

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: «Прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей»

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та технології»

(шифр і назва)

Гавришків Олексій Володимирович

(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Шарибура А.О.

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

«____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Гавришківу Олексію Володимировичу

1. Тема роботи: «Прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 28.04.2023 року № 133/к-с.

2. Строк подання студентом роботи 10.12.2024 р.

3. Вихідні дані до роботи: дані прогнозування конкурсного балу соціальних проєктів розвитку громад; методи використання технологій обчислювального інтелекту.

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

Вступ.

Аналіз стану прогнозування конкурсного балу соціальних проєктів та використання нейромереж.

Розробка концепції нейромережової моделі для прогнозування конкурсного балу та підготовка даних.

Результати створення нейромережових моделей для прогнозування конкурсного балу соціальних проєктів розвитку громад.

Охорона праці та безпека у надзвичайних ситуаціях.

Визначення показників економічної ефективності.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): аналіз стану прогнозування конкурсного балу соціальних проєктів та використання нейромереж; розробка концепції нейромережевої моделі для прогнозування конкурсного балу та підготовка даних; результати створення нейромережевих моделей для прогнозування конкурсного балу соціальних проєктів розвитку громад; економічна ефективність.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Тригуба А.М., зав. кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва</i>		

7. Дата видачі завдання

28 квітня 2023 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	<i>28.04-20.05.23</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>21.05-14.08.23</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.08.23-10.07.24</i>	
4.	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	<i>11.07-31.08.24</i>	
5.	<i>Оцінення ефективності запропонованої системи</i>	<i>01.09-31.10.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>01-30.11.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>01-10.12.24</i>	

Студент _____ Гавришків О.В.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 004.8: 519.86: 316.42

Прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей.

Гавришків О.В. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 85 с. текст. част., 18 рис., 5 табл., 15 арк. ілюстраційного матеріалу, 64 джерел.

Подано особливості та доцільність прогнозування конкурсного балу соціальних проєктів розвитку громад. Проаналізовано використання машинного навчання та нейронних мереж у соціальних сферах. Виконано аналіз наукових праць у напрямі дослідження. Подано напрямки сучасних досліджень. Сформульовано завдання кваліфікаційної роботи.

Розроблена концепції побудови нейромережових моделей. Здійснено вибір та підготовку даних. Наведено результати аналізу взаємозв'язків між даними. Наведено показники оцінки точності моделей та використання алгоритмів оптимізації для покращення моделей. Обґрунтована архітектура нейромережових моделей (Feedforward Neural Network (FNN), Recurrent Neural Network (RNN) та Convolutional Neural Network (CNN)) для прогнозування конкурсного балу соціальних проєктів. Побудовано нейромережові моделі для прогнозування конкурсного балу соціальних проєктів розвитку громад. Виконано порівняння результатів та аналіз точності. Здійснено оптимізацію моделі Convolutional Neural Network (CNN).

Виконано аналіз небезпечних чинників та розробка заходів із покращення умов праці. Визначено показники економічної ефективності від прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей.

Ключові слова: прогнозування, конкурсний бал, соціальні проєкти, розвиток громад, нейронні мережі, машинне навчання, інтелектуальні моделі.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ ПРОЄКТІВ ТА ВИКОРИСТАННЯ НЕЙРОМЕРЕЖ.....	9
1.1. Особливості та доцільність прогнозування конкурсного балу соціальних проєктів розвитку громад.....	9
1.2. Машинне навчання та нейронні мережі	11
1.3. Огляд нейронних мереж	13
1.4. Застосування машинного навчання та нейронних мереж у соціальних сферах	15
1.5. Використання машинного навчання у соціальних системах.....	18
1.6. Аналіз наукових праць у напрямі дослідження	20
1.7. Напрямки сучасних досліджень	25
1.8. Завдання кваліфікаційної роботи	26
РОЗДІЛ 2. РОЗРОБКА КОНЦЕПЦІЇ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ ТА ПІДГОТОВКА ДАНИХ...	28
2.1. Розробка концепції побудови нейромережових моделей	28
2.2. Вибір та підготовка даних	31
2.3. Результати аналізу взаємозв'язків між даними.....	34
2.4. Показники оцінки точності моделей.....	38
2.5. Використання алгоритмів оптимізації для покращення моделей.....	40
РОЗДІЛ 3. РЕЗУЛЬТАТИ СТВОРЕННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ ПРОЄКТІВ РОЗВИТКУ ГРОМАД.....	42
3.1. Архітектури нейромережових моделей для прогнозування конкурсного балу соціальних проєктів.....	42
3.1.1. Feedforward Neural Network (FNN).....	42
3.1.2. Recurrent Neural Network (RNN)	43
3.1.3. Convolutional Neural Network (CNN).....	44

