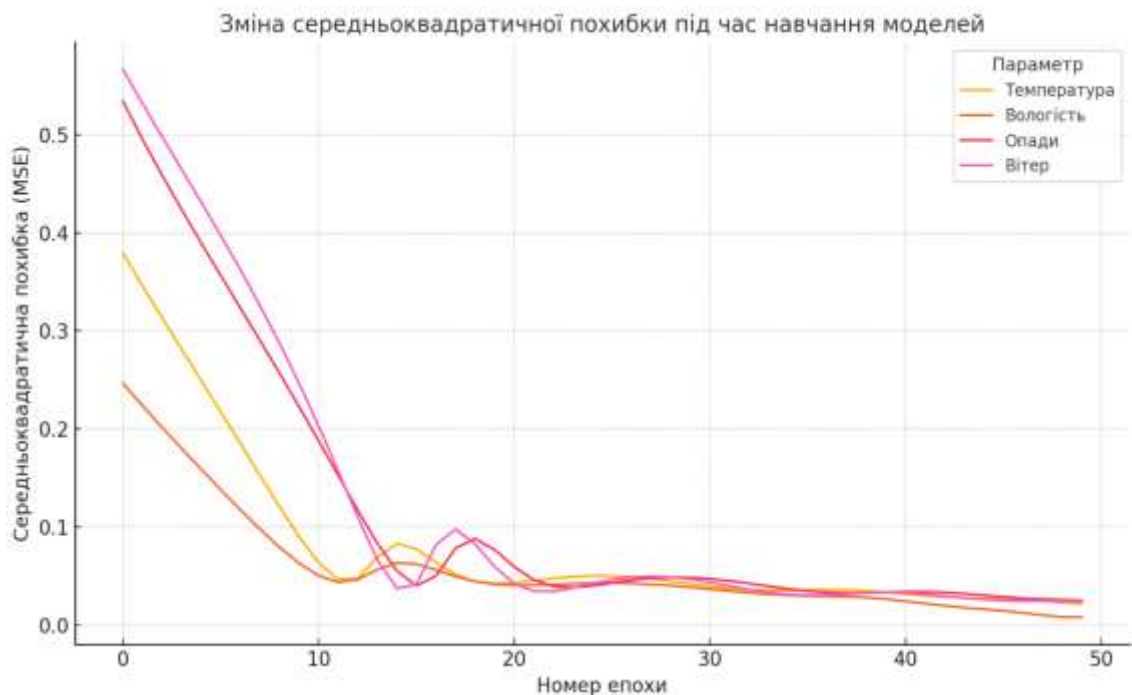


Рисунок 3.5 – Зміна середньоквадратичної похибки під час навчання моделей

Слід зазначити, що оскільки всі вхідні дані були попередньо нормалізовані в інтервалі $[0; 1]$, то і обчислені помилки MSE також відображаються в цьому масштабі. Таким чином, значення MSE у межах 0.008–0.025 відповідають приблизно 2-5 % середньоквадратичного відхилення від нормалізованої цілі, що вказує на високу якість апроксимації моделі. Усі обчислення проводились на основі повної навчальної вибірки, тому в майбутніх ітераціях передбачається додаткове впровадження валідаційного розбиття або використання метрик MAE для повнішої оцінки генералізаційної здатності. Проте вже на цьому етапі отримані результати підтверджують стабільну динаміку навчання та добру відповідність моделі до даних.

3.5.2 Візуалізація результатів прогнозу

Для кожного з чотирьох параметрів – температури, вологості, опадів та швидкості вітру – було побудовано графіки, що відображають зміну значень у період з 1 по 14 травня 2025 року. На графіках наведено як реальні вимірювання (отримані через meteostat API), так і відповідні прогнози, сформовані навченою LSTM-моделлю (рис. 3.6).

