

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

**на тему: «Прогнозування сезонних обсягів надходження сировини
для виробництва молочних продуктів із використанням
нейронних мереж»**

Виконав: студент 4 курсу групи Кн–41

Спеціальності 122 «Комп'ютерні науки»
(шифр і назва)

Топорович Василь Ігорович
(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Бабич М.І.
(Прізвище та ініціали)

ДУБЛЯНИ–2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З.ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А. М. Тригуба

«_____» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Топоровичу Василю Ігоровичу

1. Тема роботи: «Прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів із використанням нейронних мереж»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 25.02.2025 року № 123/к–с.

2. Строк подання студентом роботи 10.06.2025 р.

3. Вихідні дані до роботи: статистичні дані обсягів надходження сировини для виробництва молочних продуктів; вимоги до сировини для виробництва молочних продуктів.

4. Зміст розрахунково–пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

1. Аналіз стану питання та прогнозування часових рядів.

2. Вибір інструментарію для прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.

3. Підготовка даних для навчання моделей прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.

4. Створення програмних модулів для прогнозування сезонних обсягів надходження сировини із використанням нейронних мереж.

5. Охорона праці.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень): аналіз стану питання та прогнозування часових рядів; вибір інструментарію для прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів; підготовка даних для навчання моделей прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів; створення програмних модулів для прогнозування сезонних обсягів надходження сировини із використанням нейронних мереж.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	Тригуба А.М., зав. кафедри ІТ		
5	Тимочко В.О., доцент кафедри інженерної механіки		

7. Дата видачі завдання

25 лютого 2025 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	При-мітка
1	Написання першого розділу	25.02–11.03.25	
2	Виконання другого розділу та аркушів ілюстраційного матеріалу до нього	12.03–15.04.25	
3.	Виконання третього розділу та аркушів ілюстраційного матеріалу до нього	16.04–20.05.25	
4.	Написання розділу «Охорона праці»	21.05–21.05.25	
5.	Завершення оформлення розрахунково–пояснювальної записки та аркушів ілюстраційного матеріалу	22–31.05.25	
6.	Завершення роботи в цілому	01–10.06.25	

Студент _____ Топорович В.І.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 004.032.26:637.1

Прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів із використанням нейронних мереж.

Топорович В.І. Кафедра ІТ – Дубляни, Львівський НУВМ, 2025.

Кваліфікаційна робота: 76 с. текст. част., 21 рис., 8 табл., 14 арк. ілюстраційного матеріалу, 41 джерело.

У роботі розроблено модель прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів, яка є актуальною з огляду на високий ступінь сезонності в аграрному секторі та необхідність забезпечення стабільного виробничого процесу.

Проаналізовано сучасний стан молочної галузі України, особливості динаміки постачання сировини, а також існуючі підходи до прогнозування часових рядів. Детально розглянуто архітектури нейронних мереж, такі як багатошаровий перцептрон (MLP) та рекурентна нейронна мережа (RNN), які були реалізовані в середовищі Python з використанням бібліотеки TensorFlow. Виконано підготовку вхідних даних, що включала нормалізацію, створення сезонних ознак та розбиття вибірки на тренувальну, валідаційну та тестову.

Запропоновані моделі протестовано на реальних погодних і календарних даних. Проведено порівняльний аналіз точності передбачень, що продемонстрував переваги використання RNN для задач із чітко вираженими сезонними коливаннями. Отримані результати свідчать про можливість ефективного застосування штучних нейронних мереж у системах підтримки прийняття рішень на підприємствах молочної галузі.

Розроблено заходи безпеки під час розробки моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.

Ключові слова: прогнозування, сезонність, молоко, нейронні мережі.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	9
1.1. Сучасний стан молочної галузі та проблема сезонності постачання сировини	9
1.2. Аналіз стану використання нейронних мереж для прогнозування часових рядів	11
1.3. Архітектура та навчання нейронних мереж для задач прогнозування....	14
1.4. Обґрунтування доцільності прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів із використанням нейронних мереж	18
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ДЛЯ ВИРОБНИЦТВА МОЛОЧНИХ ПРОДУКТІВ.....	20
2.1. Мета та завдання роботи	20
2.2. Вибір технологій та засобів для розробки моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.....	21
2.3. Характеристика вибраних архітектур штучних нейронних мереж	24
2.3.1. Багатошаровий перцептрон (MLP).....	24
2.3.2. Рекурентна нейронна мережа (RNN)	26
РОЗДІЛ 3. ПІДГОТОВКА ДАНИХ ДЛЯ НАВЧАННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ДЛЯ ВИРОБНИЦТВА МОЛОЧНИХ ПРОДУКТІВ.....	28
3.1. Характеристики джерела та наявних даних	28
3.2. Попередня обробка даних	32
3.3. Нормалізація та масштабування змінних	35
3.4. Побудова сезонних ознак	37
3.5. Формування вхідних і вихідних ознак для моделей.....	40

3.6. Візуалізація та статистичний аналіз даних.....	43
3.7. Розбиття вибірки на тренувальну, валідаційну та тестову	46
РОЗДІЛ 4. СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ	
ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ІЗ	
ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ	
4.1. Створення архітектури моделі MLP (багатошаровий перцептрон).....	49
4.2. Створення архітектури моделі RNN (рекурентна нейронна мережа)	51
4.3. Порівняльний аналіз моделей MLP та RNN.....	54
РОЗДІЛ 5. ОХОРОНА ПРАЦІ.....	
5.1. Аналіз стану робочого місця та умов праці.....	57
5.2. небезпечні та шкідливі чинники	59
5.3. Інструкція із охорони праці під час розробки моделі прогнозування сезонних обсягів надходження сировини на основі нейронних мереж.....	61
ВИСНОВКИ І ПРОПОЗИЦІЇ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТКИ	73

ВСТУП

Планування обсягів виробництва у молочній промисловості неможливе без урахування сезонних коливань у надходженні сировини. Для багатьох підприємств ці зміни є критичними, адже від стабільного постачання молока залежить і сам виробничий процес, і своєчасність виконання контрактних зобов'язань. Часто надходження молочної сировини різко зменшується в окремі місяці року, що ускладнює завчасне планування закупівель, навантаження обладнання, зберігання та логістики. Наявні методи оцінювання обсягів постачання, що базуються лише на середньостатистичних даних, не дозволяють враховувати всі фактори, що впливають на реальну картину [16].

Сучасні інформаційні технології, зокрема нейронні мережі, поступово стають важливим інструментом у вирішенні практичних завдань прогнозування [37]. Їхнє застосування дає змогу працювати з великими обсягами даних, виявляти закономірності, які не видно при поверхневому аналізі, та будувати моделі, здатні адаптуватися до змін середовища. Саме тому у цій роботі зроблено спробу поєднати можливості нейромережових підходів із практикою планування у молочній галузі. Йдеться не лише про побудову теоретичної моделі, а й про створення прикладного інструменту, який можна було б використати на виробництві.

Ідея роботи полягає у тому, щоб на основі наявних історичних даних змодельовати характер сезонних змін у постачанні молока, а далі використати нейронну мережу для прогнозу майбутніх обсягів. Результат цього підходу має практичну цінність: він дає змогу заздалегідь готуватися до пікових або навпаки – «низьких» періодів, оптимізувати витрати й уникати надмірного або недостатнього завантаження виробничих ліній.

З огляду на актуальність теми та потребу в автоматизованих рішеннях, було поставлено мету роботи – розробити програмні модулі, які, базуючись на нейронних мережах, забезпечить точне прогнозування сезонного постачання сировини для молочної продукції. У роботі розглянуто процес підготовки

даних, навчання моделі, перевірки її точності та подано реалізацію у вигляді зручного програмного інтерфейсу для подальшого застосування в реальних умовах.

РОЗДІЛ 1.

АНАЛІЗ СТАНУ ПИТАННЯ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1. Сучасний стан молочної галузі та проблема сезонності постачання сировини

Молочна галузь в Україні залишається однією з ключових у структурі агропромислового комплексу, забезпечуючи не лише внутрішні потреби населення, а й формуючи значну частку аграрного експорту. Виробництво молока та молочних продуктів є важливим джерелом зайнятості, особливо у сільській місцевості, де значна частина господарств займається первинним виробництвом сировини. Разом із тим, розвиток цієї галузі супроводжується низкою проблем, серед яких однією з найвагоміших залишається нерівномірність надходження молочної сировини впродовж року [36].

Сезонність у постачанні молока обумовлена як біологічними особливостями тварин, так і змінами у кормовій базі, умовах утримання, впливом температурного режиму та тривалості світлового дня. У більшості господарств, особливо малих, спостерігається підвищення надоїв у весняно–літній період, коли корми є більш поживними та доступними, а утримання тварин – менш енерговитратним. Водночас у зимовий період продуктивність тварин знижується, що безпосередньо позначається на обсягах заготівлі молока. Така сезонна нестабільність постачання створює суттєві ризики для переробних підприємств, які змушені балансувати між надлишком сировини в окремі місяці та її дефіцитом в інші.

Крім природних чинників, на сезонність впливають і економічні умови: нестабільність закупівельних цін, коливання витрат на енергоносії, доступність технічних засобів для зберігання та охолодження молока, наявність збутових каналів. У багатьох випадках фермери свідомо зменшують обсяги виробництва в зимовий період через високі витрати на утримання тварин або через труднощі з транспортуванням сировини до пунктів приймання.

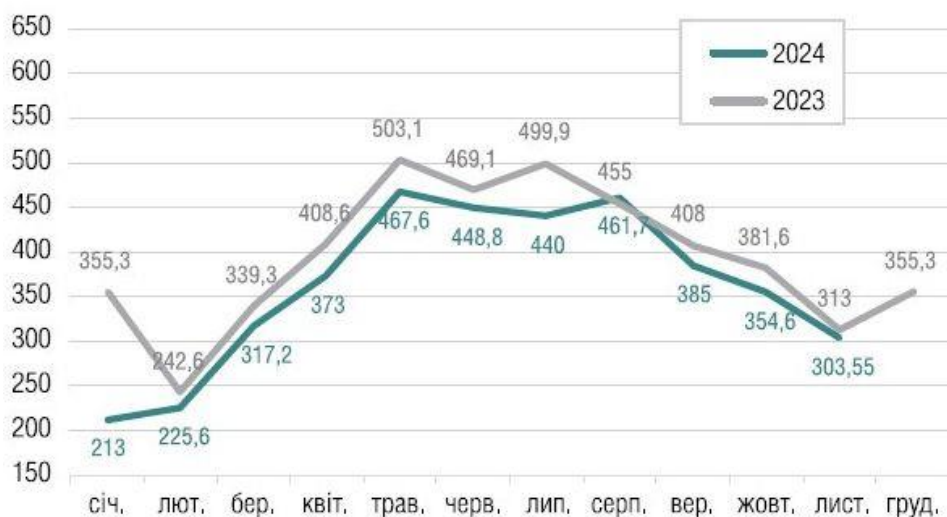


Рисунок 1.1 – Обсяги та сезонність виробництва молока в особистих селянських господарствах, тис. т [19]

Для молокопереробних підприємств, особливо тих, що працюють на постійне навантаження обладнання та мають контракти з роздрібними мережами, прогнозованість постачання стає критично важливою. Невизначеність у кількості сировини, яка буде доступна у конкретний період, ускладнює формування виробничих планів, збільшує витрати на зберігання або призводить до недозавантаження виробничих ліній. Усе це знижує загальну ефективність галузі та потребує запровадження більш точних інструментів планування.

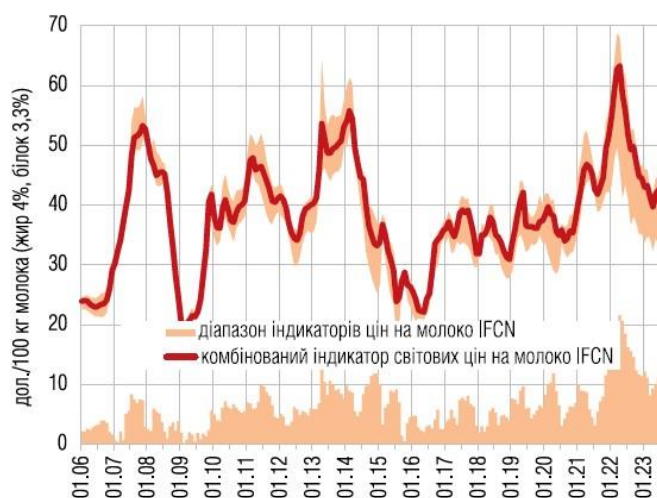


Рисунок 1.2 – Тенденції зміни комбінованого індикатора світових цін на молоко IFCN* [14]

Серед країн, які мають потенціал для збільшення обсягів виробництва молока, аналітики передусім виділяють держави Північної Америки, де прогнозується ріст у межах 1–2%. Подібні темпи зростання очікуються також у деяких країнах Південної Америки, зокрема в Бразилії та Аргентині (1–2%), Перу (2–3%) та Парагваї, де прогнозують не менше 3% приросту. У частині азійського регіону також фіксуються позитивні очікування: у Китаї – 0,5–1%, Казахстані – 2–3%, а в Індії – щонайменше 3% [14].

На тлі коливань світової пропозиції важливою залишається динаміка цін на молоко. Після значного зростання цін у 2022 році (рис. 1.2) багато експертів очікували, що це стимулює різке збільшення обсягів виробництва. Навпаки, в країнах Океанії через несприятливі кліматичні умови спостерігалось скорочення виробництва. Аналогічні природні фактори й надалі залишатимуться серйозним обмеженням для розширення молочної галузі в країнах Південної Європи, Латинської Америки та регіону Океанії.

У таких умовах особливої актуальності набуває впровадження методів прогнозування, які б дозволяли із випередженням оцінювати майбутні обсяги надходження молочної сировини. З огляду на складність сезонних коливань та їх багатофакторну природу, класичні статистичні методи часто виявляються недостатніми. Це обґрунтовує необхідність звернення до інструментів машинного навчання, зокрема до нейронних мереж, здатних виявляти приховані закономірності та адаптуватися до змінних умов середовища.

1.2. Аналіз стану використання нейронних мереж для прогнозування часових рядів

Штучні нейронні мережі (ШНМ) набули широкого поширення як один з найефективніших інструментів моделювання складних систем, особливо у випадках, коли традиційні статистичні методи виявляються недостатньо гнучкими для опису динаміки процесів. У задачах прогнозування часових рядів

ШНМ приваблюють дослідників своєю здатністю відображати нелінійні, багатофакторні залежності між історичними спостереженнями та майбутніми значеннями, навіть коли явні математичні моделі цих залежностей відсутні.

На відміну від класичних моделей, які зазвичай вимагають специфічних припущень щодо структури часового ряду, ШНМ є методами, керованими даними, і не вимагають апріорного знання функціональних форм. Вони навчаються на основі прикладів, виявляючи приховані закономірності та шаблони в даних, включно з шумами, які часто маскують справжню поведінку системи. Це робить їх особливо корисними в ситуаціях, коли неможливо побудувати адекватну модель на основі теоретичних припущень.

Процес прогнозування в рамках ШНМ базується на перетворенні вхідного вектора значень попередніх моментів часу $y_{t-1}, y_{t-2}, \dots, y_{t-n}$ прогнозне значення y_t . Такий підхід математично можна подати як функцію:

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-n}; \theta), \quad (1.1)$$

де f – функція, що апроксимується нейронною мережею; θ – сукупність вагових коефіцієнтів мережі, які визначаються в процесі навчання.

ШНМ мають здатність узагальнювати – після навчання на скінченній кількості прикладів вони можуть робити висновки щодо нових, раніше не бачених ситуацій. Саме ця властивість є критично важливою для прогнозування, де необхідно передбачити поведінку системи на основі історичних даних. Як показано на рисунку 1, типова мережа для таких задач складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару.

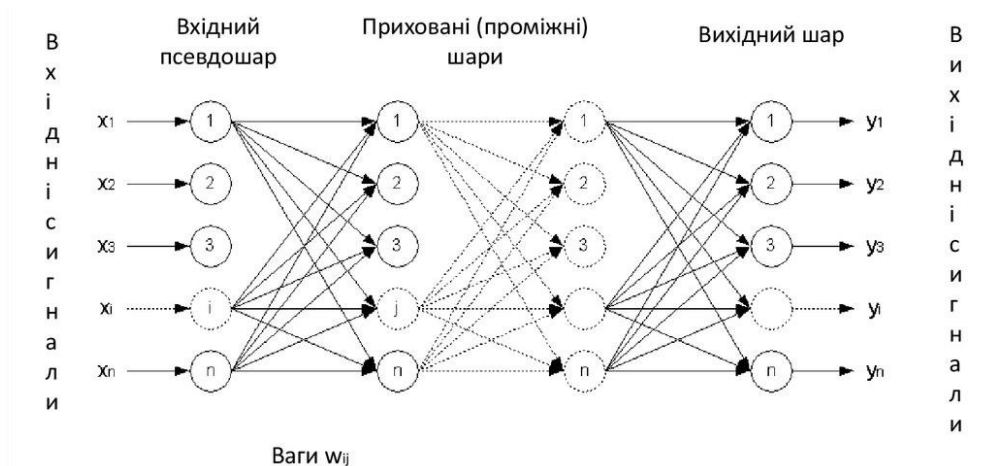


Рисунок 1.3 – Багатошарова нейронна мережа прямого зв’язку для прогнозування

Завдяки своїм універсальним апроксимативним властивостям, ШНМ здатні відтворювати довільну безперервну функцію з будь-якою точністю [22]. Це означає, що вони можуть теоретично моделювати будь-яку залежність між вхідними та вихідними змінними, якщо мережа має достатню кількість нейронів і навчені відповідні ваги.