3.2. Побудова неймережевих моделей для прогнозування конкурсного балу соціальних проєктів розвитку громад	45
3.2.1. Побудова Feedforward Neural Network (FNN)	45
3.2.2. Побудова Recurrent Neural Network (RNN)	47
3.2.3. Побудова Convolutional Neural Network (CNN)	49
3.3. Порівняння результатів та аналіз точності	52
3.4. Оптимізація моделі Convolutional Neural Network (CNN)	54

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпечних чинників та розробка заходів із покращення умов праці	57
4.2. Розробка логічно-імітаційної моделі травматизму під час монтажу інформаційної системи	58
4.3. Розробка заходів щодо безпеки у надзвичайних ситуаціях	63

РОЗДІЛ 5. ВИЗНАЧЕННЯ ПОКАЗНИКІВ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ ПРОЄКТІВ РОЗВИТКУ ГРОМАД НА ОСНОВІ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ.....

ВИСНОВКИ І ПРОПОЗИЦІЇ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	81
Додаток А.1. Код створення моделей штучних нейронних мереж.....	82
Додаток А.2. Код оптимізації архітектури CNN.....	85

ВСТУП

В сучасних умовах розвитку громади зростає значення соціальних проєктів, які спрямовані на вдосконалення інфраструктури, умов життя, освіти, охорони здоров'я та інших сфер, важливих для населення. Однією з ключових проблем процесу реалізації таких ініціатив є обмеженість фінансових ресурсів, що вимагає організації проведення конкурсного відбору для визначення проєктів, які мають найбільший потенціал впливу та користі [54]. У таких умовах вимагається потреба в розробці методів, що не можна підвищити об'єктивність оцінювання конкурсних заявок, враховуючи різноманітні критерії, такі як економічна ефективність, соціальна значущість, інноваційність тощо.

Традиційні методи оцінювання проєктів часто базуються на експертній оцінці, яка, хоча і є важливою складовою процедурою, не завжди дозволяє врахувати всі можливості факторів і зберегти об'єктивність. Завдяки розвитку технологій машинного навчання, зокрема нейронних мереж, з'явилися нові можливості для автоматизації процесу оцінювання конкурсних балів на основі аналізу великої кількості даних [43]. Нейромережеві моделі не дозволяють аналізувати чисельні параметри проєктів, виділяти їх взаємозв'язки та залежність, що досягла точності та об'єктивності конкурсної оцінки.

Використання нейромережевих моделей у прогнозуванні проєктів конкурсного балу соціальних пропозицій може значно полегшує процес відбору, зменшує суб'єктивний вплив і забезпечує справедливий розподіл фінансування між найбільш перспективними ініціативами. Розробка такої моделі є актуальним напрямком досліджень, вона здатна покращити процеси прийняття рішень у галузі соціального розвитку громад.

У кваліфікаційній роботі створено нейромережеву модель для прогнозування конкурсного балу соціальних проєктів розвитку громад, яка дозволяє підвищити об'єктивність та ефективність процесу оцінювання.

Об'єкт дослідження – процес конкурсного відбору соціальних проєктів розвитку громад.

Предмет дослідження – методи та моделі застосування нейронних мереж для прогнозування конкурсного балу соціальних проєктів.

Засоби дослідження – бібліотеки машинного навчання TensorFlow, Keras для реалізації нейромережевої моделі; інструменти для обробки та аналізу даних Python, Pandas, NumPy.

РОЗДІЛ 1.

АНАЛІЗ СТАНУ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ ПРОЄКТІВ ТА ВИКОРИСТАННЯ НЕЙРОМЕРЕЖ

1.1. Особливості та доцільність прогнозування конкурсного балу соціальних проєктів розвитку громад

Прогнозування конкурсного балу соціальних проєктів є завданням, яке спрямоване на об'єктивізацію процесу відбору проєктів для реалізації в умовах обмеженого фінансування. Соціальні проєкти мають специфічну структуру та характеристики, що відрізняє їх від комерційних ініціатив і вимагають особливого підходу до оцінювання. У таких проєктах ключовими показниками є не лише фінансова ефективність, а й соціальна значущість, вплив на громаду, інноваційність і стійкість [18]. Таким чином, оцінка кожного проєкту вимагає аналізу численних факторів, які можуть бути складними для одночасного врахування експертами. Саме тут прогнозування за допомогою нейронних мереж забезпечує ефективний підхід.