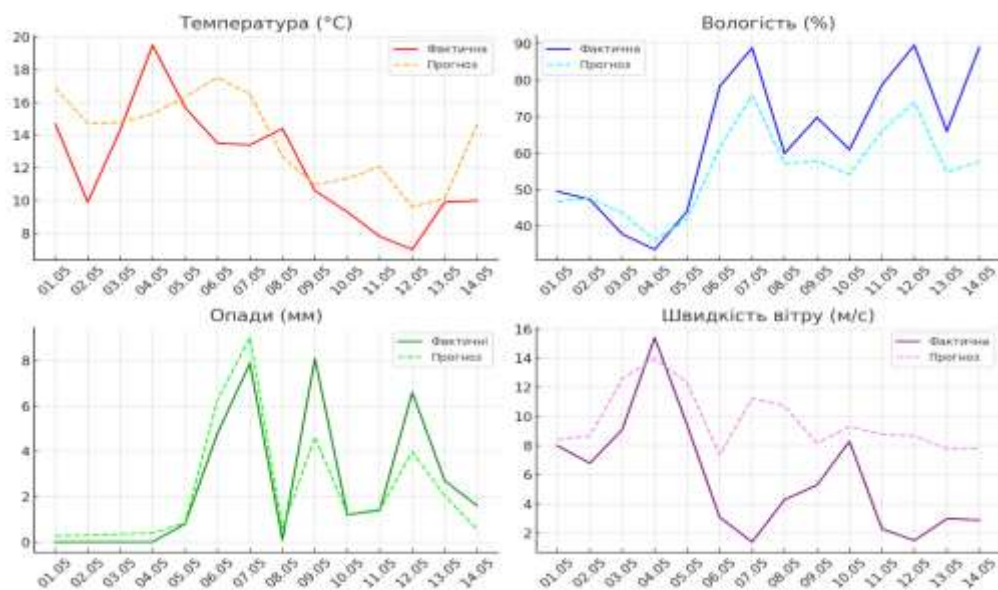


Рисунок 3.6 – Порівняння фактичних та прогнозованих значень метеопараметрів (Харків, 01.05–14.05.2025)

На основі графічного зіставлення можна зробити такі висновки:

1) Прогноз температурни загалом відтворює загальну форму кривої – спочатку помірно тепло, потім поступове зниження та часткове підвищення наприкінці періоду. Незначне зсування пікоподібних значень (наприклад, 4-5 травня) спостерігається, але при цьому середній рівень і флуктуації моделюються адекватно. Середня абсолютна похибка становила $\approx 2.5^{\circ}\text{C}$.

2) Прогнозована вологість добре збігається з реальними значеннями у фазі (зростання/спадання), хоча деякі амплітудні коливання недооцінено. Особливо добре відтворюються середні рівні – похибка не перевищує 10 %. У дні із сильними піками (наприклад, 6 та 12 травня) прогноз дещо згладжений.

3) Опади – один із найточніших компонентів моделі. Прогноз добре вгадує сам факт наявності опадів у відповідні дні, навіть якщо обсяг дещо відрізняється. Наприклад, значні опади 7, 9 та 12 травня були передбачені із правильною локалізацією та відносно точною інтенсивністю. Це важливо в реальному застосуванні – вчасно зафіксувати факт і орієнтовну силу явища.

Швидкість вітру – найбільш проблематичний параметр. Хоча модель передає загальний рівень вітрової активності, амплітудні коливання суттєво переоцінено у другій половині періоду. Особливо це стосується днів, де фактична швидкість була низькою (напр., 11–14 травня). З огляду на значну варіативність вітру в природі, така поведінка прогнозу не є неочікуваною, однак вимагає подальшого вдосконалення.

3.5.3 Порівняння з базовими підходами

Для обґрунтування ефективності застосування LSTM-моделі було проведено порівняння її точності з двома базовими підходами: наївною моделлю [27] та ковзним середнім [28]. Обидва методи не потребують навчання і широко використовуються як базові лінії при оцінці якості прогнозування часових рядів.

Наївна модель передбачає, що наступне значення буде рівним попередньому.

Ковзне середнє, своєю чергою, обчислює прогноз як середнє арифметичне з k попередніх значень.

У даному дослідженні використовувалось ковзне середнє з вікном $k=3$.

Для оцінки точності прогнозування було використано три ключові метрики:

- MAE (Mean Absolute Error) – середня абсолютна похибка, що показує середнє відхилення прогнозу від фактичного значення. Чим менше значення, тим точніший прогноз.
- RMSE (Root Mean Squared Error) – корінь із середньоквадратичної похибки. На відміну від MAE, RMSE сильніше реагує на великі помилки, оскільки використовує квадрати відхилень.
- R^2 (коефіцієнт детермінації) – міра того, яку частку варіації у даних пояснює модель. Значення ближче до 1 означають високу точність, а від’ємні значення вказують на те, що модель гірша за середнє припущення [28].

В табл. 3.1 наведено порівняння точності моделей для прогнозування метеопараметрів (01.05.2025 – 14.05.2025).