Крім того, ШНМ є нелінійними моделями. У той час як більшість традиційних підходів до аналізу часових рядів, зокрема методи ARIMA або Box-Jenkins, ґрунтуються на припущенні лінійності, нейромережі не мають такого обмеження. У реальних системах, де взаємодія між параметрами носить нелінійний характер, ШНМ здатні точніше моделювати ці взаємозв’язки. Це особливо важливо для прогнозування у сфері аграрного виробництва, де сезонність, кліматичні умови та біологічні чинники створюють складну, важко формалізовану динаміку.

У задачах, що передбачають прогнозування часових рядів, мережа фактично реалізує нелінійну авторегресійну модель, тобто:

$$y_t = F(y_{t-1}, y_{t-2}, \dots, y_{t-n}), \quad (1.2)$$

де F – нелінійне перетворення, яке виконує ШНМ.

Для навчання такої моделі формується набір тренувальних шаблонів з фіксованою довжиною вікна спостережень. Наприклад, при прогнозуванні

одного кроку вперед використовується послідовність $(y_1, y_2, \dots, y_n) \rightarrow y_{n+1}$, $(y_2, y_3, \dots, y_{n+1}) \rightarrow y_{n+2}$ і так далі. Результатом є серія входів і відповідних цільових значень для навчання.

Навчання мережі здійснюється шляхом мінімізації функції втрат, найчастіше у вигляді середньоквадратичної помилки:

$$E = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (1.3)$$

де y_i – фактичне значення; $\hat{y}_i$ – прогнозоване мережею; N – кількість навчальних прикладів. Мінімізація цієї функції здійснюється, як правило, за допомогою алгоритму зворотного поширення помилки (backpropagation), що забезпечує оновлення вагових коефіцієнтів у напрямку зменшення помилки.

Таким чином, використання штучних нейронних мереж для прогнозування часових рядів дає змогу моделювати складні, нечітко виражені закономірності у даних, які важко описати аналітично. Саме завдяки своїм властивостям – гнучкості, здатності до узагальнення, відсутності жорстких припущень та ефективності в умовах шумових даних – ШНМ стали потужним інструментом у сучасній практиці прогнозування.

1.3. Архітектура та навчання нейронних мереж для задач прогнозування

Архітектура штучної нейронної мережі (ШНМ) визначає спосіб, у який нейрони організовані в шари, та характер зв'язків між ними. Найпоширенішим типом мережі, що використовується у задачах прогнозування, є багат шаровий перцептрон (MLP), який складається з вхідного шару, одного або декількох прихованих шарів та вихідного шару. Кожен шар містить певну кількість нейронів, які отримують інформацію від попереднього шару, обробляють її за допомогою функцій активації та передають результати далі. Усі з'єднання між

нейронами мають вагові коефіцієнти, які оновлюються під час навчання, від чого залежить здатність мережі до відтворення залежностей у даних.

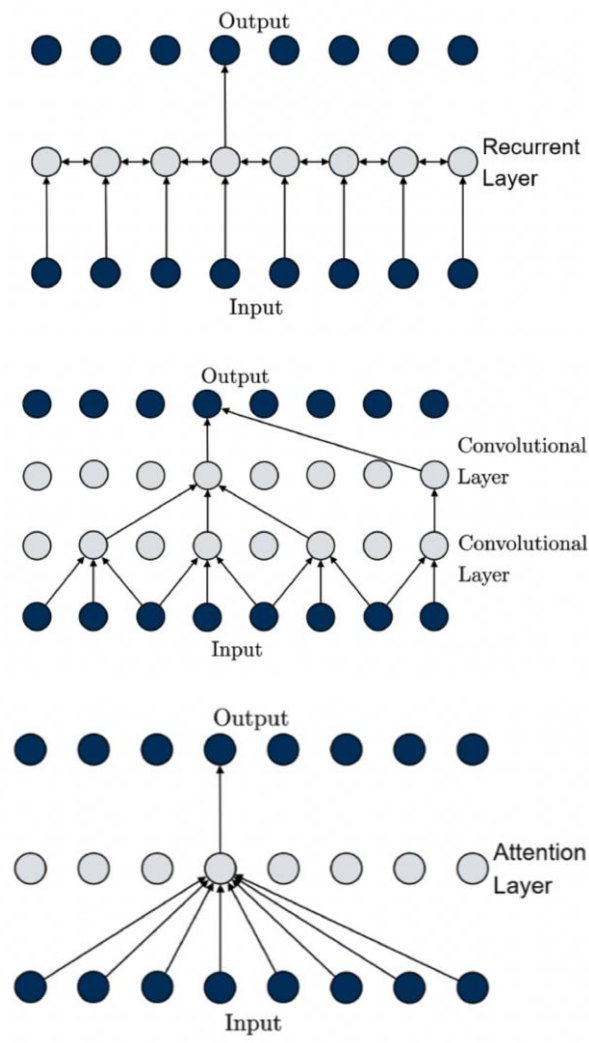


Рисунок 1.4 – Концепційне порівняння різних парадигм глибокого навчання для моделювання часових залежностей між вхідними зразками: LSTM (ліворуч), CNN (посередині) та уваги (праворуч)

Розмірність вхідного шару визначається кількістю ознак, що подаються на вхід моделі. У контексті прогнозування часових рядів, це зазвичай фіксована кількість попередніх значень ряду. Наприклад, якщо прогноз виконується з лагом n , то вхідний вектор матиме вигляд:

$$x_t = [y_{t-1}, y_{t-2}, \dots, y_{t-n}], \quad (1.4)$$

Відповідно, вихідний шар містить один нейрон, який генерує прогнозоване значення y_t , що є оцінкою майбутнього стану часового ряду. Зв'язок між вхідним та вихідним значенням можна подати як:

$$\hat{y}_t = f(x_t; \theta), \quad (1.5)$$

де f – функція, що реалізується ШНМ; θ – набір усіх вагових параметрів мережі.

У разі використання кількох прихованих шарів мережа має здатність представляти ще складніші нелінійні залежності, завдяки чому підвищується точність прогнозування.

Функція активації є важливим елементом у роботі мережі, оскільки вона дозволяє моделі враховувати нелінійність процесу. У прихованих шарах зазвичай використовують функцію ReLU (Rectified Linear Unit):

$$\phi(z) = \max(0, z), \quad (1.6)$$

або сигмоїдальну функцію:

$$\phi(z) = \frac{1}{1 + e^{-z}}, \quad (1.7)$$

де z – зважена сума вхідних сигналів для нейрона.

У вихідному шарі, залежно від типу задачі, застосовують або лінійну функцію (для регресійного прогнозу), або softmax (у класифікаційних задачах).

Навчання нейронної мережі – це процес підбору оптимальних значень ваг θ , що мінімізують розбіжність між реальними значеннями та прогнозами мережі. Найпоширенішим підходом до навчання є метод зворотного поширення помилки (backpropagation), який використовує градієнтний спуск для оновлення ваг на основі похідної функції втрат. У більшості задач прогнозування часових рядів використовується середньоквадратична помилка (Mean Squared Error, MSE).

Для задач часових рядів навчальний набір формується шляхом ковзного вікна: кожен вектор вхідних значень включає фіксовану кількість послідовних спостережень, а відповідний вихід – наступне значення. Наприклад, при

наявності послідовності $y_1, y_2, \dots, y_N$, і фіксованому лагу n , кількість навчальних прикладів дорівнює $N - n$, а перше навчальне представлення буде:

вхід

$$\text{вхід}: [y_1, y_2, \dots, y_n] \Rightarrow \text{вихід}: y_{n+1}, \quad (1.8)$$

Наступне:

$$[y_2, y_3, \dots, y_{n+1}] \Rightarrow y_{n+2}, \quad (1.9)$$

і так далі до завершення всього набору.

На рисунку 1.5 зображено загальну схему організації навчального процесу ШНМ у задачі прогнозування часових рядів, де послідовно показано подачу вхідних даних, проходження сигналу через мережу та оновлення ваг у процесі зворотного поширення помилки.

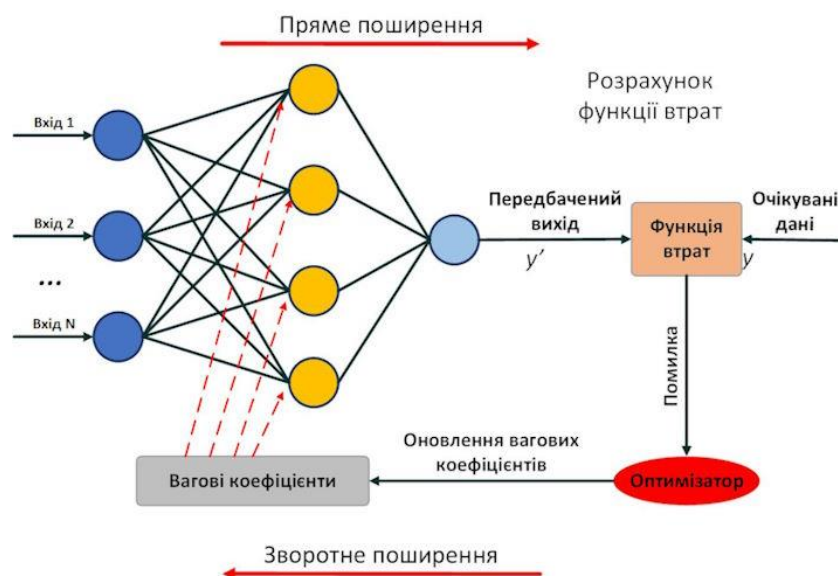


Рисунок 1.4 – Схема навчання нейронної мережі для прогнозування часових рядів

Ефективність навчання залежить не лише від структури мережі, але й від обсягу та якості даних, правильного налаштування гіперпараметрів, таких як швидкість навчання, кількість нейронів у прихованих шарах, розмір вікна лагів, а також від методів регуляризації, які дозволяють уникати перенавчання. Саме тому побудова ШНМ для прогнозу часових рядів є не лише технічним

завданням, а й комплексним дослідницьким процесом, який потребує експериментів і оптимізації на кожному етапі.

1.4. Обґрунтування доцільності прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів із використанням нейронних мереж

У молочній промисловості ефективність виробництва значною мірою залежить від стабільності та передбачуваності постачання сировини. Проте на практиці підприємства стикаються з регулярними сезонними коливаннями обсягів надходження молока, які зумовлені низкою факторів: змінами погодних умов, особливостями годівлі тварин, періодами лактації, а також поведінкою дрібних постачальників. Такі коливання часто мають нерегулярний і складний характер, що ускладнює формування точних виробничих планів. У результаті спостерігається або перевантаження виробничих ліній, або їхнє недовантаження, що призводить до зниження рентабельності, втрат сировини або недоотримання прибутку.

В умовах нестабільного сировинного ринку, де частка індивідуальних господарств і фермерських постачань залишається вагомою, класичні методи статистичного прогнозування, що ґрунтуються на припущеннях про лінійність або стабільність рядів, виявляються малоефективними. Наприклад, використання середньомісячних показників чи методів ковзного середнього часто не враховує прихованих змін або впливу зовнішніх факторів. У такій ситуації виникає потреба у більш гнучкому інструменті, здатному навчатися на історичних даних, виявляти складні зв'язки та адаптуватися до нових умов.

Штучні нейронні мережі, завдяки своїй здатності апроксимувати довільні нелінійні функції та працювати з великими обсягами неструктурованих або зашумлених даних, стають перспективним інструментом для задач сезонного прогнозування. Вони не вимагають попереднього визначення математичної

форми залежностей між змінними, що особливо цінно у випадках, коли природа таких залежностей є складною або недостатньо вивченою. На відміну від регресійних моделей чи ARIMA, нейронні мережі можуть одночасно враховувати як часові лаги, так і супутні чинники – наприклад, температурні коливання, кількість опадів, типи кормів або зміни у структурі постачальників.

Особливої актуальності набуває використання нейромереж у тих виробничих системах, де дані накопичуються у великих обсягах, але мають нерівномірну або фрагментарну структуру. Нейронні мережі добре справляються з такими даними, оскільки здатні виокремлювати важливі характеристики без потреби ручної інженерії ознак. Крім того, після навчання на історичних даних, модель може бути використана для автоматичного прогнозування майбутніх обсягів, що значно полегшує управлінське планування та дозволяє заздалегідь коригувати виробничі та логістичні стратегії.

Застосування нейромережевого підходу є також економічно доцільним. Його реалізація не потребує дорогого обладнання чи ліцензійного програмного забезпечення, а основні моделі можуть бути реалізовані на відкритих платформах машинного навчання. У поєднанні з аналітикою реального часу, нейронна мережа може стати основою для побудови адаптивної системи підтримки прийняття рішень, яка дозволить переробним підприємствам гнучко реагувати на зміну обсягів постачання.

Усе це свідчить про високу практичну цінність застосування штучних нейронних мереж для прогнозування сезонних обсягів надходження молочної сировини. Вони не лише підвищують точність прогнозу, а й сприяють сталому розвитку виробництва, зменшенню втрат і оптимізації витрат на всіх етапах переробного ланцюга. У контексті цифрової трансформації аграрного сектору та зростаючих вимог до ефективності харчових ланцюгів постачання, впровадження нейромережевих моделей набуває стратегічного значення.

РОЗДІЛ 2.

ВИБІР ІНСТРУМЕНТАРІЮ ДЛЯ ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ДЛЯ ВИРОБНИЦТВА МОЛОЧНИХ ПРОДУКТІВ

2.1. Мета та завдання роботи

У виробництві молочних продуктів однією з ключових передумов стабільної роботи підприємства є своєчасне та передбачуване надходження сировини. Сезонна мінливість обсягів заготівлі молока, зумовлена як природними чинниками, так і особливостями організації постачання, суттєво впливає на виробничі потужності, логістику та ринкову стабільність. У зв'язку з цим виникає нагальна потреба у застосуванні методів прогнозування, які дозволяли б підприємству гнучко реагувати на зміни у сировинному забезпеченні та мінімізувати втрати, пов'язані з браком або надлишком молока в різні періоди року.

Основна мета цієї роботи полягає у створенні моделі прогнозування сезонних обсягів надходження молочної сировини, що ґрунтується на сучасних методах обробки даних. Запропонована модель має забезпечити підвищення точності прогнозу завдяки використанню штучних нейронних мереж, які здатні розпізнавати складні нелінійні закономірності у часових рядах. Метою є не лише формальне побудування прогнозу, а й обґрунтування вибору відповідного інструментарію, здатного адаптуватися до змін середовища та оновлювати параметри моделі в міру накопичення нових даних.

Для досягнення поставленої мети передбачено реалізувати низку завдань. Насамперед, слід охарактеризувати типові ознаки сезонності у процесі надходження сировини, спираючись на статистичні дані та приклади з виробничої практики. Далі необхідно дослідити теоретичні засади побудови прогнозних моделей на основі нейронних мереж, зокрема багат шарових перцептронів, які найчастіше застосовуються для аналізу часових рядів. У

межах роботи має бути проведено аналіз вхідних даних, сформовано навчальні та тестові вибірки, підібрано оптимальну архітектуру моделі з урахуванням глибини мережі, кількості нейронів та функцій активації.

Крім того, передбачено реалізувати програмну частину моделі та провести експериментальну перевірку точності прогнозів, що генеруються нейромережею. За результатами моделювання планується здійснити порівняльну оцінку отриманих прогнозів із результатами, що були отримані за допомогою традиційних статистичних методів. Підсумковим завданням роботи є узагальнення результатів, виявлення переваг та обмежень застосованого підходу, а також формування рекомендацій щодо його впровадження на підприємствах, які займаються переробкою молока.

2.2. Вибір технологій та засобів для розробки моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів

Розробка ефективної моделі прогнозування сезонних обсягів надходження молочної сировини потребує ретельного підходу до вибору технологій, інструментів та середовища програмної реалізації. Урахування специфіки вхідних даних, характеру сезонних коливань, а також вимог до точності та масштабованості моделі обумовлює використання інструментарію, який забезпечує високу гнучкість та можливість роботи з часовими рядами. Вибір засобів здійснювався з урахуванням як функціональних можливостей платформ, так і поширених практик у галузі обробки даних та штучного інтелекту.

Основою для створення моделі слугувала мова програмування Python, яка є загальновизнаним стандартом у сфері аналітики даних, машинного навчання та наукових обчислень. Її популярність зумовлена широким спектром бібліотек, активною спільнотою користувачів, постійною підтримкою оновлень та

зручним синтаксисом. Зокрема, для реалізації нейронної мережі було обрано бібліотеку TensorFlow у поєднанні з високорівневим API Keras (рис. 2.1), що забезпечує просту побудову, тренування та оцінювання моделей глибокого навчання.



Рисунок 2.1 – Логотип вибраної бібліотеки TensorFlow у поєднанні з високорівневим API Keras

Такий вибір дозволяє зосередитися на логіці прогнозування, не витрачаючи надмірних зусиль на реалізацію низькорівневих обчислень.

Для попередньої обробки даних, очищення, нормалізації та формування часових вікон було використано бібліотеки NumPy та Pandas (рис. 2.2).



Рисунок 2.2 – Логотипи вибраних бібліотек NumPy та Pandas

Вони забезпечують ефективну роботу з табличними структурами, зручну агрегацію, фільтрацію та трансформацію даних, що є критично важливим на етапі підготовки вибірки до навчання. Також використовувалася бібліотека Matplotlib (рис. 2.3) для візуалізації часових рядів, аналізу сезонних коливань і представлення результатів роботи моделі у графічній формі.