Особливості прогнозування конкурсного балу соціальних проєктів представлено на рис. 1.1.



Рисунок 1.1 – Особливості прогнозування конкурсного балу соціальних проєктів

Соціальні проекти оцінюються за найкращими критеріями, що включають вплив на громадськість, очікувані соціальні вигоди, рівень залучення громади, інноваційність, екологічну стійкість тощо. Нейронні мережі не зможуть одночасно отримати всі ці критерії, виявляючи приховані взаємозв'язки між ними та надаючи точнішу оцінку.

Традиційно конкурсні оцінки обґрунтовуються на експертному аналізі, що може бути суб'єктивним. Нейромережеві моделі допомагають зменшити суб'єктивний вплив на процес автоматизації, забезпечуючи стандартизовану оцінку проектів.

Соціальні проекти відрізняються великою варіативністю, і навіть незначні зміни в параметрах можуть значно вплинути на загальну оцінку. Нейронні мережі здатні адаптуватися до складних, багатьох факторних даних та навчатися на основі попередніх оцінок, що досягли точності прогнозів.

З часом накопичується велика кількість даних про соціальні об'єкти, оцінки та результати їх реалізації [16]. Нейронні мережі можуть обробляти такі обсяги даних, використовуючи їх для навчання моделей, що підвищує точність прогнозування.

Складові, що зумовлюють доцільність прогнозування конкурсного балу представлено на рис. 1.2.



Рисунок 1.2 – Складові, що зумовлюють доцільність прогнозування конкурсного балу

Нейромережеві моделі забезпечують автоматизовану обробку даних, що знижує суб'єктивність людського фактору та забезпечує об'єктивне розміщення проєктів за конкурсними балами.

Прогнозування конкурсного балу за допомогою моделей машинного навчання дозволило значно прискорити процес оцінювання, що особливо актуально в умовах обмеженого часу та великої кількості заявок.

Оскільки фінансові ресурси на реалізацію соціальних проєктів завжди обмежені, прогнозування конкурсного балу дозволило забезпечити більш точний та справедливий розподіл коштів між об'єктами, які мають найбільший соціальний вплив.

Використання нейромережевих моделей у прогнозуванні конкурсного балу завершилося прозорістю процесу відбору, хоча оцінки проєкту відповідали до однакових стандартів, що сприяє довірі серед учасників конкурсу.

Прогнозування на основі нейронних мереж дозволяє врахувати історичні дані та ефективно аналізувати динамічні оцінки, забезпечуючи можливість для подальшого вдосконалення процесу відбору.

Таким чином, прогнозування конкурсного балу соціальних проєктів розвитку громади на основі нейромережевих моделей є доцільним і перспективним напрямком, що дозволяє забезпечити об'єктивний, швидкий та прозорий процес відбору проєктів для оптимального використання ресурсів громади.

1.2. Машинне навчання та нейронні мережі

Штучний інтелект (AI) – це підгалузь інформатики, яка включає в себе широкий спектр обчислювальних операцій, від алгоритмічного виробництва до машинного навчання та методів глибокого навчання [41]. Традиційні методи вирішення проблем штучного інтелекту базуються на правилах «якщо-тоді»,

тоді як машинне навчання та глибоке навчання прагнуть ітеративно розвивати розуміння набору даних без явного кодування будь-яких правил [51; 48]. Це дозволяє обчислювальній системі, на якій вони реалізовані, автоматично здобувати знання та розробляти прогнози, починаючи з набору вхідних даних, коригуючи параметри шляхом оптимізації стандарту продуктивності, визначеного для даних, і зменшуючи рівень помилок на кожному етапі процесу навчання [33].

Іншими словами, метою машинного навчання є створення програмного забезпечення, яке адаптується та навчається самостійно, тобто без попередньо запрограмованої системи, що диктує його поведінку. Завдяки тому, що навчальні дані використовуються як приклади, алгоритми можуть вчитися на своїх помилках. Отже, обсяг, який вивчає модель, залежить від якості та кількості прикладу інформації, якій вона була піддана [29].