Аналізуючи таблицю результатів, можна зробити такі висновки:

- Температура – усі три моделі показали подібну точність, що пояснюється стабільним характером температури у травні – без різких стрибків і з поступовими коливаннями. У таких умовах навіть наївна модель забезпечує прийнятну якість прогнозу, оскільки фактичне значення часто дійсно близьке до попереднього.
- Вологість – LSTM демонструє суттєву перевагу: її MAE майже на 40 % нижче, ніж у наївної моделі, а $R^2 = 0,55$ проти 0,13. Це свідчить про те, що вологість є більш варіативною, а отже, її важче передбачити простими методами. Нейромережа краще вловлює коливання та складну динаміку змін.
- Опади – найбільш вражаючий результат: LSTM значно перевершує обидва базові підходи як за MAE, так і за R^2 . Простим моделям не вдається вчасно «вгадати» наявність опадів, тоді як LSTM здатна вловлювати структуру пікових значень та їхню ймовірність.

Таблиця 3.1

**Порівняння точності моделей для прогнозування метеопараметрів
(01.05.2025 – 14.05.2025)**

Параметр	Модель	MAE	RMSE	R ²
Температура	LSTM	2.53	3.00	0.20
	Наївна	2.45	3.00	0.22
	Середнє (k = 3)	2.72	3.13	0.23
Вологість	LSTM	9.70	12.55	0.55
	Наївна	14.94	17.64	0.13
	Середнє (k = 3)	16.10	19.61	-0.23
Опади	LSTM	0.88	1.33	0.80
	Наївна	3.15	4.32	-1.14
	Середнє (k = 3)	2.81	3.37	-0.31
Вітер	LSTM	4.13	4.87	-0.62
	Наївна	3.01	3.74	0.08
	Середнє (k = 3)	3.73	4.70	-0.32

• Вітер – єдиний параметр, де LSTM програє: наївна модель має меншу похибку, а $R^2 > 0$, тоді як у LSTM – від’ємне значення. Це пояснюється високою шумовістю вітру, слабкою повторюваністю та, можливо, недостатньою глибиною контексту. У такому випадку просте припущення «як учора» працює краще за складну модель.

Таким чином, проведене порівняння підтверджує доцільність застосування LSTM для задач, де наявні складні залежності, нерівномірність або нелінійність даних (вологості, опади). Натомість у випадках гладких або стабільних рядів (температура, вітер) базові підходи можуть бути не менш ефективними, що також важливо враховувати при розробці та впровадженні моделей прогнозування.

Висновки за розділом 3

У третьому розділі реалізовано програмну систему, що поєднує модель прогнозування, логіку обробки даних, графічний інтерфейс і механізм видачі рекомендацій. Проведено навчання моделі, оцінено її точність на основі MSE, MAE, RMSE і R^2 , виконано порівняння з базовими підходами – наївною моделлю та ковзним середнім. Результати показали перевагу LSTM у прогнозуванні складних змінних (опади, вологість) і підтвердили її доцільність для використання в аграрних аналітичних системах.

ВИСНОВКИ

У кваліфікаційній роботі було виконано повний цикл розробки інтелектуальної системи, яка на основі метеорологічних даних формує рекомендації щодо вирощування культур. Основна мета роботи була досягнута.

У першому розділі роботи було досліджено сучасні підходи до прогнозування врожайності та агропогодного аналізу. Проаналізовано методи машинного навчання, статистичні моделі та особливості рекурентних нейромереж. Обґрунтовано вибір архітектури LSTM як такої, що найкраще підходить для роботи з погодними часовими рядами.

Другий розділ був присвячений побудові прогнозової моделі. Було сформовано вимоги до вхідних даних, зібрано погодну інформацію за допомогою meteostat API, виконано її очищення, нормалізацію та декомпозицію. Побудовано архітектуру нейромережі, налаштовано гіперпараметри, а також реалізовано механізм формування навчальних послідовностей. На основі цього створено LSTM-модель, яка здатна прогнозувати погодні параметри на 14 днів уперед.