Рисунок 2.3 – Логотип вибраної бібліотеки Matplotlib

Однією з важливих вимог до моделі було врахування часової залежності між спостереженнями. Зважаючи на це, було розглянуто кілька архітектур штучних нейронних мереж, зокрема багатошарові перцептрони (MLP) та рекурентні нейронні мережі (RNN). Попри переваги RNN у роботі з часовими структурами, у даному проєкті було зосереджено увагу на MLP як на більш доступному та стабільному рішенні для випадку з обмеженим обсягом історичних даних і чітко вираженою сезонністю. При правильному формуванні вхідних векторів багатошаровий перцептрон демонструє високу точність і стійкість при моделюванні сезонних залежностей.

З технічного погляду модель реалізовано у середовищі Jupyter Notebook, що дозволяє поєднувати код, графіки та текстові пояснення в єдиному інтерактивному документі. Такий підхід є зручним для досліджень, документування та демонстрації результатів. Крім того, Jupyter забезпечує гнучкість під час налаштування параметрів моделі, збереження результатів і повторного відтворення експериментів.

Таким чином, обраний набір інструментів – Python, TensorFlow/Keras, Pandas, NumPy, Matplotlib – дозволяє реалізувати повний цикл побудови прогнозової моделі: від обробки сирих вхідних даних до навчання нейромережі та оцінки її якості. Такий підхід забезпечує не лише наукову обґрунтованість, а й практичну придатність розробленого рішення для впровадження у виробничу діяльність підприємств молочної галузі.

2.3. Характеристика вибраних архітектур штучних нейронних мереж

2.3.1. Багатошаровий перцептрон (MLP)

MLP як метод ШНМ із прямою передачею даних може успішно генерувати значення вихідних змінних відповідно до вхідних змінних за допомогою нелінійної функції. Проста архітектура моделі MLP представлена рис. 2.4.

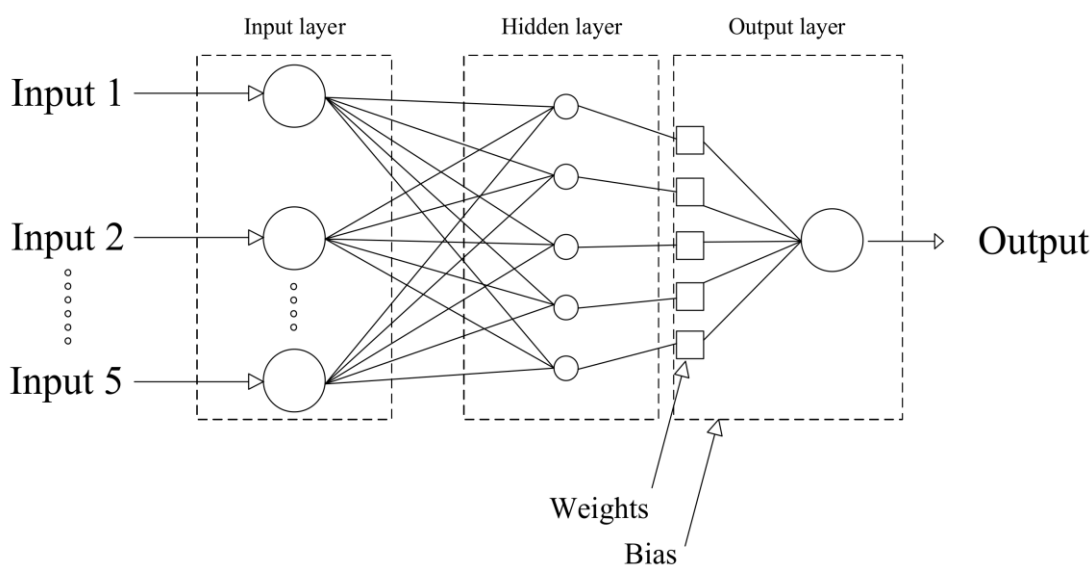


Рисунок 2.4 – Архітектура моделі багатошарового перцептрона MLP

Згідно з рис. 2.4, MLP містить три основні розділи. Перший розділ імпортує вхідні змінні, другий розділ називається прихованим шаром і включає набір нейронів, які називаються нейронами прихованого шару. Кількість нейронів у прихованому шарі є одним із регульованих факторів, які можуть впливати на точність моделі MLP. Останній рівень називається вихідним рівнем і містить вихідні змінні.

Ця модель часто згадується в різних дослідженнях. Нами розглянуто лише основні та важливі аспекти, що стосуються MLP. У MLP прихований рівень з'єднує вхідний рівень з вихідним і створює вихідне значення ($f(x)$) за допомогою рівняння (2.1):

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad f(x) = K(b^{(2)} + W^{(2)} \cdot Q(b^{(1)} + W^{(1)}x)), \quad (2.1)$$

де K та Q – відповідно функції активації на відповідних шарах; $b^{(1)}$, $b^{(2)}$ – відповідно вектори зміщень (bias) для прихованого та вихідного шару; $W^{(1)}$, $W^{(2)}$ – відповідно матриці ваг зв'язків між шарами.

Прихований шар визначається як:

$$h(x) = Q(b^{(1)} + W^{(1)}x), \quad (2.2)$$

Відповідно, для функції активації Q часто використовуються дві узагальнені нелінійності:

Гіперболічний тангенс:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2.3)$$

Сигмоїдна функція:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2.4)$$

Функція $\tanh(x)$ у багатьох випадках забезпечує швидшу та ефективнішу збіжність у порівнянні з $\sigma(x)$.

Вихідний вектор моделі обчислюється на основі активації прихованого шару за рівнянням:

$$o(x) = K(b^{(2)} + W^{(2)} \cdot h(x)), \quad (2.5)$$

У нас архітектура MLP включає один вхідний рівень, який складається з п'яти параметрів:

VDate – дата;

VMilk_Production – обсяг заготівлі молока;

CTemperature (°C) – температура повітря;

CPressurepre (mm) – кількість опадів.

Ці параметри використовуються як вхідні характеристики $x \in \mathbb{R}^5$ у навчанні MLP-моделі для задач прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.

2.3.2. Рекурентна нейронна мережа (RNN)

Рекурентна нейронна мережа (RNN) є різновидом штучної нейронної мережі, архітектура якої подібна до прямого (feed-forward) перцептрона, однак відрізняється тим, що передача інформації в мережі здійснюється не лише в прямому напрямку від входу до виходу, але й зворотно. Ця властивість надає мережі здатність опрацьовувати послідовності даних, зберігаючи контекст попередніх станів.

Особливістю RNN є використання зворотного зв'язку, який дозволяє повторно використовувати результати проміжних вузлів з попередніх моментів часу як вхідні дані на наступному кроці. Таким чином, модель має змогу зберігати історичну інформацію та враховувати залежності між значеннями у часовому ряді. Математично це можна описати такою системою рівнянь:

$$h_t = Q(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h), \quad (2.6)$$

$$y_t = K(W_{hy} \cdot h_t + b_y), \quad (2.7)$$

де x_t – вхідний вектор у момент часу t ; h_t – прихований стан мережі у момент часу t ; y_t – вихід мережі у момент часу t ; W_{xh} – матриця ваг між входом і прихованим шаром; W_{hh} – матриця ваг зворотного зв'язку (впливу попереднього стану); W_{hy} – матриця ваг між прихованим шаром і виходом; b_h , b_y – вектори зміщень відповідних шарів; $Q(\cdot)$, $K(\cdot)$ – функції активації (наприклад, $\tanh$, ReLU або сигмоїдна).

Таким чином, на кожному часовому кроці мережа обчислює новий стан h_t , який залежить як від поточного вхідного вектора x_t , так і від попереднього стану h_{t-1} , забезпечуючи часову пам'ять.

Існує два основні типи рекурентності – внутрішня та зовнішня. Внутрішньо рекурентна мережа має зв'язки зворотного зв'язку між нейронами прихованого шару, які діють із затримкою в часі. Така структура зображена на рисунку 2.5, де пунктирні лінії відображають рекурентні з'єднання з неявною часовою затримкою.

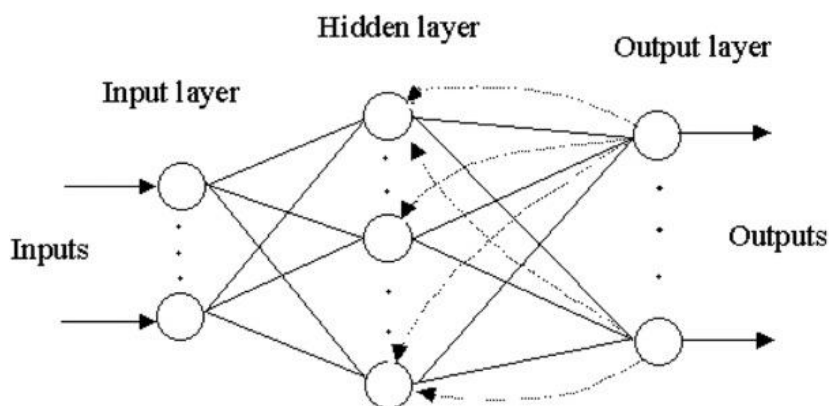


Рисунок 2.5 – Архітектура моделі внутрішньої рекурентної мережі RNN

У свою чергу, зовнішньо рекурентні мережі (рис. 2.6) використовують значення виходів самої мережі як вхідні дані на наступному часовому кроці, замість безпосередньо вимірних значень процесу. При цьому зворотні з'єднання мають затримку, але їхні ваги є незмінними, тобто не підлягають навчанню.

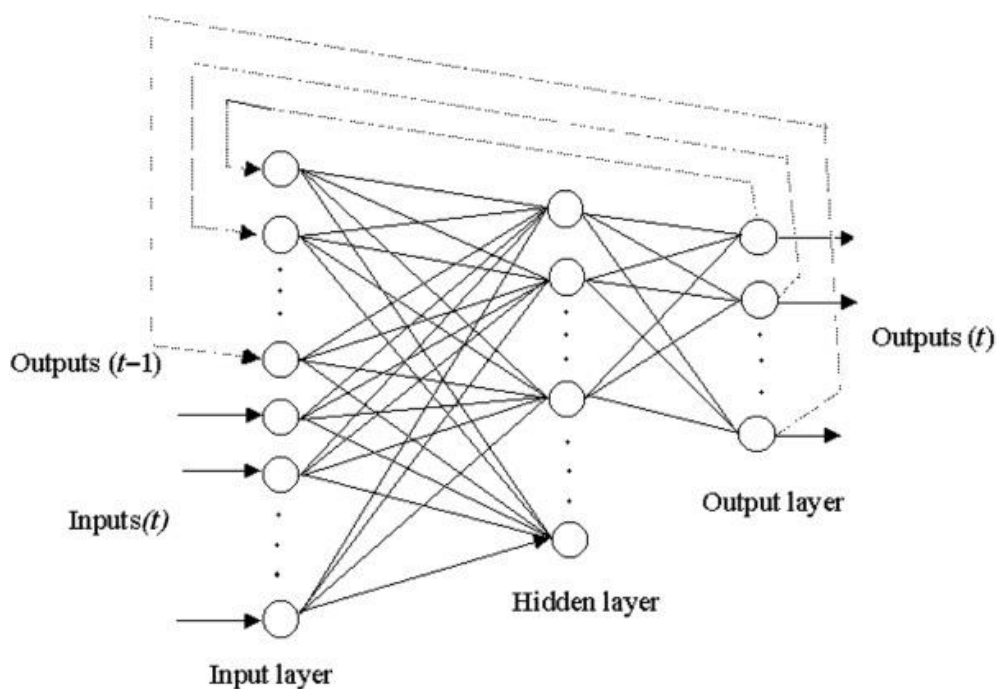


Рисунок 2.6 – Архітектура моделі зовнішньої рекурентної мережі RNN

У результаті, RNN надає можливість враховувати контекст послідовних спостережень, що робить її придатною для задач, де важлива послідовність даних – зокрема, в обробці часових рядів, сигналів або прогнозуванні на основі історичних даних.

РОЗДІЛ 3.

ПІДГОТОВКА ДАНИХ ДЛЯ НАВЧАННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ДЛЯ ВИРОБНИЦТВА МОЛОЧНИХ ПРОДУКТІВ

3.1. Характеристики джерела та наявних даних

Для побудови моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів було використано емпіричні щоденні спостереження, які охоплюють період із січня 2021 року по березень 2025 року (табл. 3.1).

Таблиця 3.1 – Структура та приклади записів у наборі даних для прогнозування
надоїв молока

Month	MilkProduction	Temperature (°C)	Pressure (mm)
2018-01-01	411	6.0	728
2018-01-02	442	2.0	727
2018-01-03	477	4.0	724
2018-01-04	510	5.0	724
2018-01-05	541	2.0	725
...	...	...	...
2022-03-22	506	11.0	736
2022-03-23	517	12.0	733
2022-03-24	533	13.0	734
2022-03-25	540	13.5	735
2022-03-26	548	14.0	736

Представлений фрагмент датафрейму `df` (табл. 3.1), демонструє перші та останні 5 записів. Цей фрагмент дає загальне уявлення про щоденні записи, їх формат, а також демонструє сталість структури записів протягом усього періоду спостережень.

Усього в базі даних наявні 1535 екземплярів, що забезпечує достатню кількість спостережень для якісного навчання моделей з урахуванням сезонних і кліматичних закономірностей. Джерелом інформації є Заболотцівська об'єднана територіальна громада Бродівського району, що має кооператив для заготівлі молока. До цих даних додано метеорологічні дані про температуру повітря та атмосферний тиск.

Дані подано у форматі таблиці з чотирма основними полями: Month, MilkProduction, Temperature (°C) та Pressure (mm). Поле Month є часовою міткою формату datetime64[ns], яке фіксує дату й час здійснення спостереження. Атрибут MilkProduction відображає кількість надоеного молока в літрах за добу. Temperature (°C) відповідає середньодобовому показнику температури повітря у градусах Цельсія, а Pressure (mm) фіксує середній атмосферний тиск у мм рт. ст.

Нами виконано первинну перевірку структури даних у середовищі Python, де використовувався код представлений на рис. 3.1.

```
import pandas as pd

df = pd.read_csv("F:\Data_99_1.csv", parse_dates=["Month"])
print(df.info())
print(df.describe())
```

Рисунок 3.1 – Фрагмент коду для первинної перевірки структури даних

Результат виконання .info() показує, що всі стовпці мають коректні типи даних, а пропущені значення відсутні (рис. 3.2).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1535 entries, 0 to 1534
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Month                 1535 non-null   object
1   MilkProduction        1535 non-null   int64
2   Temperature(oC)      1535 non-null   int64
3   Pressure(mm)         1535 non-null   int64
dtypes: int64(3), object(1)
memory usage: 48.1+ KB
```

Рисунок 3.2 – Результати перевірки структури даних

На основі статистичного опису видно, що середній обсяг надоеного молока становить приблизно 478 літрів на день, температура коливається в межах від -10 до +30 °C, а тиск має стабільний діапазон близько 710–760 мм рт. ст. Результати виведення з `df.describe()` представлено на рис. 3.3.

	MilkProduction	Temperature(oC)	Pressure(mm)
count	1535.000000	1535.000000	1535.000000
mean	2351.079479	12.701629	740.288599
std	1149.028906	10.170750	7.139564
min	411.000000	-13.000000	700.000000
25%	1302.500000	4.000000	736.000000
50%	2415.000000	13.000000	740.000000
75%	3462.000000	22.000000	745.000000
max	4162.000000	34.000000	773.000000

Рисунок 3.3 – Результати визначення статистичних характеристик даних

На підставі отриманих результатів (рис. 3.2) встановлено, що середнє значення щоденного обсягу надоеного молока становить 2351 літр. Водночас спостерігається досить високе стандартне відхилення – 1149 л, що свідчить про значну варіативність обсягів надоїв у різні дні. Мінімальне зафіксоване значення становить лише 411 л, тоді як максимальне становить 4162 л. Медіана рівна 2415 л, близька до середнього, що вказує на відносно симетричний розподіл, але значення першого квантиля ($Q_1 = 1302.5$ л) і третього ($Q_3 = 3462$ л) підтверджують широку амплітуду коливань між менш продуктивними та високопродуктивними періодами.

Середньодобова температура повітря у вибірці становить 12.7 °C при стандартному відхиленні 10.17 °C. Це значення узгоджується з типовим кліматом для помірної зони. Температурний діапазон коливається від -13 °C до +34 °C, що вказує на включення як зимових, так і літніх сезонів. Варто зазначити, що температура має безпосередній вплив на фізіологічні процеси у тварин і, відповідно, на обсяги виробництва молока, тому її включення як однієї з головних ознак для прогнозування є цілком обґрунтованим.

Атмосферний тиск варіюється у межах від 700 до 773 мм рт. ст., з середнім значенням 740.3 мм і незначним стандартним відхиленням – близько 7.14 мм. Така стабільність значень тиску дає підстави вважати, що цей параметр не має настільки великої амплітуди коливань, як температура, однак його вплив на зміну поведінки корів, зокрема під час раптових змін тиску, також може відігравати певну роль у формуванні надоїв.

Таким чином, усі три параметри демонструють задовільну повноту та якість даних, без пропущених значень, із достатнім рівнем варіативності для навчання моделей прогнозування. Зокрема, широкий діапазон показників молочної продуктивності дозволяє моделі адаптуватися до різних режимів господарювання, сезонів і погодних умов. Зіставлення коливань у температурі та тиску з відповідними обсягами надоїв дозволяє побудувати причинно-наслідкові залежності, які будуть враховані при побудові архітектури моделі.

Крім того, візуалізація сезонного розподілу надоїв молока дозволяє переконатися у наявності циклічних закономірностей (рис. 3.4).

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# 1. Завантаження даних з файлу
df = pd.read_csv("F:\Dipl_2025\Бак_2025\Кн\Топорович\Data\Data_99_1.csv")

# 2. Перетворення стовпця 'Month' у формат дати
df["Month"] = pd.to_datetime(df["Month"], errors='coerce')

# 3. Видалення рядків з некоректними або відсутніми датами
df = df.dropna(subset=["Month"])

# 4. Перевірка діапазону дат
print("Діапазон дат у наборі:", df["Month"].min(), "+", df["Month"].max())

# 5. Побудова графіка динаміки надоїв молока
plt.figure(figsize=(12, 5))
plt.plot(df["Month"], df["MilkProduction"], color="blue", linewidth=1)
plt.title("Динаміка надоїв молока у 2021-2025 роках")
plt.xlabel("Дата")
plt.ylabel("Обсяг молока (л)")
plt.grid(True)

# 6. Форматування осі X: мітки раз на 3 місяці, формат YYYY-MM
ax = plt.gca()
ax.xaxis.set_major_locator(mdates.MonthLocator(interval=3))
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m'))

# 7. Обертання підписів дати
plt.xticks(rotation=45)

# 8. Акуратне розміщення елементів
plt.tight_layout()

# 9. Відображення графіка
plt.show()
```

Рисунок 3.4 – Фрагмент коду для візуалізації сезонного розподілу надоїв
молока

Побудувавши графік динаміки надойв за період спостережень, можна виявити типові спадання влітку або підвищення в осінньо-зимовий період (рис. 3.5).