Машинне навчання надає різноманітні математичні інструменти для вирішення широкого кола проблем. Штучні нейронні мережі, які навчені вирішувати конкретне завдання, на даний момент є найбільш поширеним і необхідним інструментом. Нейрони організовані в шари та точно з'єднані, щоб утворити мережу. Як було зазначено раніше, нейронна мережа вважається «глибокою», якщо кількість шарів є значною. Підхід глибокого навчання намагається математично змоделювати, як людський мозок обробляє дані зору та слуху: стимули вух і очей, що проходять через людський мозок, розбиваються на прості концепції та поступово реконструюються у все більш складні та абстрактні візуалізації [49].

Подібним чином, у глибокій нейронній мережі візаж представлений як масив значень пікселів. Початковий шар може легко визначити ребра з різною орієнтацією. Наступні шари поєднують ці елементи для створення закруглених кутів і розширених контурів. Виявляючи конкретні групи контурів і кутів, наступні шари можуть виявляти цілі частини конкретних об'єктів. Потім вони поєднуються з додатковими шарами обробки, щоб ми могли представити обличчя, які ми хочемо вивчити [47; 44].

Машинне навчання – це підхід до розробки штучного інтелекту на основі даних. Це підмножина штучного інтелекту з багатьма активами та використовує методи прогнозу статистики [23]. Він був створений у 1940-х роках, але лише нещодавно стало можливим включити його в щоденну рутину. Загалом, алгоритми машинного навчання використовують як неконтрольовані, так і контрольовані методи навчання. Навчання без контролю не потребує зворотного зв'язку з людиною, тоді як навчання під контролем вимагає [62]. Найважливішими перевагами машинного навчання є його методологія та навчання з підкріпленням. Методологія машинного навчання передбачає навчання алгоритму для розпізнавання шаблонів у нових наборах даних [24]. Навчання з підкріпленням – це підполе машинного навчання, у якому інтелектуальна система отримує знання методом проб і помилок, отримуючи стимули або покарання за свої дії.

1.3. Огляд нейронних мереж

Нейронна мережа – це модель, яка імітує роботу нервової системи людського мозку шляхом моделювання та з'єднання нейронів [45], основних одиниць людського мозку, і створення штучної системи з інтелектуальними функціями обробки інформації, такими як розпізнавання образів, пам'ять, асоціації та навчання [64]. Важливою характеристикою нейронної мережі є її здатність навчатися в навколишньому середовищі та зберігати результати свого навчання у своїх синаптичних зв'язках. Навчання нейронної мережі – це процес. Під впливом навколишнього середовища в мережу подається послідовність шаблонів зразків, а вагова матриця кожного рівня мережі регулюється відповідно до набору правил. Процес навчання завершується, коли вага кожного шару мережі збігається з певним значенням. Нейронна мережа – це ациклічний граф, що складається з взаємопов'язаних нейронів. Вихідні дані попереднього шару нейронів функціонують як вхідні дані для наступного шару

нейронів, які зазвичай розташовані регулярно та складаються з кількох нейронів у шарах з'єднань. Типова структура нейронної мережі називається рівнем повного з'єднання [59]. Нейрони в одному шарі не з'єднані.

Соціальні медіа все більше залежать від нейронних мереж. Вони дають змогу отримувати персоналізовані рекомендації щодо вмісту на основі вподобань користувачів, аналізу настроїв, розпізнавання зображень і відео, обробки природної мови (NLP), виявлення шахрайства та безпеки, а також NLP зображень і відео. Платформи соціальних медіа можуть використовувати нейронні мережі для покращення залучення користувачів [38], курування вмісту [32] та безпеки платформи [28]. Однак є наслідки, пов'язані з використанням нейронних мереж у соціальних мережах. Алгоритмічне упередження може сприяти виникненню дискримінаційних рекомендацій і практик щодо вмісту. Оскільки платформам соціальних мереж необхідно керувати та захищати конфіденційну інформацію користувачів, конфіденційність і безпека даних є проблемою. Персоналізація може призвести до бульбашок фільтрів і поляризації, а дезінформація та неправдиві новини можуть підірвати довіру до інформації, яка поширюється на цих платформах.