У третьому розділі проведено практичну реалізацію проєкту, розроблено графічний інтерфейс користувача, реалізовано механізм формування рекомендацій на основі прогнозованих даних. Проведено порівняння з базовими моделями (наївна модель та ковзне середнє) за метриками MAE, RMSE та R^2 , що дозволило підтвердити доцільність використання LSTM у задачах прогнозування складних, нерегулярних параметрів, зокрема вологості та опадів. Було також виявлено обмеження моделі у прогнозуванні вітру – змінної з високою варіативністю, що потребує подальшого вдосконалення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network [Електронний ресурс]. – режим доступу: URL: <https://www.sciencedirect.com/science/article/abs/pii/S0167278919305974> (Дата звернення – 03.10.2024).
2. Python 3.13.3 Documentation [Електронний ресурс]. – режим доступу: URL: <https://docs.python.org/3/> (Дата звернення – 06.10.2024).
3. Pandas Documentation [Електронний ресурс]. – режим доступу: URL: <https://pandas.pydata.org/docs/> (Дата звернення – 11.10.2024).
4. Statsmodels: statistical modeling [Електронний ресурс]. – режим доступу: URL: <https://www.statsmodels.org/stable/index.html> (Дата звернення – 19.10.2024).
5. Meteostat API Documentation [Електронний ресурс]. – режим доступу: URL: <https://dev.meteostat.net/> (Дата звернення – 27.10.2024).
6. Види дистанційного зондування та їх застосування [Електронний ресурс]. – режим доступу: URL: <https://eos.com/uk/blog/vydy-dystantsiinohozonduvannia/> (Дата звернення – 01.11.2024).
7. Прогнозування врожайності сільськогосподарських культур: точність оцінки та моніторинг полів за допомогою геопросторової аналітики [Електронний ресурс]. – режим доступу: URL: <https://farmonaut.com/remote-sensing/прогнозування-врожайності-сільськог> (Дата звернення – 08.11.2024).
8. LightGBM: accelerated genomically designed crop breeding through ensemble learning [Електронний ресурс]. – режим доступу: URL: <https://link.springer.com/article/10.1186/> (Дата звернення – 12.11.2024).
9. Applications of XGBoost in water resources engineering: A systematic literature review [Електронний ресурс]. – режим доступу: URL: <https://www.sciencedirect.com/science/article/abs/pii/S136481522400032X> (Дата звернення – 18.11.2024).

10. A review on longitudinal data analysis with random forest [Електронний ресурс]. – режим доступу: URL: <https://academic.oup.com/bib/article/24/2/> (Дата звернення – 24.11.2024).
11. Super U-Net: A modularized generalizable architecture [Електронний ресурс]. – режим доступу: URL: <https://www.sciencedirect.com/science/article/abs/> (Дата звернення – 30.11.2024).
12. Regression Algorithms [Електронний ресурс]. – режим доступу: URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/regression-algorithms> (Дата звернення – 03.12.2024).
13. Статистичні моделі [Електронний ресурс]. – режим доступу: URL: https://stud.com.ua/80643/geografiya/statistichni_modeli (Дата звернення – 08.12.2024).
14. Five machine learning types to know [Електронний ресурс]. – режим доступу: URL: <https://www.ibm.com/think/topics/machine-learning-types> (Дата звернення – 13.12.2024).
15. Machine Learning Applications based on SVM Classification A Review [Електронний ресурс]. – режим доступу: URL: <https://journal.qubahan.com/index.php/> (Дата звернення – 18.12.2024).
16. Machine learning and deep learning [Електронний ресурс]. – режим доступу: URL: <https://link.springer.com/article/> (Дата звернення – 05.01.2025).
17. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks [Електронний ресурс]. – режим доступу: URL: <https://www.mdpi.com/1999-5903/15/8/255> (Дата звернення – 12.01.2025).
18. Cloud Based Services [Електронний ресурс]. – режим доступу: URL: <https://www.geeksforgeeks.org/cloud-based-services/> (Дата звернення – 18.01.2025).
19. Moving average (Ковзна Середня): як не погоріти на ілюзії тренду [Електронний ресурс]. – режим доступу: URL: <https://bikotrading.pro/moving-average> (Дата звернення – 23.01.2025).
20. pandas.DataFrame.bfill [Електронний ресурс]. – режим доступу: URL:

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.bfill.html> (Дата звернення – 28.01.2025).

21. Methods for interpolating missing data in aerobiological databases [Електронний ресурс]. – режим доступу: URL: <https://www.sciencedirect.com/science/article/abs/> (Дата звернення – 04.02.2025).

22. Trend-seasonal components identification at the stage of time series pre forecasting analysis [Електронний ресурс]. – режим доступу: URL: <https://iopscience.iop.org/article/10.1088/1755-1315/786/1/012012/meta> (Дата звернення – 10.02.2025).

23. Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (KNN) algorithm to test the accuracy of types of breast cancer [Електронний ресурс]. – режим доступу: URL: <https://ijjis.org/index.php/IJIS/article/view/73> (Дата звернення – 17.02.2025).