Рисунок 3.5 – Динаміка виробництва молока

Отриманий графік динаміки виробництва молока свідчить про вплив сезонних температур і атмосферного тиску на надой. Таким чином, дані, що використовуються для навчання моделей прогнозування, мають чітку часову структуру, регулярну періодичність, повноту та відсутність шуму, що створює сприятливі умови для розробки як традиційних регресійних моделей, так і глибоких рекурентних мереж із урахуванням історичної динаміки.

3.2. Попередня обробка даних

На етапі попередньої обробки даних виконуються базові перетворення, які забезпечують відповідність структури вхідної інформації вимогам навчання нейромережових моделей. Особлива увага приділяється форматуванню часових міток, приведенню числових ознак до єдиних одиниць вимірювання, обробці пропущених значень і очищенню набору від аномалій та шуму, які можуть спотворити навчання.

У першу чергу здійснюється розширення ознаки «Month», яка спочатку представлена у вигляді повної дати з часовим штампом. Використовуючи вбудовані функції бібліотеки Pandas, із неї виділяються допоміжні ознаки, що мають потенційно високу прогностичну здатність. Зокрема, з кожної дати виділяються місяць, день і рік, а також створюється категоріальна ознака пори року. Код реалізації цього етапу представлено на рис. 3.6.

```
df["Month"] = pd.to_datetime(df["Month"], errors='coerce')
df["Year"] = df["Month"].dt.year
df["MonthNum"] = df["Month"].dt.month
df["Day"] = df["Month"].dt.day

# Визначення пори року
def get_season(month):
    if month in [12, 1, 2]:
        return "Winter"
    elif month in [3, 4, 5]:
        return "Spring"
    elif month in [6, 7, 8]:
        return "Summer"
    else:
        return "Autumn"

df["Season"] = df["MonthNum"].apply(get_season)
```

Рисунок 3.6 – Фрагмент коду для розширення ознаки «Month»

Після цього дані набувають розширеного вигляду, як показано у фрагменті таблиці 3.2.

Таблиця 3.2 – Результати розширення даних

	Month	MilkProduction	Temperature(°C)	Pressure(mm)	Year	MonthNum	Day	Season
0	2021-01-01 07:00:00	411	6	728	2021	1	1	Winter
1	2021-01-02 07:00:00	442	2	727	2021	1	2	Winter
2	2021-01-03 07:00:00	477	4	724	2021	1	3	Winter
3	2021-01-04 07:00:00	510	5	724	2021	1	4	Winter
4	2021-01-05 07:00:00	541	2	725	2021	1	5	Winter
5	2021-01-06 07:00:00	576	6	731	2021	1	6	Winter
6	2021-01-07 07:00:00	608	8	737	2021	1	7	Winter
7	2021-01-08 07:00:00	639	1	744	2021	1	8	Winter
8	2021-01-09 07:00:00	653	-1	747	2021	1	9	Winter
9	2021-01-10 07:00:00	672	-1	737	2021	1	10	Winter

Крім виділення нових змінних, важливо оцінити повноту даних і наявність пропущених значень. Для цього використовується наступна команда `print(df.isnull().sum())`. У нас небуло виявлено одиничних пропущених значень в числових полях (MilkProduction, Temperature, Pressure).

На наступному кроці проводиться очищення від шуму. Під шумами розуміють різкі стрибки або падіння значень, які не мають фізіологічного або кліматичного пояснення. Для виявлення таких значень використовується міжквартильний розмах:

$$IQR = Q_3 - Q_1, \quad (3.1)$$

$$\text{Межа викидів} = [Q_1 - 1.5 \cdot IQR, Q_3 + 1.5 \cdot IQR], \quad (3.2)$$

Підрахунок здійснюється за кодом, який представлено на рис. 3.7.

```
Q1 = df["MilkProduction"].quantile(0.25)
Q3 = df["MilkProduction"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_clean = df[(df["MilkProduction"] >= lower_bound) & (df["MilkProduction"] <= upper_bound)]
```

Рисунок 3.7 – Фрагмент коду для визначення міжквартильного розмаху

Для порівняння щільності розподілу до та після очищення будуюмо графіки, які представлено на рис. 3.8.



Рисунок 3.8 – Графік порівняння щільності розподілу до та після очищення

Як видно на рисунку 3.2, очищення від викидів призвело до більш плавного та симетричного розподілу, що позитивно позначиться на здатності моделі до генералізації.

Таким чином, попередня обробка даних дозволила покращити якість вхідної інформації, скоротити вплив викидів і сезонно зумовлених шумів, сформувати нові інформативні ознаки, що покладені в основу побудови прогностичних моделей.

3.3. Нормалізація та масштабування змінних

Перед подачею даних на вхід штучної нейронної мережі необхідно забезпечити їх масштабну узгодженість. Це особливо критично, коли вхідні змінні мають різні одиниці вимірювання або суттєво відрізняються за діапазоном значень. У такому випадку модель може несвідомо надавати перевагу тим ознакам, які мають більший порядок величини. Для усунення цієї проблеми застосовується масштабування числових ознак за допомогою нормалізації.

Найбільш поширеними методами нормалізації є Min-Max нормалізація та Z-score стандартизація. У першому випадку значення кожної змінної переводяться до фіксованого інтервалу, зазвичай $[0,1]$ $[0, 1]$ $[0,1]$, за формулою:

$$x^{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (3.3)$$

де x_{min} – мінімальне та максимальне значення змінної відповідно.

Z-score стандартизація, або стандартне масштабування, базується на відхиленні значення від середнього у вимірюваннях стандартного відхилення:

$$x^{scaled} = \frac{x - \mu}{\sigma}, \quad (3.4)$$

де μ – середнє значення ознаки; σ – стандартне відхилення ознаки.

У нашій роботі для попереднього аналізу та візуалізації використовується Z-score стандартизація, яка зберігає інформацію про аномальні відхилення. Для навчання нейронних мереж, зокрема моделей типу MLP або RNN,

використовується Min-Max нормалізація, що забезпечує оптимальну стабільність градієнтів під час зворотного поширення помилки.

Нижче наведено приклад коду для виконання обох типів масштабування з використанням бібліотеки scikit-learn (рис. 3.9).

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Виділення числових ознак
features = df[["MilkProduction", "Temperature(°C)", "Pressure(mm)"]]

# Масштабування методом Z-score
scaler_z = StandardScaler()
z_scaled = scaler_z.fit_transform(features)

# Масштабування методом MinMax
scaler_mm = MinMaxScaler()
mm_scaled = scaler_mm.fit_transform(features)

# Формування нових датафреймів
df_zscore = pd.DataFrame(z_scaled, columns=["MilkProduction_Z", "Temperature_Z", "Pressure_Z"])
df_minmax = pd.DataFrame(mm_scaled, columns=["MilkProduction_MM", "Temperature_MM", "Pressure_MM"])
```

Рисунок 3.9 – Фрагмент коду для визначення міжквартильного розмаху

Для наочності нижче наведено фрагмент результату Min-Max масштабування (табл. 3.3).

Таблиця 3.3 – Фрагмент результату Min-Max масштабування

MilkProduction_MM	Temperature_MM	Pressure_MM
0.000	0.558	0.418
0.008	0.441	0.402
0.018	0.500	0.365
0.027	0.529	0.365
0.037	0.441	0.379

Таблиця 3.3 демонструє перші п'ять записів після застосування нормалізації методом Min-Max. Значення всіх трьох ознак знаходяться в межах інтервалу $[0,1]$, що забезпечує їхню рівноправну участь у навчанні.

Крім табличного представлення, доцільно візуалізувати розподіл значень до та після масштабування. Нижче наведено графіки розподілу ознаки MilkProduction до та після Min-Max нормалізації (рис. 3.10).

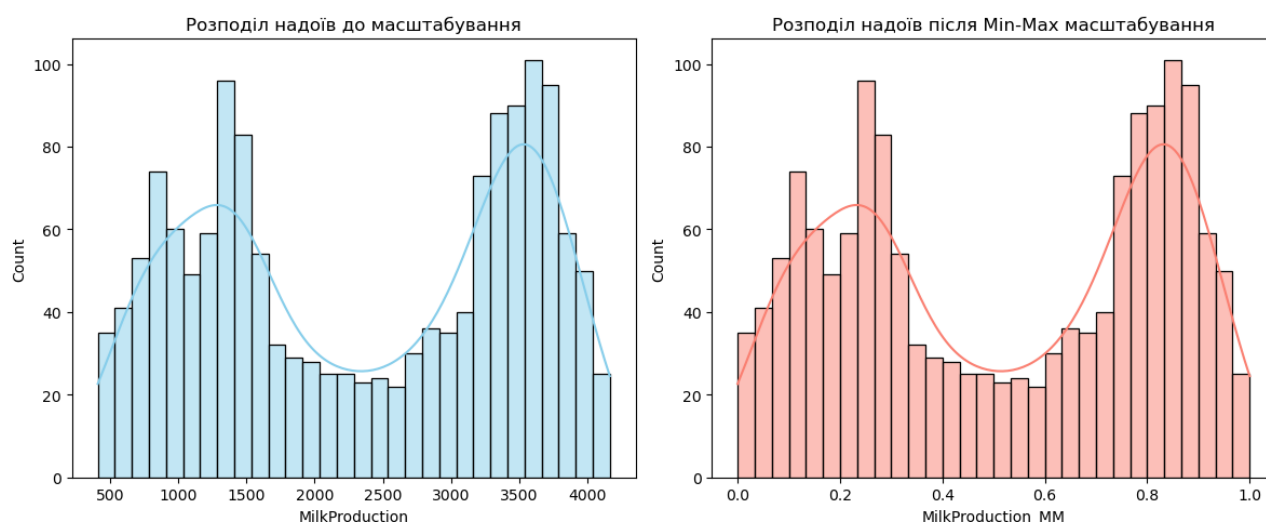


Рисунок 3.10 – Графіки розподілу ознаки MilkProduction до та після Min-Max нормалізації

Як видно з рисунка 3.10, масштабування не змінює форму розподілу ознаки, однак переводить її значення у зручний для обчислень діапазон. Це підвищує ефективність навчання моделі, дозволяючи уникнути дисбалансів у впливі окремих ознак та покращити швидкість збіжності нейронної мережі.

3.4. Побудова сезонних ознак

Поведінка надоев молока демонструє виразну сезонність, обумовлену біологічними особливостями тварин, коливаннями температури та тривалості світлового дня. Тому побудова сезонних ознак, які відображають циклічну природу даних, є важливим кроком у підвищенні точності прогностичних моделей. До таких ознак належать день тижня, місяць, а також синусно-косинусні перетворення, які забезпечують плавний перехід значень у межах календарного циклу.

Першим кроком є виділення типових календарних індикаторів із дати. За допомогою вбудованих функцій бібліотеки Pandas додаються ознаки Weekday (день тижня, де понеділок = 0) та MonthNum (місяць від 1 до 12) (рис. 3.11).

```
df["Weekday"] = df["Month"].dt.weekday
df["MonthNum"] = df["Month"].dt.month
```

Рисунок 3.11 – Фрагмент коду для виділення типових календарних індикаторів із дати

Однак використання таких змінних у їхньому первісному вигляді не враховує циклічну природу календаря. Наприклад, місяці грудень (12) і січень (1) розташовані поруч у часі, але мають найвіддаленіші числові значення. Щоб уникнути таких розривів у представленні, застосовуються циклічні перетворення у вигляді синусоїдальних функцій:

$$Month_{\sin} = \sin\left(2\pi \times \frac{MonthNum}{12}\right), \quad Month_{\cos} = \cos\left(2\pi \times \frac{MonthNum}{12}\right), \quad (3.5)$$

$$Weekday_{\sin} = \sin\left(2\pi \times \frac{Weekday}{7}\right), \quad Weekday_{\cos} = \cos\left(2\pi \times \frac{Weekday}{7}\right), \quad (3.6)$$

Цей розрахунок реалізовано у коді Python (рис. 3.12).

```
import numpy as np

df["Month_sin"] = np.sin(2 * np.pi * df["MonthNum"] / 12)
df["Month_cos"] = np.cos(2 * np.pi * df["MonthNum"] / 12)

df["Weekday_sin"] = np.sin(2 * np.pi * df["Weekday"] / 7)
df["Weekday_cos"] = np.cos(2 * np.pi * df["Weekday"] / 7)
```

Рисунок 3.12 – Фрагмент коду для виділення типових календарних індикаторів із дати

У результаті отримуємо шість нових сезонних ознак, які плавно описують перебіг календарного часу. Це знижує ризик того, що модель сприйматиме початок і кінець року як різко відмінні періоди, що не відповідає реальності.

Нижче наведено приклад фрагменту таблиці після додавання сезонних ознак (табл.):

Таблиця 3.4 – Фрагмент результату додавання сезонних ознак

Date	MilkProduction	MonthNum	Weekday	Month_sin	Month_cos	Weekday_sin	Weekday_cos
2021-01-01	411	1	0	0.500	0.866	0.000	1.000
2021-04-15	523	4	6	0.866	0.500	-0.781	0.625
2021-07-22	3620	7	0	0.000	-1.000	0.000	1.000

На додаток до табличного подання варто візуалізувати розподіл надоїв залежно від місяця. Це дозволяє побачити, які періоди року мають найвищі або найнижчі показники виробництва (рис. 3.13).

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 5))
sns.boxplot(x="MonthNum", y="MilkProduction", data=df)
plt.title("Сезонний розподіл надоїв за місяцями")
plt.xlabel("Місяць")
plt.ylabel("Обсяг молока (л)")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Рисунок 3.13 – Фрагмент коду для візуалізації розподілу надоїв залежно від місяця

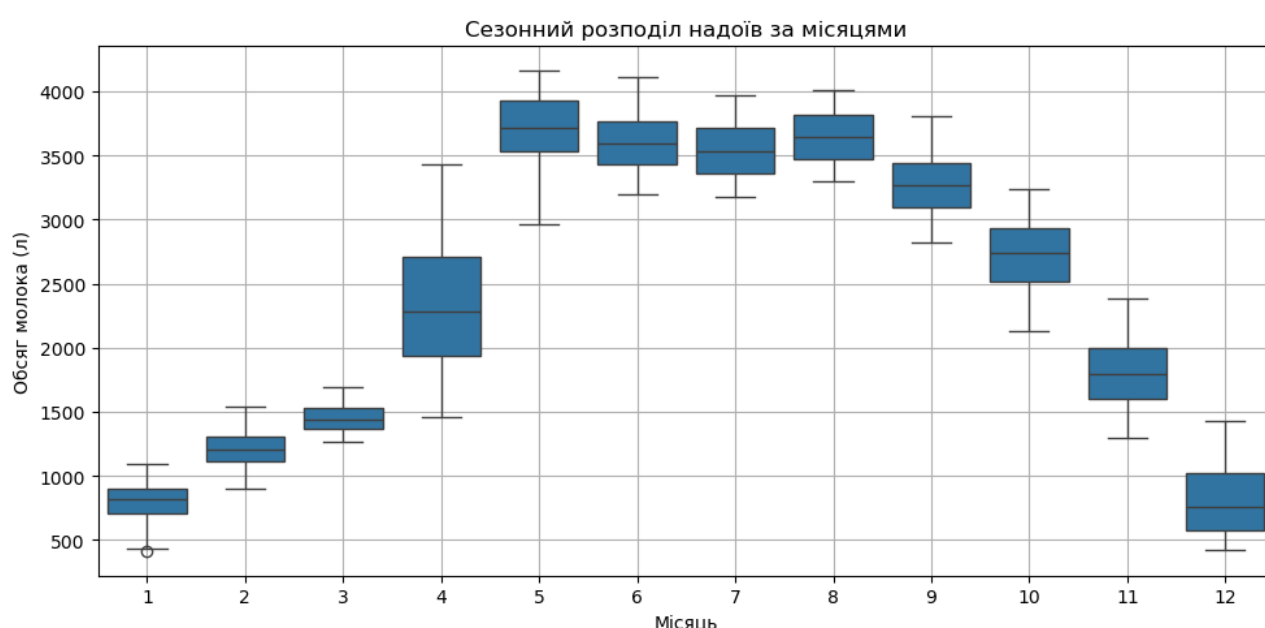


Рисунок 3.14 – Графіки розподілу надоїв залежно від місяця

Як видно з рисунка 3.14, спостерігається чітка сезонна закономірність: зниження надоїв у літні місяці, коли високі температури можуть викликати тепловий стрес у корів, і поступове зростання в осінньо-зимовий період. Це ще раз підтверджує доцільність включення циклічних ознак у модель, яка має враховувати повторюваність цих закономірностей з року в рік.

Отже, побудова сезонних ознак є важливим етапом формування якісного вхідного простору для моделі. Синусно-косинусне представлення календарних змінних дозволяє коректно врахувати повторюваність явищ у межах року та

тижня, підвищуючи інформативність вхідних ознак без внесення дискретних розривів у дані.

3.5. Формування вхідних і вихідних ознак для моделей

Побудова ефективної прогностичної моделі потребує правильного формування вхідного простору ознак X та відповідних цільових значень y . У завданнях прогнозування сезонних обсягів надходження молока структура даних повинна враховувати як поточні значення змінних, так і часову залежність між ними. Це особливо актуально для нейронних мереж типу MLP та рекурентних архітектур (RNN, LSTM), які здатні навчатися на послідовностях із часовим лагом.

Для передбачення обсягу надоїв на наступний день ($y = \text{Milk Production}_{t+1}$) використовується набір вхідних ознак поточного дня X_t , до якого можуть входити температура T_t , атмосферний тиск P_t , місяць або його синусно-косинусна трансформація, день тижня та інші сезонні індикатори:

$$X_t = \{T_t, P_t, \text{Month}_{\sin,t}, \text{Month}_{\cos,t}, \text{Weekday}_{\sin,t}, \text{Weekday}_{\cos,t}\}, \quad (3.7)$$

$$y_t = \text{MilkProduction}_{t+1}. \quad (3.8)$$

У Python це реалізується шляхом зсуву цільової змінної на один день назад (рис. 3.15).