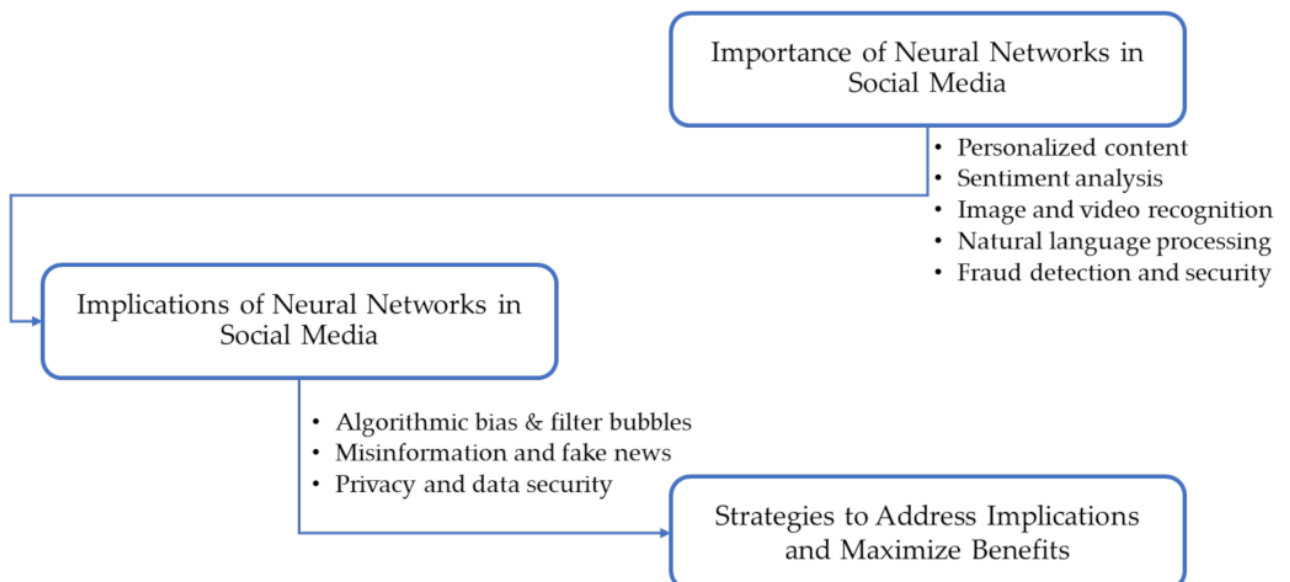


Рисунок 1.3 – Розкриття потужності нейронних мереж у соціальних мережах.

Щоб усунути та пом'якшити помилки алгоритмів, платформи соціальних мереж мають запровадити надійні процеси збору даних, різноманітні навчальні набори та регулярні аудити. Щоб вирішити проблеми конфіденційності та безпеки даних, потрібні відповідні механізми анонімізації даних, шифрування та згоди. Платформам необхідно встановити баланс між персоналізацією та доступом до різних точок зору. Щоб зменшити поширення дезінформації, необхідні постійний моніторинг, механізми перевірки фактів і системи звітності користувачів. Таким чином, платформи соціальних мереж можуть максимізувати переваги нейронних мереж, мінімізуючи їхні потенційні наслідки. Рисунок 1.3 ілюструє значення нейронних мереж у соціальних медіа, висвітлюючи їх різноманітні застосування та наслідки.

1.4. Застосування машинного навчання та нейронних мереж у соціальних сферах

Машинне навчання та нейронні мережі зробили революцію в кількох дисциплінах, зокрема в соціальних сферах. У сфері аналізу соціальних медіа методи машинного навчання полегшують вилучення цінних ідей із величезних масивів даних, таких як аналіз настроїв [19], визначення тенденцій і виявлення фейкових новин. Крім того, системи рекомендацій на основі нейронної мережі пропонують персоналізовані пропозиції щодо вмісту на основі поведінки користувача, тим самим покращуючи взаємодію з користувачем.

Розуміння та аналіз повідомлень у соціальних мережах, коментарів та відгуків неможливі без НЛП. Використовуючи нейронні мережі, такі як рекурентні нейронні мережі або трансформатори, алгоритми машинного навчання дозволяють виконувати такі завдання, як аналіз настроїв, тематичне моделювання та класифікація тексту. Розшифровуючи значення текстових даних, НЛП дозволяє більш глибоко зрозуміти настрої та вподобання користувачів.

Так само машинне навчання та нейронні мережі незамінні для аналізу соціальних мереж. Ці методи можуть аналізувати складні структури соціальних мереж, ідентифікувати впливових користувачів, виявляти спільноти та передбачати індивідуальні стосунки. Графові нейронні мережі (GNN) особливо ефективні в моделюванні та осягненні даних соціальних мереж, таким чином виявляючи цінну інформацію та зв'язки.