24. Mean Squared Error, Deconstructed [Електронний ресурс]. – режим доступу: URL: <https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/21MS> (Дата звернення – 25.02.2025).

25. A modified Adam algorithm for deep neural network optimization [Електронний ресурс]. – режим доступу: URL: <https://link.springer.com/article/10.1007/s00521> (Дата звернення – 03.03.2025).

26. What are SSOT, DRY and SRP [Електронний ресурс]. – режим доступу: URL: <https://davidvlijmincx.com/posts/spot-dry-and-srp/> (Дата звернення – 14.03.2025).

27. Why Naive Models Are Still Relevant in the age of complex AI [Електронний ресурс]. – режим доступу: URL: <https://www.algorithmia.se/our-latest-thinking/why-naive-models-are-still-relevant-in-the-age-of-complex-ai> (Дата звернення – 25.03.2025).

28. Essential regression evaluation metrics (MSE, RMSE, MAE, R²) [Електронний ресурс]. – режим доступу: URL: <https://farshadabdulazeez.medium.com/essential-regression-evaluation-metrics-mse-rmse-mae-r%C2%B2-and-adjusted-r%C2> (Дата звернення – 10.04.2025).

ДОДАТОК А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ Харківський
національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного
інтелекту Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 15 – Автоматизація та приладобудування
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки

к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

ПОЛЮСКЕВИЧА Данила Станіславовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи **МОДЕЛЬ ПРОГНОЗУВАННЯ ВПЛИВУ ПОГОДНИХ
ФАКТОРІВ НА ВРОЖАЙНІСТЬ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР**

керівник роботи **Чуб Ольга Ігорівна, кандидат економічних наук, доцент**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від № 4101-5/962 від 16 квітня 2025 року

2. Строк подання студентом роботи **30 травня 2025 року**

3. Перелік питань, які потрібно розробити:

- 1) Огляд сучасних підходів до прогнозування врожайності сільськогосподарських культур із використанням математичних моделей, статистичних методів і машинного навчання.
- 2) Розробка концептуальної моделі нейронної мережі для прогнозування впливу погодних факторів на врожайність.
- 3) Тестування розробленої моделі на реальних або симульованих даних для оцінки її точності та ефективності.
- 4) Аналіз можливих обмежень та проблем прогнозування, таких як нестача даних, вплив непередбачуваних факторів або адаптація моделі до різних регіонів.
- 5) Підготовка рекомендацій щодо вдосконалення моделі та її практичного застосування в агропромисловому секторі.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Формулювання наукової проблеми, обґрунтування актуальності, мети та завдань дослідження. Визначення об'єкта, предмета та методів дослідження	01.10.2024 – 15.10.2024
2	Аналіз наукових досліджень та наявних методів прогнозування врожайності, включно з статистичними моделями, машинним навчанням та нейронними мережами	15.10.2024 – 31.10.2024
3	Огляд архітектур нейронних мереж та їх застосування для прогнозування часових рядів у сільському господарстві	01.11.2024 – 15.11.2024
4	Огляд та порівняльний аналіз наявних програмних рішень для прогнозування врожайності та аналізу погодних умов	15.11.2024 – 30.11.2024
5	Виклад матеріалу першого розділу кваліфікаційної роботи	01.12.2024 – 15.12.2024
6	Формування вимог до моделі, вибір архітектури нейронної мережі та необхідних вхідних параметрів	15.12.2024 – 31.12.2024
7	Розробка архітектури моделі прогнозування, вибір нейронної мережі (LSTM) та її гіперпараметрів	01.01.2025 – 15.01.2025
8	Реалізація програмного рішення для обробки та прогнозування врожайності на основі погодних даних	15.01.2025 – 31.01.2025
9	Тестування моделі та аналіз точності прогнозів	01.02.2025 – 15.02.2025
10	Виклад матеріалу другого розділу кваліфікаційної роботи	15.02.2025 – 01.03.2025
11	Підготовка практичних рекомендацій щодо використання моделі в агропромисловому секторі	01.03.2025 – 15.03.2025
12	Оформлення звіту про переддипломну практику	16.03.2025 – 31.03.2025
13	Остаточне оформлення пояснювальної записки кваліфікаційної роботи та проходження перевірки на антиплагіат	01.04.2025 – 15.04.2025
14	Передзахист кваліфікаційної роботи	16.04.2025 – 30.04.2025
15	Отримання відгуку керівника та рецензії	01.05.2025 – 31.05.2025

5. Дата видачі завдання 02 жовтня 2024 року.