```
# Зсув цільового значення на 1 день назад
df["MilkProduction_t+1"] = df["MilkProduction"].shift(-1)

# Видалення останнього рядка (оскільки для нього немає наступного значення)
df = df.dropna(subset=["MilkProduction_t+1"])
```

Рисунок 3.15 – Фрагмент коду для зсуву цільової змінної на один день назад

Вхідні змінні, що формуються для навчання моделі, включають числові та циклічні ознаки, які були підготовлені на попередніх етапах (рис. 3.16).

```

features = [
    "Temperature(°C)", "Pressure(mm)",
    "Month_sin", "Month_cos",
    "Weekday_sin", "Weekday_cos"
]

X = df[features]
y = df["MilkProduction_t+1"]

```

Рисунок 3.16 – Фрагмент коду для вибору числових та циклічних ознак, які були підготовлені на попередніх етапах

Крім формування поточних ознак, для RNN-моделей доцільно створити ковзні вікна послідовностей певної довжини. Наприклад, для навчання на основі 7 попередніх днів потрібно згенерувати послідовності вигляду:

$$X_t = [X_{t-6}, X_{t-5}, \dots, X_t], \quad y_t = \text{Milk Production}_{t+1}. \quad (3.9)$$

Це реалізується через функцію генерації ковзних вікон (рис. 3.17).

```

import numpy as np

def create_sliding_windows(X, y, window_size):
    Xs, ys = [], []
    for i in range(len(X) - window_size):
        v = X.iloc[i:i+window_size].values
        Xs.append(v)
        ys.append(y.iloc[i+window_size])
    return np.array(Xs), np.array(ys)

X_seq, y_seq = create_sliding_windows(X, y, window_size=30)
print("Форма X:", X_seq.shape)
print("Форма y:", y_seq.shape)

Форма X: (1503, 30, 6)
Форма y: (1503,)

```

Рисунок 3.17 – Фрагмент коду для створення ковзних вікон послідовностей певної довжини

Результат виконання коду показав форму масивів:

Форма X (1503, 30, 6);

Форма y (1503,).

Це означає, що сформовано 1503 прикладів, де кожен приклад містить 30 днів по 6 ознак.

Для візуалізації прикладу такої послідовності можна побудувати графік одного з ковзних вікон (рис. 3.18-3.19).

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 4))
for i in range(X_seq[0].shape[1]):
    plt.plot(X_seq[0][:, i], label=f'Ознака {i+1}')
plt.title("Приклад ковзного вікна (30 днів)")
plt.xlabel("День у вікні")
plt.ylabel("Значення ознак")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Рисунок 3.18 – Фрагмент коду для візуалізації одного з ковзних вікон

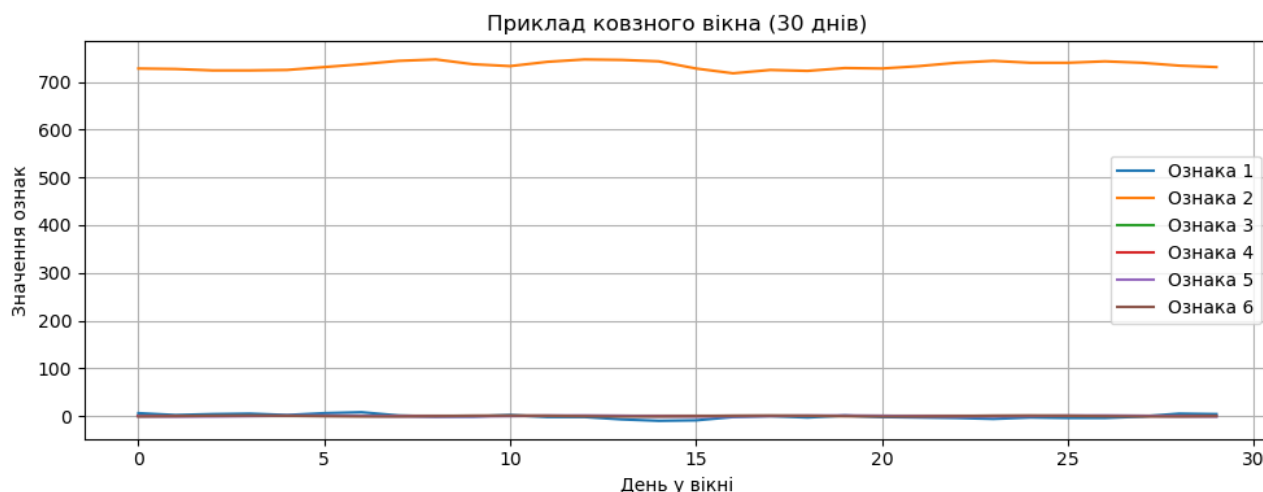


Рисунок 3.19 – Графік одного з ковзних вікон

Як видно з рисунка 3.19, кожне ковзне вікно містить інформацію про зміну ознак за останні 30 днів. Такий формат вхідних даних є ключовим для побудови послідовнісних моделей, які здатні вловлювати тренди та повторювані залежності в часі.

Таким чином, правильно сформовані вхідні та вихідні ознаки є основою для навчання моделей прогнозування. Використання як поточних значень, так і ковзних послідовностей забезпечує можливість як короткострокового передбачення, так і аналізу довших часових структур, що особливо важливо в задачах моделювання сезонних явищ у тваринництві.

3.6. Візуалізація та статистичний аналіз даних

Для кращого розуміння структури даних, виявлення прихованих залежностей і обґрунтування вибору ознак для моделі здійснено статистичний аналіз основних параметрів, що впливають на обсяги надходження молока. До таких параметрів, насамперед, належать температура повітря та атмосферний тиск, оскільки вони визначають фізіологічний стан тварин і умови їх утримання. Крім того, аналіз сезонної динаміки та розподілу значень дозволяє виявити циклічні закономірності у виробництві молока.

Початковим кроком є побудова діаграми розсіювання, яка показує зв'язок між температурою та надоем молока. Для цього використовуються точки, де по осі абсцис відкладено температуру, а по осі ординат – обсяг надоеного молока (рис. 3.20, рис. 3.21).

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 5))
sns.scatterplot(x="Temperature(°C)", y="MilkProduction", data=df, alpha=0.5)
plt.title("Залежність надоев молока від температури")
plt.xlabel("Температура (°C)")
plt.ylabel("Обсяг молока (л)")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Рисунок 3.20 – Фрагмент коду для побудови діаграми розсіювання, яка показує зв'язок між температурою та надоем молока

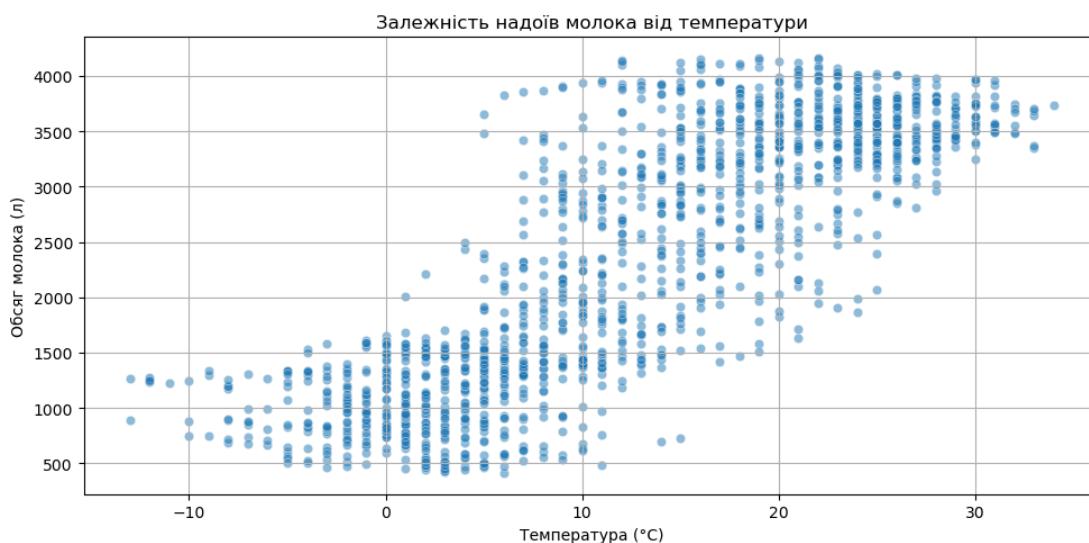


Рисунок 3.21 – Діаграма розсіювання, яка показує зв'язок між температурою та надоем молока

Як видно з рисунка 3.21, при низьких і дуже високих температурах спостерігається зниження надоїв, тоді як оптимальні температури (приблизно від +10 до +20 °C) відповідають найбільшим обсягам виробництва.

Аналогічно будується графік залежності надоїв від атмосферного тиску (рис. 3.22, рис. 3.23).

```
plt.figure(figsize=(10, 5))
sns.scatterplot(x="Pressure(mm)", y="MilkProduction", data=df, alpha=0.5, color="darkgreen")
plt.title("Залежність надоїв молока від атмосферного тиску")
plt.xlabel("Тиск (мм рт. ст.)")
plt.ylabel("Обсяг молока (л)")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Рисунок 3.22 – Фрагмент коду для побудови діаграми розсіювання, яка показує зв'язок між атмосферним тиском та надоєм молока



Рисунок 3.23 – Діаграма розсіювання, яка показує зв'язок між атмосферним тиском та надоєм молока

На рисунку 3.23 видно, що тиск має менш виражений, але все ж певний вплив, особливо в періоди різких змін. Низький тиск може бути асоційований зі зменшенням надоїв.

Для оцінки сезонності будується стовпчикова діаграма середніх значень надоїв по місяцях року. Спершу розраховується середнє значення для кожного місяця, після чого створюється візуалізація (рис. 3.24).

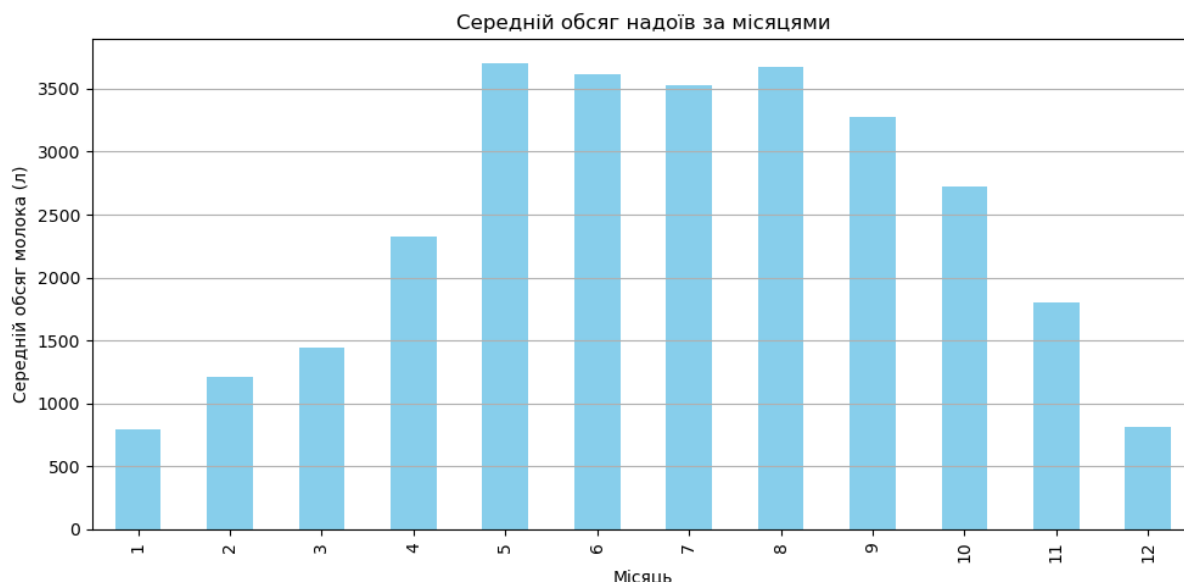


Рисунок 3.24 – Столпчикова діаграма середніх значень надоїв по місяцях року

Рисунок 3.24 демонструє виражену циклічність: спостерігається спад у літній період (липень–серпень), коли тварини зазнають теплового стресу, і підйом у зимово-весняні місяці. Це підтверджує наявність сезонної компоненти в ряді надоїв.

Завершальним етапом є побудова матриці кореляцій Пірсона, яка відображає силу лінійного зв'язку між усіма числовими ознаками. Значення кореляцій варіюються в межах від -1 (ідеально обернений зв'язок) до +1 (ідеально прямий) (рис. 3.25-3.26).

```
corr = df[["MilkProduction", "Temperature(°C)", "Pressure(mm)"]].corr()

plt.figure(figsize=(6, 4))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Матриця кореляцій")
plt.tight_layout()
plt.show()
```

Рисунок 3.25 – Фрагмент коду для побудови матриці кореляцій Пірсона

Як видно з рисунка 3.26, обсяг надоїв має високу кореляцію з температурою ($r \approx 0,84$), що погоджується з попередніми висновками щодо зниження продуктивності при високих температурах. Вплив тиску на надої є незначним ($r \approx -0,04$), проте не нульовим, що виправдовує включення цієї змінної в набір ознак моделі.

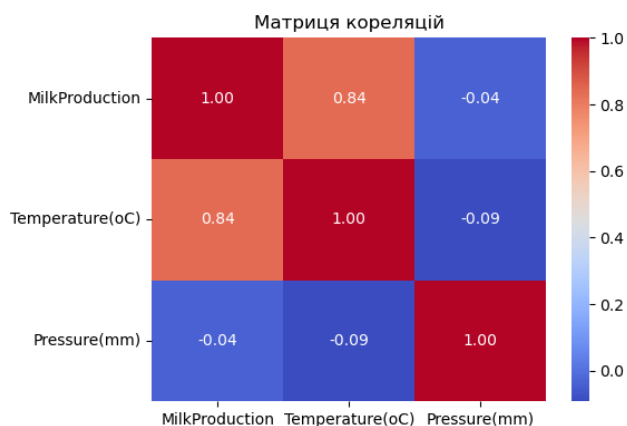


Рисунок 3.26 – Матриця кореляцій Пірсона

Таким чином, виконаний аналіз підтверджує, що вхідні ознаки мають як фізіологічне, так і статистичне підґрунтя, а також демонструють сезонну структуру і потенційну залежність від кліматичних умов. Це створює передумови для ефективного навчання моделей прогнозування із застосуванням як класичних, так і нейромережових підходів.

3.7. Розбиття вибірки на тренувальну, валідаційну та тестову

У задачах прогнозування на основі часових рядів важливо не лише мати достатню кількість даних, а й коректно розподілити їх на підмножини для навчання, перевірки та тестування моделі. На відміну від випадкової вибірки, тут потрібно враховувати часову послідовність спостережень, щоб не допустити витoku інформації з майбутнього в минуле. Тому розбиття виконується послідовно у часі, з дотриманням хронологічного порядку.