Застосування машинного навчання поширюється не тільки на соціальні медіа, але й на соціальне забезпечення та гуманітарну діяльність. Використовуючи прогностичні моделі та нейронні мережі, машинне навчання підтримує реагування на катастрофи, ініціативи в галузі охорони здоров'я та розподіл ресурсів гуманітарної допомоги.

Таблиця 1.1 – Застосування машинного навчання та нейронних мереж у соціальних сферах.

Напрямок	Додатки
Аналіз соціальних мереж	Аналіз настроїв. Виявлення тенденцій. Виявлення фейкових новин.
Рекомендаційні системи	Персоналізовані пропозиції вмісту на основі поведінки користувачів.
НЛП	Аналіз настроїв. Моделювання теми. Класифікація тексту.
Аналіз соціальних мереж	Ідентифікація впливового користувача. Виявлення спільноти. Прогнозування стосунків.
Соціальне благо та гуманітаризм	Реагування на стихійні лиха. Ініціативи в галузі охорони здоров'я. Виділення ресурсів на гуманітарну допомогу.
Інтернет-реклама	Цільова реклама на основі вподобань і поведінки користувачів.
Персоналізована освіта	Адаптивний навчальний контент на основі індивідуальних стилів навчання.
Виявлення шахрайства	Виявлення шахрайських дій, таких як шахрайство з кредитними картками та онлайн-шахрайство.
Кібербезпека	Аналіз мережевого трафіку для виявлення аномалій і ідентифікації кіберзагроз.
Аналіз психічного здоров'я	Виявлення осіб із ризиком психічних проблем за допомогою аналізу соціальних мереж.

Ці технології можуть передбачати спалахи захворювань, визначати вразливі зони під час стихійних лих та оптимізувати зусилля з надання допомоги, зрештою рятуючи життя та зменшуючи страждання. Застосування машинного навчання та нейронних мереж у різних соціальних дисциплінах узагальнено в таблиці 1.1.

Різноманітні алгоритми машинного навчання та архітектури нейронних мереж використовувалися для аналізу соціальних мереж і систем рекомендацій вмісту, кожен із яких має свої сильні сторони та обмеження. Традиційні алгоритми, такі як логістична регресія та дерева рішень, пропонують інтерпретацію та простоту, що робить їх придатними для розуміння взаємозв'язків і створення явних прогнозів на основі функцій. Однак їхня ефективність може бути обмеженою, якщо вони представлені разом зі складними даними або просторами функцій із великими розмірами. Крім того, архітектури глибокого навчання, такі як згорткові нейронні мережі та рекурентні нейронні мережі, чудово вловлюють складні шаблони та тимчасові залежності, що робить їх корисними для таких завдань, як аналіз настроїв і моделювання послідовності. Однак для досягнення оптимальної продуктивності ці архітектури часто вимагають значних обчислювальних ресурсів і великої кількості позначених даних. Крім того, їхня природа чорного ящика перешкоджає інтерпретації, що є життєво важливим для певних застосувань, таких як пояснення рекомендацій або виявлення упереджених шаблонів. Гібридні підходи, які поєднують сильні сторони різних алгоритмів і архітектур, таких як ансамблеві методи та гібридні моделі глибокого навчання, стали багатообіцяючими рішеннями, забезпечуючи покращену точність, інтерпретацію та масштабованість аналізу соціальних мереж і систем рекомендацій щодо контенту. Оптимальна стратегія визначається конкретною місією, наявними даними, потребами в інтерпретації та обчислювальними ресурсами.

1.5. Використання машинного навчання у соціальних системах

Нейронні мережі та алгоритми машинного навчання революціонізували різні аспекти нашого життя, які пропонують величезний потенціал і можливості. Однак широке впровадження цих технологій викликає важливі соціальні проблеми, які необхідно вирішити, щоб забезпечити їх відповідальний розвиток і впровадження.

Потенціал для упередженого прийняття рішень є основною проблемою для нейронних мереж і моделей машинного навчання [35]. Коли ці алгоритми вивчають упереджені або упереджені навчальні дані, вони можуть увічнити та посилити існуючі упередження, що призведе до дискримінації або несправедливого ставлення до певних осіб або груп. Особливо це стосується таких сфер, як зайнятість, кредитування та кримінальне правосуддя.