Студент Полюсевич Д.С. Керівник роботи Чуб О.І. 

**Технічне завдання
на розробку програмного виробу
«Модель прогнозування впливу погодних факторів на врожайність
сільськогосподарських культур»**

Назва розділу	Назва та зміст підрозділу
1. Вступ	1.1. Назва програмного виробу: Модель прогнозування впливу погодних факторів на врожайність сільськогосподарських культур; 1.2. Сільське господарство, агроаналітика, погодозалежне планування, навчальні та дослідницькі цілі
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 151 – Автоматизація, комп'ютерно-інтегровані технології; 2.2. Завдання на кваліфікаційну роботу бакалавра.
3. Призначення розробки	3.1. Створення програмного засобу для автоматизованого прогнозування зміни погодних умов і прогнозування з метою розробки рекомендації до посіву польових культур на основі часових рядів. 3.2. Функціональні можливості: Отримання метеорологічних даних (температура, вологість, опади, вітер). Декомпозиція, нормалізація та побудова LSTM-прогнозу. Формування короткострокового прогнозу погоди на 14 днів та генерація рекомендацій щодо оптимальних культур для посіву за погодними умовами. 3.3. Вхідні дані: Історичні погодні дані з відкритих джерел та координати місць. 3.4. Вихідні дані: графіки прогнозованих значень та список культур з рекомендованим вирощуванням.
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: Програма повинна забезпечувати завантаження метеоданих, їх обробку, побудову прогнозу, генерацію рекомендацій та відображення графіків через графічний інтерфейс користувача; 4.2. Вимоги до надійності: Програма повинна працювати стабільно з

	датасетами обсягом не менше двох років; 4.3. Вимоги до умов експлуатації: Робота на персональному комп'ютері з ОС Windows 10/11; 4.4. Вимоги до складу параметрів технічних засобів: ПК з процесором не нижче Intel i3, не менше 4 ГБ ОЗП. Встановлене Python-середовище (версія 3.10+) з бібліотеками: torch, pandas, matplotlib, tkinter, meteostat, statsmodels; 4.5. Вимоги до інформаційної та програмної сумісності: Програма має бути сумісна з Windows та працювати без потреби в інтернеті після первинного завантаження метеоданих; 4.6. Вимоги до маркіровки та упаковки: немає; 4.7. Вимоги до транспортування та зберігання: немає; 4.8. Спеціальні вимоги: немає.
5. Вимоги до програмної документації	Програмною документацією до виробу «Модель прогнозування впливу погодних факторів на врожайність сільськогосподарських культур.» вважати: 1) Технічне завдання (Додаток Б до пояснювальної записки кваліфікаційної роботи); 2) Програму та методику випробувань розробленого програмного виробу (Додаток В до пояснювальної записки кваліфікаційної роботи) 3) Опис програмного виробу (Розділ 3 пояснювальної записки кваліфікаційної роботи); 4) Текст програми (Додаток Г пояснювальної записки до кваліфікаційної роботи)
6. Техніко-економічні показники	Оцінка економічної ефективності не проводиться, оскільки програмний продукт розроблено для навчально-дослідницьких цілей.
7. Стадії та етапи розробки	1) Огляд та порівняльний аналіз наявних програмних рішень для прогнозування врожайності та аналізу погодних умов 08.02.2025 – 12.02.2025 2) Формування вимог до моделі, вибір архітектури нейронної мережі та необхідних вхідних параметрів 18.02.2025 – 26.02.2025 3) Розробка архітектури моделі прогнозування, вибір нейронної мережі

	(LSTM) та її гіперпараметрів 26.02.2025 – 10.03.2025 4) Реалізація програмного рішення для обробки та прогнозування врожайності на основі погодних даних 10.03.2025 – 21.03.2025 5) Тестування моделі та аналіз точності прогнозів 21.03.2025 – 30.03.2025 6) Підготовка практичних рекомендацій щодо використання моделі в агропромисловому секторі 08.04.2025 – 15.04.2025
8.Порядок контролю та приймання	1) Поточний контроль виконання – раз на два тижні 2) Подача всієї документації в електронному та друкованому вигляді 3) Остаточне приймання програмного виробу здійснюється Атестаційною комісією під час захисту кваліфікаційної роботи.

Виконавець:

студент групи КУ-41

ПОЛЮСКЕВИЧ Д.С.