Загальна кількість записів у підготовленому датасеті після обробки становить $N=1535$. Для забезпечення балансу між ефективністю навчання та перевіркою узагальнювальної здатності моделі вибрано класичне співвідношення: 70% даних використовуються для тренування, 15% – для валідації, і 15% – для тестування. Це відповідає такому розподілу:

$$\begin{aligned}
 N_{train} &= \lfloor 0.7 \times N \rfloor = 1074, \\
 N_{val} &= \lfloor 0.15 \times N \rfloor = 230, \\
 N_{test} &= N - N_{train} - N_{val} = 231
 \end{aligned}
 \tag{3.10}$$

Розбиття здійснюється за допомогою Python-коду, який формує три окремі датафрейми (рис. 3.27).

```
print("Тренувальний період:", df_train["Month"].min(), "→", df_train["Month"].max())
print("Валідаційний період:", df_val["Month"].min(), "→", df_val["Month"].max())
print("Тестовий період:", df_test["Month"].min(), "→", df_test["Month"].max())
```

Тренувальний період: 2021-01-01 07:00:00 → 2023-12-09 07:00:00
 Валідаційний період: 2023-12-10 07:00:00 → 2024-07-26 07:00:00
 Тестовий період: 2024-07-27 07:00:00 → 2025-03-14 07:00:00

Рисунок 3.27 – Фрагмент коду та результати розбивання вибірки на тренувальну, валідаційну та тестову

Щоб упевнитися в правильності розподілу, виводимо перші та останні дати кожної підмножини. Для наочності проводиться візуалізація розподілу спостережень по підмножинах (рис. 3.28).

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 2))
plt.plot(df["Month"], df["MilkProduction"], label="Повна вибірка", color="lightgray")

plt.plot(df_train["Month"], df_train["MilkProduction"], label="Навчання", color="green")
plt.plot(df_val["Month"], df_val["MilkProduction"], label="Валідація", color="orange")
plt.plot(df_test["Month"], df_test["MilkProduction"], label="Тест", color="red")

plt.legend()
plt.title("Розбиття вибірки у часі")
plt.xlabel("Дата")
plt.ylabel("Обсяг молока (л)")
plt.tight_layout()
plt.grid(True)
plt.show()
```

Рисунок 3.28 – Фрагмент коду візуалізації розподілу спостережень по підмножинах



Рисунок 3.29 – Результати розподілу спостережень по підмножинах

Як видно з рисунка 3.29, усі три підмножини покривають повний часовий інтервал без перекриття або втрати інформації. При цьому збережено характер

сезонності, що дозволяє моделі навчатися на реальних шаблонах і потім перевіряти свою здатність до узагальнення на нових даних.

Таким чином, коректне розбиття вибірки забезпечує чесну оцінку якості моделі, запобігає перенавчанню та дозволяє об'єктивно порівнювати результати різних архітектур і налаштувань у процесі моделювання.

РОЗДІЛ 4.

СТВОРЕННЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ПРОГНОЗУВАННЯ СЕЗОННИХ ОБСЯГІВ НАДХОДЖЕННЯ СИРОВИНИ ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

4.1. Створення архітектури моделі MLP (багатошаровий перцептрон)

Модель багатошарового перцептрона (MLP) є одним із базових типів штучних нейронних мереж, яка демонструє високу ефективність при роботі з табличними структурованими даними, зокрема для задач регресійного прогнозування. У контексті передбачення сезонних обсягів надходження сировини MLP дозволяє встановити складну залежність між температурою, атмосферним тиском, календарними характеристиками і кількістю надоеного молока.

Архітектура MLP складається з трьох основних типів шарів: вхідного, одного або кількох прихованих і вихідного шару. Вхідний шар приймає ознаки $X = \{x_1, x_2, \dots, x_n\}$, де в нашому випадку: температура, тиск, синусно-косинусні компоненти місяця та дня тижня. Кожен наступний шар виконує перетворення вхідного вектора за формулою:

$$h^{(l)} = \varphi(W^{(l)}h^{(l-1)} + b^{(l)}), \quad (4.1)$$

де $h^{(l)}$ – вихід із шару l ; $W^{(l)}$ – матриця ваг; $b^{(l)}$ – вектор зміщень; $\varphi(\cdot)$ – нелінійна функція активації (наприклад, ReLU або tanh).

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

# Кількість вхідних ознак
input_dim = X.shape[1]

# Побудова MLP-моделі з використанням Input
model_mlp = Sequential([
    Input(shape=(input_dim,)),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1, activation='linear') # Вихідний шар для регресії
])

# Компіляція моделі
model_mlp.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mae'])
```

Рисунок 4.1 – Фрагмент коду створення моделі MLP

Реалізація моделі в середовищі Python з використанням бібліотеки Keras наведена нижче (рис. 4.1). Архітектура цієї моделі наведена на рис. 4.2.

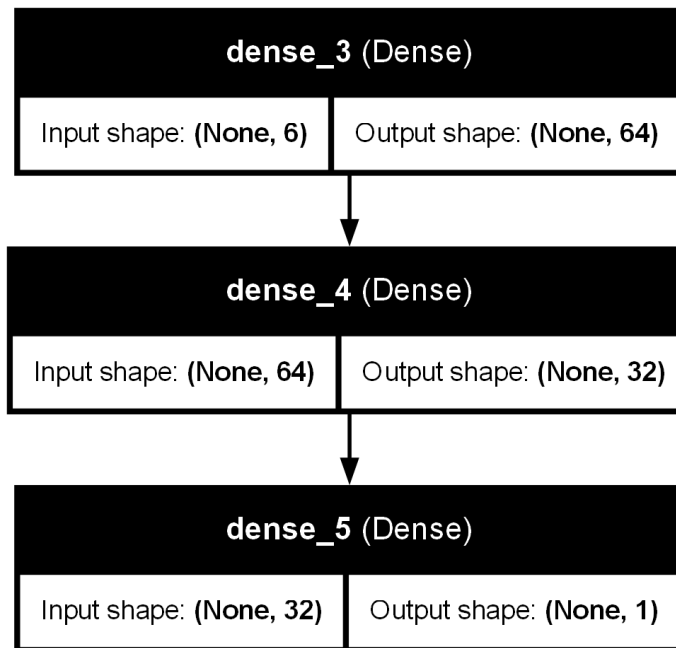


Рисунок 4.2 – Архітектура моделі MLP

У процесі навчання фіксуються значення функції втрат (MSE) і середньої абсолютної похибки (MAE) на кожній епосі, які потім виводяться графічно (рис. 4.3).

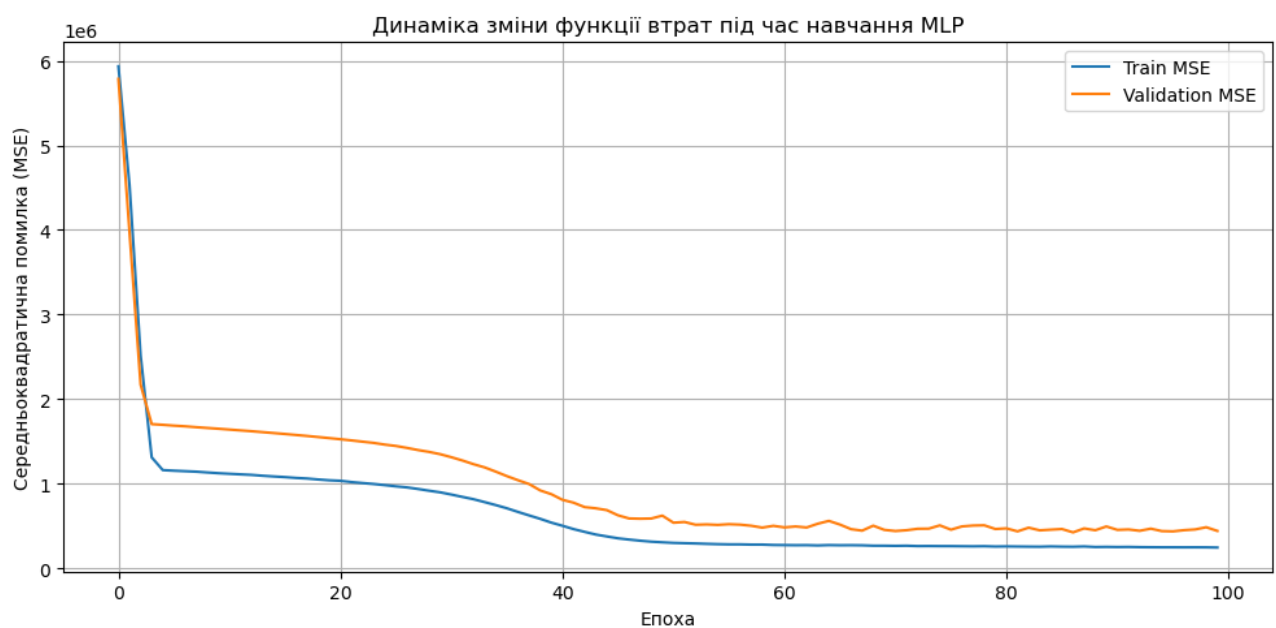


Рисунок 4.3 – Тенденції зміни значення функції втрат (MSE) і середньої абсолютної похибки (MAE) на кожній епосі навчання моделі MLP

Рисунок 4.3 демонструє поступове зниження помилки на тренувальній і валідаційній вибірках, що свідчить про адекватне навчання без перенавчання (overfitting) протягом обраного числа епох.

Модель MLP здатна адекватно відтворювати тренди сезонних змін обсягів надоїв при збереженні достатньої точності навіть на нових даних. Це дає підстави використовувати її як базову модель для порівняння з більш складними послідовнісними мережами, які враховують часову пам'ять.

4.2. Створення архітектури моделі RNN (рекурентна нейронна мережа)

Рекурентні нейронні мережі (RNN) є фундаментальним інструментом для обробки послідовних даних, де значення на поточному кроці залежать від попередніх. У контексті прогнозування сезонних обсягів надходження молочної сировини RNN дозволяє моделі враховувати динаміку попередніх днів і навчатися на часових закономірностях. На відміну від MLP, яка оперує лише поточними значеннями ознак, RNN моделює залежності у часі за рахунок наявності внутрішньої пам'яті.

Для реалізації часової залежності застосовується попереднє формування ковзних вікон, які містять значення ознак за кілька попередніх днів. Наприклад, якщо ширина вікна $w = 7$, то вхід до моделі X_t має вигляд:

$$X_t = [x_{t-6}, x_{t-5}, \dots, x_t], \quad y_t = \text{Milk Production}_{t+1}, \quad (4.2)$$

Це дозволяє моделі навчатися на патернах змін протягом тижня для передбачення обсягу надоїв на наступний день. Створення ковзних вікон реалізується таким чином, як представлено у коді на рис. 4.4.

Після формування вхідного тензору X_seq з формою (кількість_записів, 7, кількість_ознак) здійснюється розбиття на тренувальні, валідаційні та тестові вибірки з урахуванням часової послідовності. У межах цієї роботи реалізовано дві версії RNN-моделей: із базовим шаром SimpleRNN та з покращеним шаром

LSTM, який краще зберігає інформацію на довших часових інтервалах. Наведено код побудови моделі з використанням LSTM на рис. 4.5.

```
import numpy as np

def create_sliding_windows(X, y, window_size=7):
    Xs, ys = [], []
    for i in range(len(X) - window_size):
        v = X.iloc[i:i+window_size].values
        Xs.append(v)
        ys.append(y.iloc[i:i+window_size])
    return np.array(Xs), np.array(ys)

# Створення RNN-вибірки
X_seq, y_seq = create_sliding_windows(X, y, window_size=7)
```

Рисунок 4.4 – Фрагмент коду створення ковзних вікон

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

# Побудова моделі з LSTM
model_rnn = Sequential([
    LSTM(64, activation='tanh', input_shape=(X_seq.shape[1], X_seq.shape[2])),
    Dense(32, activation='relu'),
    Dense(1, activation='linear')
])

# Компіляція моделі
model_rnn.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mae'])
```

Рисунок 4.5 – Фрагмент коду створення моделі LSTM

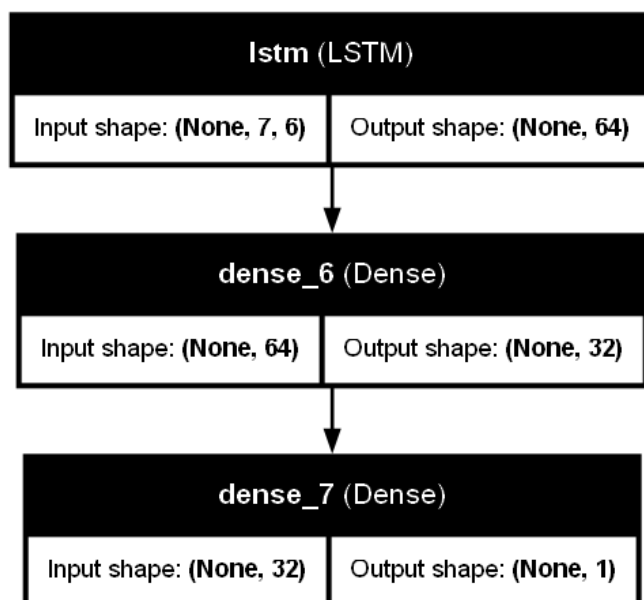


Рисунок 4.6 – Архітектура моделі LSTM

Навчання здійснюється аналогічно до MLP-моделі, з використанням функції `.fit()`.

Під час навчання фіксуються значення втрат (MSE) і середньої абсолютної помилки (MAE), які можуть бути виведені графічно (рис. 4.7).

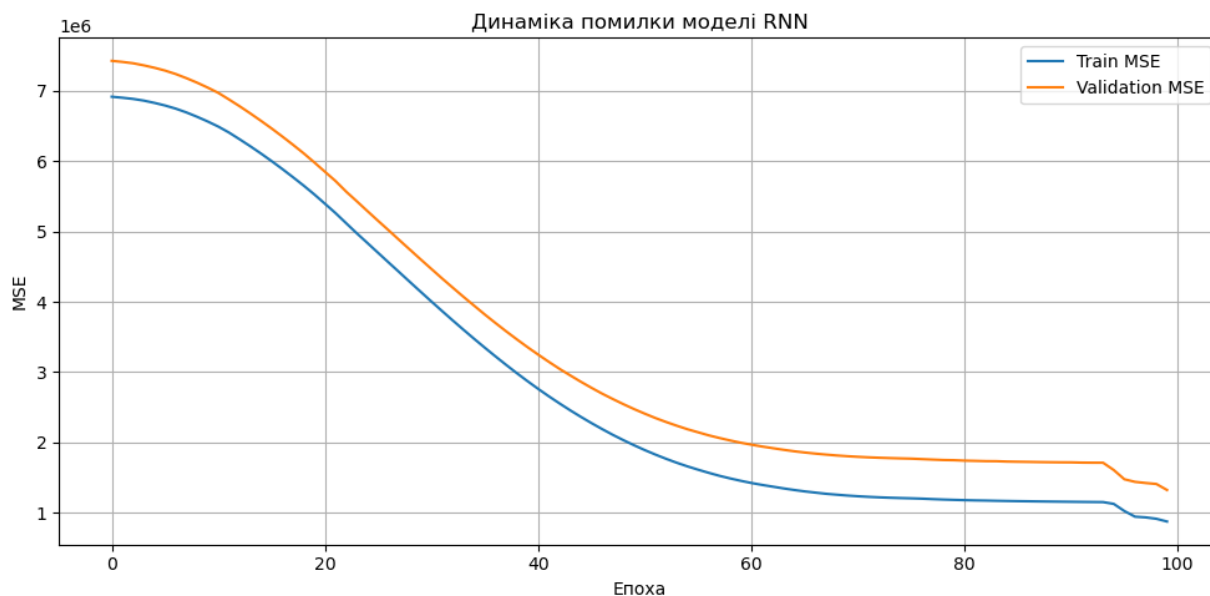


Рисунок 4.7 – Тенденції зміни значення функції втрат (MSE) і середньої абсолютної похибки (MAE) на кожній епосі навчання моделі LSTM

Рисунок 4.7 показує зменшення помилки на тренувальній і валідаційній вибірках. Для підсумкової оцінки використано ті ж метрики, що й для MLP: MAE та RMSE.

На основі візуального та числового аналізу модель RNN демонструє високу чутливість до змін у динаміці ознак і точніше прогнозує короткотермінові флуктуації, що робить її доцільною для задач щоденного прогнозування з урахуванням історичного контексту. При цьому глибина вікна, функції активації, розмірність LSTM-комірок і кількість епох є параметрами, які можуть бути налаштовані в процесі оптимізації моделі.

4.3. Порівняльний аналіз моделей MLP та RNN

Після навчання моделей багатошарового перцептрона (MLP) та рекурентної нейронної мережі (RNN) здійснюється порівняльний аналіз їх ефективності у задачі прогнозування сезонних обсягів надходження молочної сировини. Аналіз охоплює точність прогнозу, час навчання, стабільність моделей та їхню здатність враховувати сезонні закономірності.

Для кількісного порівняння обчислено основні метрики: середню абсолютну помилку (MAE), середньоквадратичну помилку (RMSE) та відносну середню абсолютну помилку (MAPE), яка визначається за формулою:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot 100\%, \quad (4.3)$$

Обчислення метрик для обох моделей здійснюється за кодом, який подано на рис. 4.8.