Щоб зменшити упередженість, важливо збирати різноманітні та репрезентативні навчальні дані, які точно відображають різноманітність населення. Регулярні перевірки моделі, методи інтерпретації та метрики справедливості можуть допомогти у виявленні та усуненні упередженості в алгоритмічному прийнятті рішень. Більше того, залучення до процесу розробки міждисциплінарних команд, у тому числі фахівців з етики, соціальних науковців та експертів у галузі, може надати цінну інформацію для зменшення упередженості та сприяння справедливості.

Нейронні мережі та алгоритми машинного навчання покладаються на величезну кількість даних для підвищення продуктивності та навчання. Однак ця залежність викликає проблеми з конфіденційністю та безпекою. Особиста інформація, зібрана під час збору та обробки даних, може бути сприйнятливою до несанкціонованого доступу, неправильного використання або вторгнення, що може призвести до порушення конфіденційності або викрадення особистих даних.

Щоб захистити конфіденційність, розробники повинні дотримуватися суворих правил захисту даних, таких як анонімізація та шифрування, а також

відповідати чинним нормам конфіденційності. Прозорі політики використання даних, механізми згоди користувачів і стратегії мінімізації даних можуть дозволити людям приймати обґрунтовані рішення щодо своїх даних. Крім того, забезпечення безпечної інфраструктури, проведення регулярних оцінок вразливості та встановлення надійного контролю доступу є важливими для захисту конфіденційних даних.

Може бути важко зрозуміти внутрішню роботу нейронних мереж і алгоритмів машинного навчання через їх складність, що викликає занепокоєння щодо підзвітності та прозорості алгоритмів [60]. У сферах із значним впливом, таких як автономні транспортні засоби чи охорона здоров'я, важливо розуміти, як і чому приймається рішення.

Сприяння прозорості вимагає розробки інтерпретованих методів машинного навчання, надання обґрунтувань для рішень і встановлення чітких меж відповідальності між розробниками, користувачами та регуляторними органами. Щоб забезпечити надійність алгоритмів, вони повинні пройти суворе тестування та валідацію, а також мають бути реалізовані механізми аудиту та оскарження алгоритмічних рішень.

У міру прогресу технологій автоматизації, що керуються нейронними мережами та машинним навчанням, з'являються побоювання переміщення робочих місць і збільшення розриву в кваліфікації [36]. Незважаючи на те, що ці технології можуть оптимізувати процеси та підвищити ефективність, вони також можуть замінити певні завдання, які зараз виконують люди, що призведе до безробіття та соціально-економічних проблем, особливо для тих, хто працює на низькокваліфікованих або рутинних роботах.

Щоб вирішити ці проблеми, робочу силу необхідно перекваліфікувати та переосвітити, щоб відповідати мінливим вимогам ринку праці. Уряди, навчальні заклади та підприємства повинні співпрацювати, щоб забезпечити навчальні програми та ресурси, які озброюють людей навичками, необхідними для нових сфер. Крім того, дослідження нових форм організації роботи, таких як розподіл робочих місць або скорочений робочий день, може допомогти у

розподілі доступної роботи та пом'якшити наслідки автоматизації. Взаємозв'язок між соціальними науками, машинним навчанням і нейронними мережами зображено на рисунку 1.4.

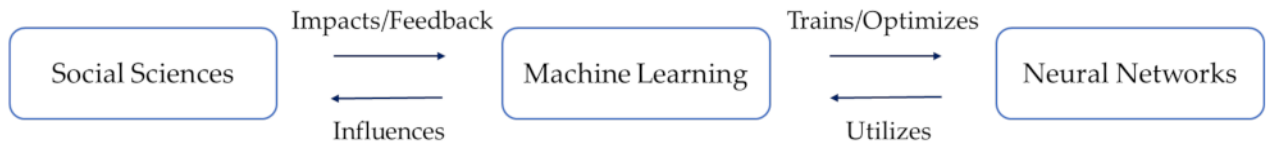


Рисунок 1.4 – Ітеративний зв'язок ідей і досягнень машинного навчання та нейронних мереж у соціальних науках

Соціальні науки, до яких належать соціологія та психологія, впливають на розвиток машинного навчання та нейронних мереж шляхом визначення дослідницьких питань і збору відповідних даних. Нейронні мережі, тип моделі машинного навчання, розуміють складні шаблони з даних. Машинне навчання використовує дані соціальних наук для навчання та оптимізації моделей. На соціальні науки впливають результати та ідеї, створені машинним навчанням і нейронними мережами, які формують теорії, політику та майбутні дослідження. Ця ітераційна процедура сприяє прогресу в соціальних науках, машинному навчанні та нейронних мережах.