Замовник:

к.е.н., доцент кафедри комп'ютерних систем та робототехніки

ЧУБ О.І.



**Програма та методика випробувань програмного виробу
«Модель прогнозування впливу погодних факторів на врожайність
сільськогосподарських культур»**

1. Об'єкт випробувань

Об'єктом випробувань є реалізація методики побудови прогнозної моделі для погодних параметрів та визначення на її основі доцільності посіву сільськогосподарських культур.

2. Мета випробувань

Перевірка відповідності функціональності розробленого програмного виробу заявленим вимогам, визначеним у технічному завданні (Додаток Б до пояснювальної записки).

3. Загальні положення

3.1. Підстави щодо випробувань

Підставою для проведення випробувань є затвердження завдання на бакалаврську кваліфікаційну роботу.

3.2. Місце та тривалість випробувань

Приймальні випробування проводяться на базі персонального комп'ютера студента вдома в період роботи атестаційної комісії.

3.3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі, визначеному цією Програмою та методикою випробувань.

3.4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання за участю Замовника, Виконавця та інших присутніх осіб, призначених для прийому бакалаврської кваліфікаційної роботи.

4. Вимоги до програмного виробу

Програма повинна:

1. Завантажувати погодні дані з відкритого джерела (Meteostat API);
2. Виконувати обробку даних: агрегацію, нормалізацію, декомпозицію часового ряду;
3. Побудувати прогноз на основі LSTM-моделі щонайменше для 4 параметрів (температура, вологість, опади, вітер);
4. Виводити текстові рекомендації щодо культур, доцільних до посіву;
5. Мати інтерфейс користувача з вибором міста та виведенням результатів.

5. Вимоги до програмної документації

Склад програмної документації, що подається на випробування включає:

- 1) Справжнє технічне завдання (Додаток Б до пояснювальної записки);
- 2) Програму та методику випробувань (Додаток В);
- 3) Опис програмного виробу (розділ 2 пояснювальної записки);
- 4) Текст програми (Додаток Г).

6. Засоби та порядок випробувань

6.1 Засоби випробувань

Для виконання випробувань необхідно мати:

1. Персональний комп'ютер з ОС Windows 10/11;
2. Встановлене середовище Python (версія не нижче 3.10);
3. Встановлені бібліотеки torch, pandas, matplotlib, tkinter, statsmodels, meteostat;
4. Інтернет-з'єднання для отримання первинних погодніх даних.

6.2 Порядок проведення випробувань

Зазвичай випробування проводяться у два етапи:

- 1-й етап ознайомчий

- 2-й етап функціональні випробування

До першого етапу входить:

- Перевірка комплектності програмної документації відповідно до вимог технічного завдання
- Перевірка коректності встановлення всіх необхідних програмних засобів (Python 3.10+, бібліотеки torch, pandas, matplotlib, tkinter, statsmodels, meteostat).
- Перевірка доступу до Інтернету та працездатності підключення до meteostat API.

Другий етап передбачає проведення трьох тестів, спрямованих на перевірку виконання функціональних вимог до програмного виробу.

Тест 1: Побудова погодного прогнозу

Мета: перевірити наявність прогнозу для чотирьох параметрів: температура, вологість, опади, вітер.

Послідовність дій:

- Запустити програму
- Обрати будь-яке доступне місто в інтерфейсі
- Натиснути на кнопку «Make a forecast» та дочекатися результату
- Перевірити, що відображається 4 графіки прогнозу

Результат тесту № 1 наведений на рис. В.1.

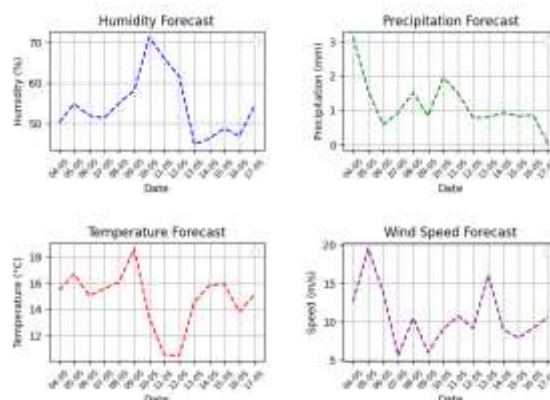


Рисунок В.1 – Вікно графіків прогнозу

Як ми бачимо, розроблена програма пройшла перше тестування.

Тест 2: Перевірка правдоподібності прогнозу

Мета: Перевірити, чи значення прогнозу знаходиться у фізично реальних межах.