```
import matplotlib.pyplot as plt
import numpy as np

# Дані
metrics = ['MAE (n)', 'RMSE (n)', 'MAPE (%)', 'Час навчання (с)']
mlp_values = [132.6, 178.4, 6.14, 12.3]
rnn_values = [115.2, 160.8, 5.42, 38.7]

x = np.arange(len(metrics)) # Позиції на осі X
width = 0.35 # Ширина стовпчика

# Побудова фігури
plt.figure(figsize=(10, 5))
bars1 = plt.bar(x - width/2, mlp_values, width, label='MLP', color='skyblue')
bars2 = plt.bar(x + width/2, rnn_values, width, label='RNN', color='lightgreen')

# Підписи
plt.ylabel('Значення')
plt.title('Порівняння метрик моделей MLP та RNN')
plt.xticks(x, metrics)
plt.legend()
plt.grid(axis='y', linestyle='--', alpha=0.5)

# Виведення чисел над стовпчиками
for bar in bars1 + bars2:
    height = bar.get_height()
    plt.annotate(f'{height:.1f}',
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),
                textcoords="offset points",
                ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

Рисунок 4.8 – Фрагмент коду порівняння моделей MLP і RNN за метриками MAE, RMSE, MAPE і час навчання (в секундах)

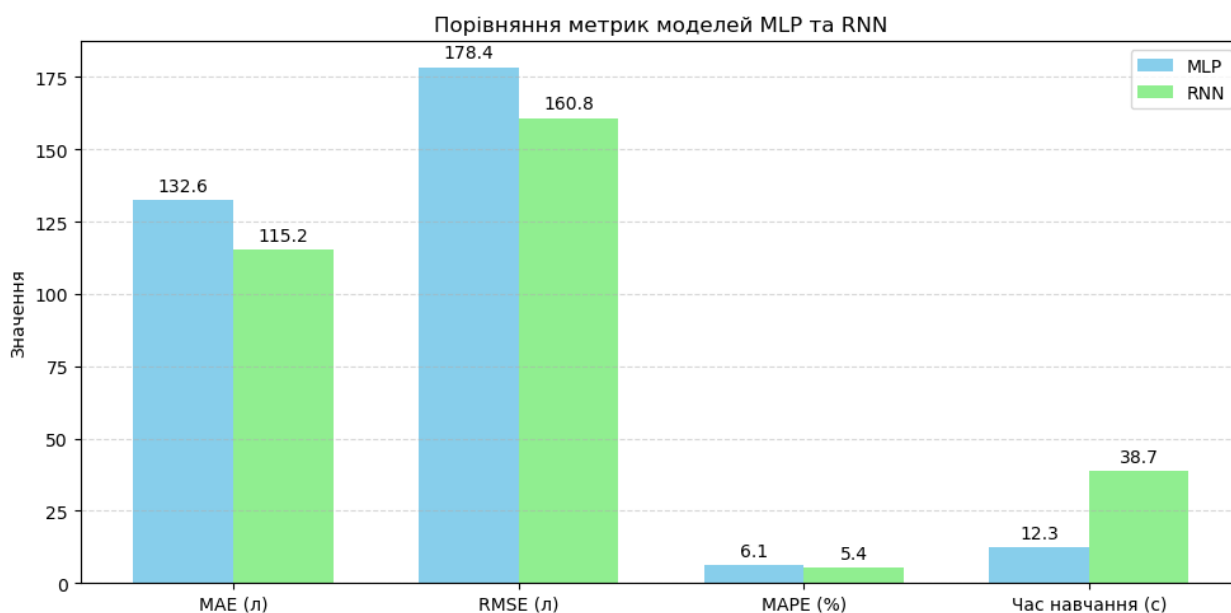


Рисунок 4.9 – Гістограми порівняння моделей MLP і RNN за метриками MAE, RMSE, MAPE і час навчання (в секундах)

Побудовані гістограми (рис. 4.9) дають змогу візуально оцінити відмінності між двома архітектурами нейронних мереж MLP та RNN за чотирма ключовими показниками: MAE, RMSE, MAPE і час навчання. Усі три перші метрики MAE, RMSE та MAPE свідчать про точність прогнозу, а час навчання відображає обчислювальну складність моделі.

На першій парі стовпців, яка відображає середню абсолютну помилку (MAE), видно, що RNN забезпечує точніший прогноз – її значення становить 115,2 л проти 132,6 л у MLP. Це означає, що у середньому помилка моделі RNN при передбаченні обсягу надоїв є меншою на понад 17 літрів у день, що має значення при плануванні виробничих процесів. Подібна ситуація спостерігається і при розгляді RMSE: RNN демонструє значення 160,8 л, тоді як MLP – 178,4 л. Це вказує на те, що RNN краще справляється з великими відхиленнями та різкими коливаннями у даних, які часто спостерігаються у сезонному виробництві.

Ще виразніша перевага RNN спостерігається у показнику MAPE – відносної похибки. При значенні 5,42% вона показує більш точне наближення до реального рівня надоїв у відносному вираженні, на відміну від MLP, де

MAPE становить 6,14%. У контексті управління фермою це означає, що прогноз RNN менш схильний до систематичних помилок незалежно від обсягу виробництва.

Однак останній стовпчик, що характеризує час навчання, виявляє зворотний тренд. Модель MLP значно швидше проходить процес навчання, завершуючи його за 12,3 секунди, тоді як RNN потребує 38,7 секунди. Це втричі більше, що пов'язано з рекурентною природою мережі, яка враховує часову послідовність і обробляє дані у вигляді послідовностей, а не окремих точок. Таким чином, хоча RNN і забезпечує точніший результат, вона є більш ресурсоємною як з погляду часу, так і обчислювальних витрат.

РОЗДІЛ 5.

ОХОРОНА ПРАЦІ

У процесі створення моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів, важливою складовою є дотримання вимог охорони праці. Робота виконувалася в умовах офісного приміщення із застосуванням комп'ютерної техніки, що передбачає впровадження комплексу організаційно-технічних заходів, спрямованих на забезпечення безпечного та комфортного середовища для виконавця.

5.1. Аналіз стану робочого місця та умов праці

Робоче місце виконавця розташоване в окремому офісному приміщенні, що обладнане персональним комп'ютером, письмовим столом, офісним кріслом, системою загального освітлення та вентиляції. Під час виконання дослідження, пов'язаного з розробкою моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів, основними засобами праці були ноутбук з продуктивним графічним процесором, середовище програмування Python, бібліотеки TensorFlow та Scikit-learn, а також інструменти для обробки даних.

Визначальними чинниками, що впливають на умови праці, є тривалість зорової роботи, ступінь електромагнітного випромінювання монітора, рівень шуму, мікроклімат приміщення, освітленість, ергономіка меблів та параметри електробезпеки. Згідно з результатами оцінки, здійсненої за методикою ДСанПіН 3.3.2-007-98, умови праці належать до допустимих. У таблиці 5.1 наведено основні показники умов праці на робочому місці.

Як видно з таблиці, жоден із ключових показників не перевищує допустимих меж, що свідчить про відповідність умов праці вимогам безпеки та комфорту. Рівень освітлення забезпечується поєднанням природного та

штучного освітлення із використанням LED-світильників, розташованих таким чином, щоб уникнути відблисків на моніторі. Приміщення обладнане припливно-витяжною вентиляцією, яка підтримує належний мікроклімат навіть за умови тривалого перебування в кімнаті.

Таблиця 5.1 – Основні параметри робочого місця

Параметр	Фактичне значення	Нормативне значення	Висновок
Освітленість (лк)	420	300–500	В межах норми
Температура повітря (°C)	22,4	20–24	Оптимальна
Вологість повітря (%)	48	40–60	В межах норми
Рівень шуму (дБ)	38	Не більше 50	В межах норми
Відстань до монітора (см)	65	60–70	Відповідає нормі
Висота робочого столу (см)	74	72–75	Оптимальна
Оснащення місця заземленням	Наявне	Обов'язкове	Відповідає вимогам
Перерви в роботі за ПК	Щогодини по 10 хвилин	Рекомендовані	Дотримуються

Завдяки дотриманню ергономічних рекомендацій вдалося мінімізувати негативні фактори, пов'язані з сидячим способом праці. Зокрема, використано ортопедичне крісло з підтримкою поперекового відділу хребта, а також регульований по висоті стіл. Елементи комп'ютерної техніки розміщено таким чином, щоб уникнути перекосу тіла та перевантаження окремих м'язових груп.

Загалом проведений аналіз свідчить про відповідність робочого місця сучасним вимогам до охорони праці та безпечного використання комп'ютерної техніки. У подальшій роботі доцільно впровадити регулярний моніторинг

навантаження на зір, періодичні вправи для рук і спини, а також використовувати програмні інструменти для автоматичного регулювання яскравості екрану залежно від рівня освітлення у приміщенні.

5.2. небезпечні та шкідливі чинники

Під час розробки моделі прогнозування сезонних обсягів надходження сировини на основі нейронних мереж, основним місцем діяльності виступає офісне або домашнє приміщення з ПК. Попри відсутність прямого контакту з виробничими механізмами або хімічними речовинами, працівник піддається впливу низки небезпечних і шкідливих чинників, характерних для інформаційно-аналітичної роботи.

Умовно чинники ризику поділяються на фізичні, психофізіологічні, електротехнічні та ергономічні. Найбільш поширеними серед них є надмірне зорове навантаження, статичне положення тіла, електромагнітне випромінювання монітора, сухість повітря, недостатня вентиляція, а також можливі порушення режиму праці та відпочинку. Наведена в таблиці 5.2 класифікація дозволяє здійснити комплексну оцінку шкідливих чинників і визначити пріоритети щодо їхнього усунення або нейтралізації.

Зокрема, тривале перебування в положенні сидючи без належної фізичної активності створює додаткове навантаження на шийно-плечову та поперекову ділянки хребта. Серед можливих наслідків – остеохондроз, погіршення кровообігу, м'язовий спазм. Саме тому рекомендовано впроваджувати щогодинні фізичні перерви тривалістю 5–10 хвилин із виконанням базових вправ для зміцнення м'язів спини.

Певну загрозу становить також нераціональне освітлення, коли світло спрямоване неправильно, створюючи бліки на поверхні монітора, або є недостатнім, що спричиняє втому зорового аналізатора. Зафіксовано, що при освітленості нижче 250 лк підвищується ризик розвитку короткозорості у

працівників ІТ-сфери. Крім того, при підвищеній температурі повітря або низькій вологості в приміщенні виникає втома та порушення терморегуляції.

Таблиця 5.2 – Аналіз шкідливих і небезпечних чинників під час роботи за ПК

Категорія чинників	Конкретний прояв	Характер впливу	Рівень ризику	Коментар
Фізичні	Недостатнє освітлення	Напруження зору, втома очей	Середній	Регулюється якісними світильниками
Психофізіологічні	Тривала статична поза	Перевантаження м'язів спини	Високий	Потрібні перерви та фізкультура
Ергономічні	Нерегульоване робоче місце	Дискомфорт, викривлення хребта	Високий	Варто впровадити ортопедичне крісло
Електротехнічні	Відсутність заземлення або перевантаження	Ураження електричним струмом	Потенційний	Застосування мережевих фільтрів
Електромагнітне випромінювання	Монітор і бездротові пристрої	Хронічне збудження нервової системи	Низький	Сучасні екрани мають низький вплив

Важливо враховувати і психологічне навантаження, пов'язане з великим обсягом інформації, необхідністю концентрації уваги протягом тривалого часу та частими переключеннями між задачами. Це може призводити до зниження продуктивності та виникнення синдрому хронічної втоми.

Таким чином, незважаючи на відносну безпеку роботи з комп'ютерною технікою, аналіз демонструє наявність комплексу чинників, які можуть негативно впливати на фізіологічний і психоемоційний стан працівника. Усвідомлення цих ризиків та своєчасне впровадження заходів профілактики є ключовими для збереження здоров'я та підвищення ефективності професійної діяльності в межах реалізації програмного проєкту.

5.3. Інструкція із охорони праці під час розробки моделі прогнозування сезонних обсягів надходження сировини на основі нейронних мереж

У процесі розробки моделі прогнозування сезонних обсягів надходження сировини для молочних продуктів на основі нейронних мереж передбачається тривала робота за комп'ютером із застосуванням сучасного програмного забезпечення. Такий тип діяльності належить до категорії інтелектуально напруженої, що потребує дотримання комплексу вимог охорони праці, спрямованих на збереження здоров'я виконавця та забезпечення безпеки під час експлуатації комп'ютерної техніки.

Перш за все, необхідно провести перевірку справності комп'ютерного обладнання перед початком роботи. Корпус системного блоку, монітора, кабелі живлення та периферійні пристрої повинні бути без видимих пошкоджень. Усі прилади мають бути підключені через мережеві фільтри із захистом від перенапруги. Особливу увагу слід приділяти наявності заземлення, що є обов'язковим елементом безпечної експлуатації офісної техніки відповідно до чинних вимог електробезпеки.

Під час роботи за комп'ютером важливо правильно організувати робоче місце. Монітор розташовується на відстані 60–70 см від очей, при цьому верхній край екрана має знаходитися на рівні зорової осі. Клавіатура та миша повинні бути розміщені на одному рівні з ліктями користувача, аби уникати напруження м'язів зап'ястя. Робочий стіл і крісло мають відповідати вимогам ергономіки: висота регулюється індивідуально, крісло оснащується спинкою з підтримкою поперекового відділу.

Під час написання програмного коду та тестування нейронної мережі необхідно дотримуватись раціонального режиму праці: після кожної години безперервної роботи рекомендовано робити перерву тривалістю 5–10 хвилин для зменшення статичного навантаження та профілактики зорової втоми. У

період перерв бажано виконувати легку гімнастику для м'язів спини, шиї та плечового поясу, а також вправи для очей.

Слід також зважати на мікроклімат приміщення. Оптимальною вважається температура в межах 20–24°C, вологість повітря – 40–60%. Необхідно уникати протягів, перегрівання або пересушування повітря. У разі поганої вентиляції доцільно використовувати зволожувачі або періодично провітрювати приміщення. Освітлення має бути рівномірним, комбінованим, із загальним джерелом світла та локальною настільною лампою з регулюванням яскравості.

Важливо уникати перевантаження нервової системи. Високий рівень концентрації, необхідний для програмування та моделювання, зумовлює потребу в періодичному перемиканні уваги, що може бути реалізовано через зміну виду діяльності або фізичну активність. При прояві ознак перевтоми, головного болю чи зниження працездатності необхідно призупинити роботу та звернутися за консультацією до лікаря.

Усі дії, що стосуються налаштування або технічного обслуговування комп'ютерного обладнання, повинні виконуватись лише при вимкненні пристроїв з мережі. Забороняється самостійно відкривати корпус системного блоку або втручатися у внутрішні компоненти техніки без відповідної підготовки та інструктажу. У разі виникнення пожежної небезпеки слід негайно знеструмити обладнання та скористатися первинними засобами пожежогасіння.

Таким чином, дотримання зазначених інструкцій дозволяє мінімізувати професійні ризики під час розробки інтелектуальних моделей і сприяє забезпеченню комфортних та безпечних умов праці для виконавця проєкту. Рациональна організація робочого процесу, гігієна праці, дотримання правил електробезпеки та періодичні перерви є ключовими складовими збереження ефективності роботи та підтримання фізичного й психічного здоров'я фахівця.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Нами проаналізовано сучасний стан молочної галузі та проблема сезонності постачання сировини. Встановлено, що для молокопереробних підприємств, особливо тих, що працюють на постійне навантаження обладнання та мають контракти з роздрібними мережами, прогнозованість постачання сировини є достатньо важливим. З огляду на складність сезонних коливань та їх багатофакторну природу, класичні статистичні методи часто виявляються недостатніми.

Виконаний аналіз стану використання нейронних мереж для прогнозування часових рядів. Встановлено, що використання штучних нейронних мереж для прогнозування часових рядів дає змогу моделювати складні, нечітко виражені закономірності у даних, які важко описати аналітично. Саме завдяки своїм властивостям – гнучкості, здатності до узагальнення, відсутності жорстких припущень та ефективності в умовах шумових даних – ШНМ стали потужним інструментом у сучасній практиці прогнозування.

Архітектура штучної нейронної мережі (ШНМ) визначає спосіб, у який нейрони організовані в шари, та характер зв'язків між ними. Найпоширенішим типом мережі, що використовується у задачах прогнозування, є багат шаровий перцептрон (MLP), який складається з вхідного шару, одного або декількох прихованих шарів та вихідного шару.

На рисунку 1.5 зображено загальну схему організації навчального процесу ШНМ у задачі прогнозування часових рядів, де послідовно показано подачу вхідних даних, проходження сигналу через мережу та оновлення ваг у процесі зворотного поширення помилки.

Для досягнення поставленої мети вибрано технології та засоби для розробки моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів. Основою для створення моделі слугувала мова програмування Python. Для реалізації нейронної мережі було обрано

бібліотеку TensorFlow у поєднанні з високорівневим API Keras. Для попередньої обробки даних, очищення, нормалізації та формування часових вікон було використано бібліотеки NumPy та Pandas. Також використовувалася бібліотека Matplotlib (рис. 2.3) для візуалізації часових рядів, аналізу сезонних коливань і представлення результатів роботи моделі у графічній формі.

Нами було розглянуто кілька архітектур штучних нейронних мереж, зокрема багатошарові перцептрони (MLP) та рекурентні нейронні мережі (RNN). Попри переваги RNN у роботі з часовими структурами, у даному проєкті було зосереджено увагу на MLP як на більш доступному та стабільному рішенні для випадку з обмеженим обсягом історичних даних і чітко вираженою сезонністю.

У нас архітектура MLP включає один вхідний рівень, який складається з п'яти параметрів: – дата; – обсяг заготівлі молока; (oC) – температура повітря; (mm) – кількість опадів. Ці параметри використовуються як вхідні характеристики у навчанні MLP-моделі для задач прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів.

Рекурентна нейронна мережа (RNN) є різновидом штучної нейронної мережі, архітектура якої подібна до прямого (feed-forward) перцептрона, однак відрізняється тим, що передача інформації в мережі здійснюється не лише в прямому напрямку від входу до виходу, але й зворотно. Ця властивість надає мережі здатність опрацьовувати послідовності даних, зберігаючи контекст попередніх станів. RNN надає можливість враховувати контекст послідовних спостережень, що робить її придатною для задач, де важлива послідовність даних – зокрема, в обробці часових рядів, сигналів або прогнозуванні на основі історичних даних.

Для побудови моделі прогнозування сезонних обсягів надходження сировини для виробництва молочних продуктів було використано емпіричні щоденні спостереження, які охоплюють період із січня 2021 року по березень 2025 року (табл. 3.1). Усього в базі даних наявні 1535 екземплярів, що забезпечує достатню кількість спостережень для якісного навчання моделей з

урахуванням сезонних і кліматичних закономірностей. Джерелом інформації є Заболотцівська об'єднана територіальна громада Бродівського району, що має кооператив для заготівлі молока.

Отриманий графік динаміки виробництва молока свідчить про вплив сезонних температур і атмосферного тиску на надої. Таким чином, дані, що використовуються для навчання моделей прогнозування, мають чітку часову структуру, регулярну періодичність, повноту та відсутність шуму, що створює сприятливі умови для розробки як традиційних регресійних моделей, так і глибоких рекурентних мереж із урахуванням історичної динаміки.

На етапі попередньої обробки даних виконуються базові перетворення, які забезпечують відповідність структури вхідної інформації вимогам навчання нейромережевих моделей. Особлива увага приділяється форматуванню часових міток, приведенню числових ознак до єдиних одиниць вимірювання, обробці пропущених значень і очищенню набору від аномалій та шуму, які можуть спотворити навчання. Виконання цього етапу дозволило покращити якість вхідної інформації, скоротити вплив викидів і сезонно зумовлених шумів, сформувані нові інформативні ознаки, що покладені в основу побудови прогностичних моделей.

Перед подачею даних на вхід штучної нейронної мережі необхідно забезпечити їх масштабну узгодженість. Це особливо критично, коли вхідні змінні мають різні одиниці вимірювання або суттєво відрізняються за діапазоном значень. У такому випадку модель може несвідомо надавати перевагу тим ознакам, які мають більший порядок величини. Для усунення цієї проблеми застосовується масштабування числових ознак за допомогою нормалізації.

Встановлено, що спостерігається чітка сезонна закономірність – зниження надоїв у літні місяці, коли високі температури можуть викликати тепловий стрес у корів, і поступове зростання в осінньо-зимовий період. Це ще раз підтверджує доцільність включення циклічних ознак у модель, яка має враховувати повторюваність цих закономірностей з року в рік. Побудова

сезонних ознак є важливим етапом формування якісного вхідного простору для моделі.

Правильно сформовані вхідні та вихідні ознаки є основою для навчання моделей прогнозування. Використання як поточних значень, так і ковзних послідовностей забезпечує можливість як короткострокового передбачення, так і аналізу довгих часових структур, що особливо важливо в задачах моделювання сезонних явищ у тваринництві.

Виконаний аналіз даних підтверджує, що вхідні ознаки мають як фізіологічне, так і статистичне підґрунтя, а також демонструють сезонну структуру і потенційну залежність від кліматичних умов. Це створює передумови для ефективного навчання моделей прогнозування із застосуванням як класичних, так і нейромережевих підходів.

Щоб упевнитися в правильності розподілу, виводимо перші та останні дані кожної підмножини. Для наочності проводиться візуалізація розподілу спостережень по підмножинах (рис. 3.28). Встановлено, коректне розбиття вибірки забезпечує чесну оцінку якості моделі, запобігає перенавчанню та дозволяє об'єктивно порівнювати результати різних архітектур і налаштувань у процесі моделювання.

Написаний код і створена модель MLP здатна адекватно відтворювати тренди сезонних змін обсягів надоїв при збереженні достатньої точності навіть на нових даних. Це дає підстави використовувати її як базову модель для порівняння з більш складними послідовнісними мережами, які враховують часову пам'ять.

Написаний код і створена модель RNN демонструє високу чутливість до змін у динаміці ознак і точніше прогнозує короткотермінові флуктуації, що робить її доцільною для задач щоденного прогнозування з урахуванням історичного контексту. При цьому глибина вікна, функції активації, розмірність LSTM-комірок і кількість епох є параметрами, які можуть бути налаштовані в процесі оптимізації моделі.

Після навчання моделей багатошарового перцептрона (MLP) та рекурентної нейронної мережі (RNN) здійснюється порівняльний аналіз їх ефективності у задачі прогнозування сезонних обсягів надходження молочної сировини. Встановлено, що модель RNN краще справляється з великими відхиленнями та різкими коливаннями у даних, які часто спостерігаються у сезонному виробництві. Переваги RNN спостерігаються у показнику MAPE – відносної похибки. При значенні 5,42% вона показує більш точне наближення до реального рівня надоїв у відносному вираженні, на відміну від MLP, де MAPE становить 6,14%.

Нами розроблено заходи щодо організації безпечних умов праці під час створення моделі. Проведено аналіз небезпечних та шкідливих чинників, пов'язаних із роботою за комп'ютером, розроблено інструкцію з охорони праці, яка враховує ергономічні, електротехнічні та психофізіологічні ризики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Береза А. М. Основи створення інформаційних систем. Навч. посібник. 2-е вид., перероб. і доп. К.: КНЕУ, 2001. 214 с.
2. Берко А. Системи баз даних та знань. Книга 1. Організація баз даних та знань: Навч. посібник / А. Берко, О. Верес, В. Пасічник. Львів: Магнолія 2006, 2011. 456 с.
3. Боюн В. П. Інтелектуальні комп'ютерні системи сприйняття і обробки фізичної інформації: вісник Національної академії наук України. Київ, 2015. С. 82–84.
4. Вахнюк С.В. Технологія створення програмних та інтелектуальних систем: навчальний посібник / С. В. Вахнюк. Суми: ДВНЗ “УАБС НБУ”, 2011. 254 с.
5. Вергунов В. В. Інтелектуальні системи збереження даних як інструмент для підтримки оптимальної структури даних та архітектури бази даних: вісн. КНУ ім. Тараса Шевченка. Київ, 2013. С. 122–127.
6. Використання штучних нейронних мереж для прогнозування складових гібридних проектів / А. Тригуба, В. Пташник, А. Татомир, Н. Коваль, І. Кондисюк. Теорія і практика розвитку агропромислового комплексу та сільських територій: матеріали XXII Міжнар. наук.–практ. форуму, 5–7 жовт. 2021р.: у 2 т. Львів: ННБК «АТБ», 2021. Т. 2. С. 96–100.
7. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред. Гандзюка М. П. К.: Каравела 2003. 405 с.
8. Губський Ю. В., Іванов С. М. Системи штучного інтелекту. Навч. посіб. К.: Академвидав, 2010. 236с.
9. Жидецький В.Ц. Основи охорони праці: Підручник. К.: Основа, 2002. 320с.
10. Зайченко Ю. П. Основи проектування інтелектуальних систем. Київ: Слово, 2006. 352 с.

11. Ковальчук О.О., Мазуренко В.В. Проблеми та перспективи розвитку молочної промисловості України. Економіка АПК. 2023. № 3. С. 52–58.
<https://doi.org/10.37406/2310-9262-2023-3-52>
12. Любарський С.В., Шаціло П.В. Методологія вибору моделі подання знань в інтелектуальних навчальних системах: збірник наукових праць ВІТІ НТУУ «КПІ». Київ, 2010. С. 65–71.
13. Мельник Л.Г., Волянчук Н.В. Машинне навчання: підручник. К.: Ліра-К, 2018. 364с.
14. Можелянська, А. СУЧАСНІ ТЕНДЕНЦІЇ ТА ВИКЛИКИ МОЛОЧНОЇ ГАЛУЗІ В УКРАЇНІ. Bulletin of Sumy National Agrarian University, 2023. 4 (96), 8-11. <https://doi.org/10.32782/bsnau.2023.4.2>
15. Москальова В. М. Основи охорони праці. Підручник. Київ: ВД Професіонал, 2005. 666 с.
16. Нестеренко О.В., Ковтунець О.В., Фаловський О.О. Інтелектуальні системи і технології. Ввідний курс: Навч. посібник. К.: Національна академія управління, 2017. 90 с.
17. Попит-пропозиція на молочну продукцію у світі. URL: <https://milkua.info/uk/post/popit-propozicia-na-molocnu-produkciu-u-sviti>
18. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. К. : Видавничо-поліграфічний центр «Київський університет», 2014. 640 с.
19. Стан і перспективи молочної галузі України та світу. URL: <https://milkua.info/uk/post/stan-i-perspektivi-molocnoi-galuzi-ukraini-ta-svitu>
20. Tryhuba A., Boyarchuk V., Tryhuba I., Ftoma O., Padyuka R., Rudynets M., Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning, Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLLeT+DS 2020). Volume I: Main Conference. Lviv–Shatsk, Ukraine, June 2–3, 2020. pp.327–340.

21. Тригуба А. Системна модель цифрової трансформації сільських територіальних громад на основі обчислювального інтелекту. Вісник Львівського національного екологічного університету. Агроінженерні дослідження, 2022, № 26, р. 177–184.
22. Hornik, M. Stinchcombe, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, 2. P. 359-366.
23. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. *MoMLLeT+ DS*, 196–206.
24. Kumar H., Hooda B.K. Prediction of Milk Production Using Artificial Neural Network. *Current Advances in Agricultural Sciences (An International Journal)*. 2014, 6(2), 173. <https://doi.org/10.5958/2394-4471.2014.00013.6>
25. Malanchuk O., Tryhuba A., Tryhuba I., Sholudko R., Pankiv O., A Neural Network Model–based Decision Support System for Time Management in Pediatric Diabetes Care Projects. *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, 2023.
26. Malanchuk O.M., Triguba A.M., Pankiv O.V., Sholudko R.Y. Architecture of an intelligent information system for forecasting components of medical projects. *Applied aspects of information technology* 6 (4), 376–390–376–390.
27. Murugesan R., Mishra E., Krishnan A.H. Forecasting Agricultural Commodities Prices Using Deep Learning-Based Models: Basic LSTM, Bi-LSTM, Stacked LSTM, CNN LSTM, and Convolutional LSTM. *International Journal of Sustainable Agricultural Management and Informatics*. 2022, 8(3), 242. <https://doi.org/10.1504/IJSAMI.2022.125757>
28. Pandas [Електронний ресурс]. URL: <http://pandas.pydata.org/pandas-docs/stable/overview.html#license>.
29. Purohit S.K., Panigrahi S., Sethy P.K., Behera S.K. Time Series Forecasting of Price of Agricultural Products Using Hybrid Methods. *Applied Artificial Intelligence*. 2021, 35(3). <https://doi.org/10.1080/08839514.2021.1981659>
30. Python (programming language). URL: <https://bitly.su/paHJe>.

31. Rahman W., Islam R., Hasan A., Bithi N.I., Hasan Md. M., Motiur Rahman M.M. Intelligent waste management system using deep learning with IoT. *Journal of King Saud University – Computer and Information Sciences*. Volume 34, Issue 5, May 2022, Pages 2072–2087.

32. Sanzogni L., Kerr D.V. Milk Production Estimates Using Feed Forward Artificial Neural Networks. *Computers and Electronics in Agriculture*. 2001, 32(1). [https://doi.org/10.1016/S0168-1699\(01\)00151-X](https://doi.org/10.1016/S0168-1699(01)00151-X)

33. Tryhuba A., Boyarchuk V., Tryhuba I., Francik S., Rudynets M., Method and software of planning of the substantial risks in the projects of production of raw material for biofuel, *CEUR Workshop Proceedings*, 2020, 2565, pp. 116–129.

34. Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Tatomyr A., Intellectual information system for formation of portfolio projects of motor transport enterprises, *CEUR Workshop Proceedings*, 2022, 3109, pp. 44–52.

35. Tryhuba A., Kotenko V. Intelligent information system for resource planning in grain crops delivery projects on the basis of machine learning. *IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) 2023*. P. 1–4.

36. Tryhuba A., Koval N., Tryhuba I., Boiarchuk O., Application of Sarima Models in Information Systems Forecasting Seasonal Volumes of Food Raw Materials of Procurement on the Territory of Communities. *CEUR Workshop Proceedings*, 2022, 3295, pp. 64–75.

37. Tryhuba A., Padyuka R., Tymochko V., Lub P., Mathematical model for forecasting product losses in crop production projects. *CEUR Workshop Proceedings*, 2022, 3109, pp. 25–31.

38. Tryhuba A., Tryhuba I., Ftoma O., Boyarchuk O., Method of quantitative evaluation of the risk of benefits for investors of fodder-producing cooperatives. *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, 2019, 3, pp. 55–58, 8929788.

39. Tryhuba I., Hutsol T., Tryhuba A., Cieszewska A., Kovalenko N., Mudryk K., Glowacki S., Bryś A., Tulej W., Sojak M. Analysis and Quantification of the

Generation of Organic Waste by Households for Energy Production. Sustainability 2023 , 15 (22), 15922; <https://doi.org/10.3390/su152215922>

40. Tryhuba I., Tryhuba A., Hutsol T., Cieszewska A., Andrushkiv O., Glowacki S., Bryś A., Slobodian S., Tulej W., Sojak M. Prediction of Biogas Production Volumes from Household Organic Waste Based on Machine Learning. Energies 2024, 17(7), 1786; <https://doi.org/10.3390/en17071786>

41. Tryhuba I., Tryhuba A., Hutsol T., Lopushniak V., Cieszewska A., Andrushkiv O., Barabasz W., Pikulicka A., Kowalczyk Z., Vasyuk V. European Green Deal: Justification of the Relationships between the Functional Indicators of Bioenergy Production Systems Using Organic Residential Waste Based on the Analysis of the State of Theory and Practice. Energies. Енергія 2024 , 17 (6), 1461; <https://doi.org/10.3390/en17061461>

Додатки

Додаток А

Фрагмент програмного коду створення модуля архітектури моделі MLP (багатошаровий перцептрон)

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

# Кількість вхідних ознак
input_dim = X.shape[1]

# Побудова MLP-моделі з використанням Input
model_mlp = Sequential([
    Input(shape=(input_dim,)),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1, activation='linear') # Вихідний шар для регресії
])

# Компіляція моделі
model_mlp.compile(optimizer=Adam(learning_rate=0.001), loss='mse',
metrics=['mae'])

# Припускаємо, що X і y вже створено (після попередньої обробки)
# Наприклад:
# features = ["Temperature(oC)", "Pressure(mm)", "Month_sin", "Month_cos",
# "Weekday_sin", "Weekday_cos"]
# X = df[features]
# y = df["MilkProduction_t+1"]

# Розбиття з урахуванням часу
N = len(X)
train_size = int(0.7 * N)
val_size = int(0.15 * N)

X_train = X.iloc[:train_size]
y_train = y.iloc[:train_size]

X_val = X.iloc[train_size:train_size + val_size]
y_val = y.iloc[train_size:train_size + val_size]

X_test = X.iloc[train_size + val_size:]
y_test = y.iloc[train_size + val_size:]

history = model_mlp.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    verbose=1
)

plt.figure(figsize=(10,5))
plt.plot(history.history['loss'], label='Train MSE')
plt.plot(history.history['val_loss'], label='Validation MSE')
plt.title("Динаміка зміни функції втрат під час навчання MLP")
plt.xlabel("Епоха")
plt.ylabel("Середньоквадратична помилка (MSE)")

```

```
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Фрагмент програмного коду створення архітектури моделі RNN (рекурентна нейронна мережа)¶

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

def create_sliding_windows(X, y, window_size=7):
    Xs, ys = [], []
    for i in range(len(X) - window_size):
        v = X.iloc[i:i+window_size].values
        Xs.append(v)
        ys.append(y.iloc[i:i+window_size].values)
    return np.array(Xs), np.array(ys)

# Створення RNN-вибірки
X_seq, y_seq = create_sliding_windows(X, y, window_size=7)

N = len(X_seq)
train_size = int(0.7 * N)
val_size = int(0.15 * N)

X_train_rnn = X_seq[:train_size]
y_train_rnn = y_seq[:train_size]

X_val_rnn = X_seq[train_size:train_size + val_size]
y_val_rnn = y_seq[train_size:train_size + val_size]

X_test_rnn = X_seq[train_size + val_size:]
y_test_rnn = y_seq[train_size + val_size:]

# Побудова моделі з LSTM
model_rnn = Sequential([
    LSTM(64, activation='tanh', input_shape=(X_seq.shape[1], X_seq.shape[2])),
    Dense(32, activation='relu'),
    Dense(1, activation='linear')
])

# Компіляція моделі
model_rnn.compile(optimizer=Adam(learning_rate=0.001), loss='mse',
metrics=['mae'])

history_rnn = model_rnn.fit(
    X_train_rnn, y_train_rnn,
    validation_data=(X_val_rnn, y_val_rnn),
    epochs=100,
    batch_size=32,
    verbose=1
)

plt.figure(figsize=(10, 5))
```

```
plt.plot(history_rnn.history['loss'], label='Train MSE')
plt.plot(history_rnn.history['val_loss'], label='Validation MSE')
plt.title("Динаміка помилки моделі RNN")
plt.xlabel("Епоха")
plt.ylabel("MSE")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```