Послідовність дій:

- Провести прогноз для довільного міста
- Оцінити графіки: відсутність аномальних або нелогічних значень
- Зіставити результати з відкритими джерелами онлайн-прогнозів

Використовуючи отриманий прогноз з попереднього тесту (рис. В.1), ми бачимо, що аномальні значення відсутні. Порівняємо прогноз на 05.05.2025 з онлайн ресурсом meteofor.com.ua (рис. В.2):



Рисунок В.2 – Онлайн-прогноз погоди на 05.05.2025

Як ми бачимо, середня температура відповідає онлайн-прогнозу. Також наша програма передбачила пориви вітру та кількість опадів (наші 1.9 мм

проти 2.1 мм, отриманий з онлайн-ресурсів). Виходячи з цього, ми можемо стверджувати, що програма пройшла другий тест.

Тест 3: Генерація рекомендацій щодо культур

Мета: перевірити наявність текстових рекомендацій після виконання прогнозу.

Послідовність дій:

- Запустити програму, обрати місто
- Створити прогноз
- Натиснути на кнопку «Recommendation» та перевірити наявність рекомендацій.

Виконавши приведені дії, ми отримаємо результат (рис. В.3):

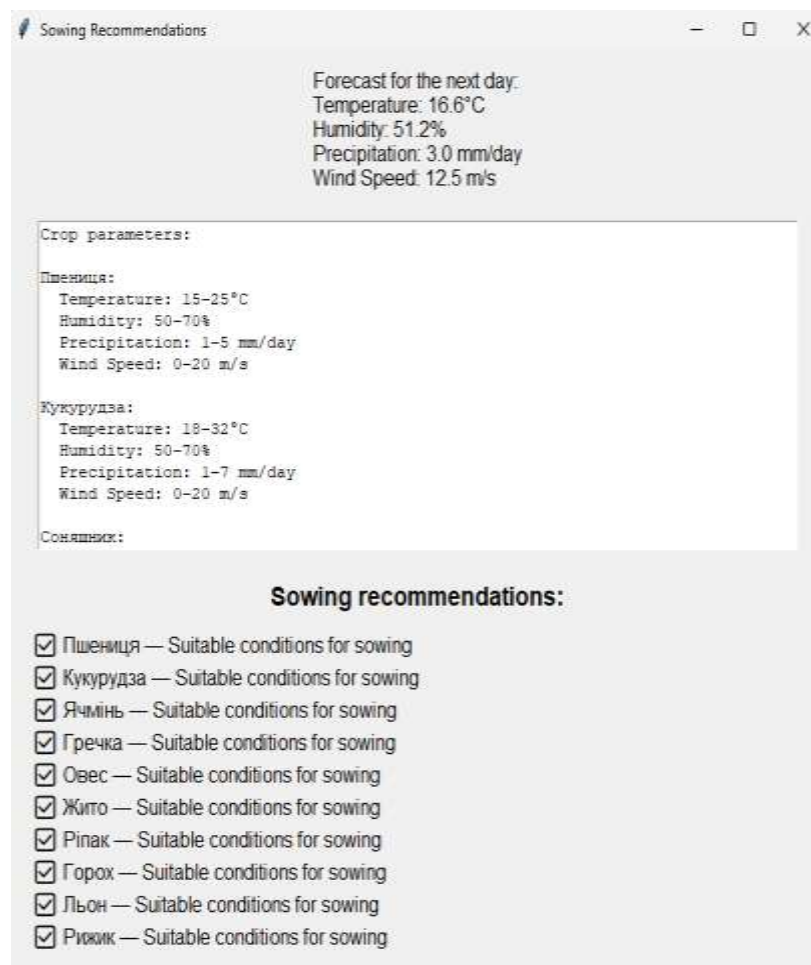


Рисунок В.3 – Лист рекомендацій до отриманого прогнозу

Як ми бачимо, програма надає лист культур, рекомендованих для посіву.

За результатами проведених тестів, ми впевнилися, що програма пройшла випробування успішно

Виконавець:

студент групи КУ-41

Полюскевич Д.С.



ЛИСТИНГ ПРОГРАМИ

Програмна реалізація моделі зберігається у відкритому репозиторії за посиланням:

<https://github.com/DanielPoliuskevych/-.git>

Репозиторій містить два основні файли:

- 1) AgroForecast_LSTM_System.py – основний код програми
- 2) README.md – технічний опис програми, вимоги, інструкції до запуску.