1.6. Аналіз наукових праць у напрямі дослідження

Для вибору найбільш релевантних статей було використано двоетапний підхід. Назви та тези були перевірені в першому раунді на основі заздалегідь визначених критеріїв включення та виключення. Повнотекстові публікації були отримані та оцінені для остаточного включення на другому етапі. Додаткові джерела було знайдено шляхом перегляду відповідних посилань у вибраних статтях. Нарешті, було обрано 26 статей із загальної кількості 955 (рис. 1.5).

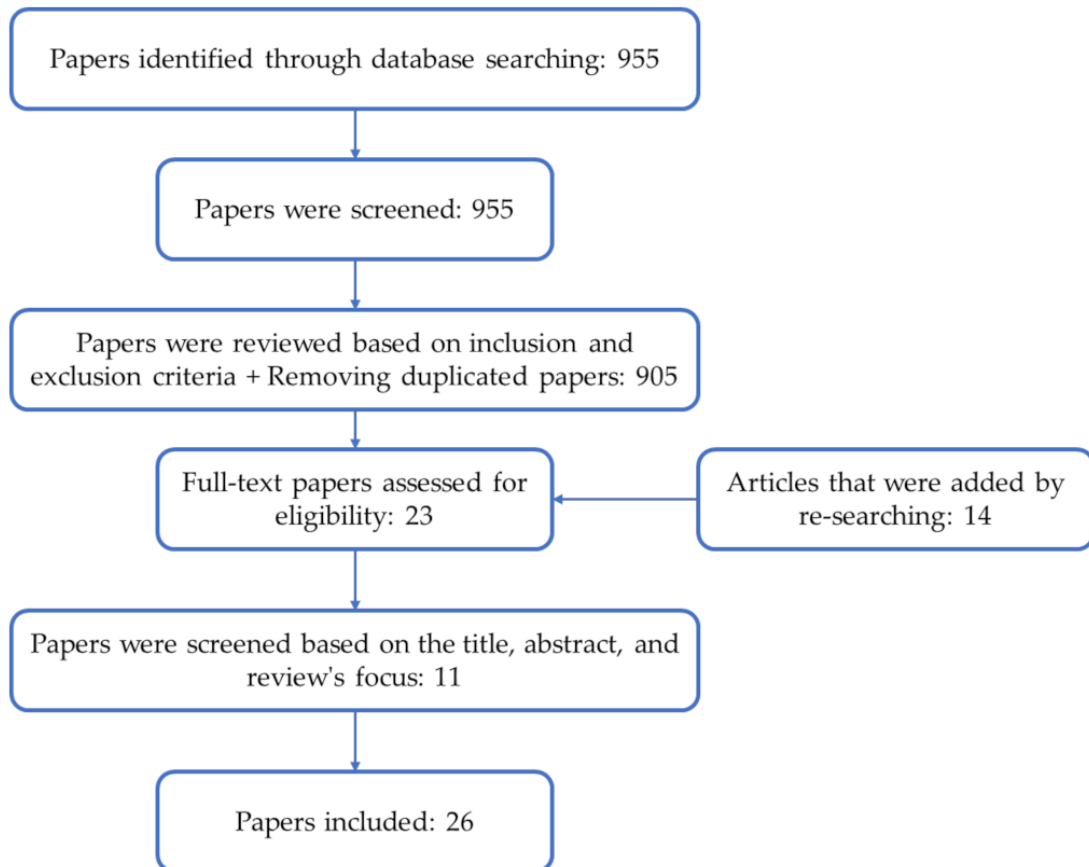


Рисунок 1.5 – Процес відбору статей

Щоб отримати важливу інформацію з вибраних статей, було виконано вилучення даних. Автор(и), рік публікації, методологія дослідження, важливі висновки та наслідки, пов'язані з соціальними проблемами машинного навчання та нейронних мереж, були розглянуті.

Висновки вибраних статей були узагальнені та структуровані відповідно до питання дослідження та мети огляду. Щоб забезпечити повне розуміння соціальних наслідків, пов'язаних із машинним навчанням і нейронними мережами, було розкрито загальні теми, шаблони та ідеї.

З 955 статей було обрано 26. Машинне навчання та нейронні мережі – це дві теми ШІ, які викликали значний інтерес в останні роки через їхній потенціал для автоматизації прийняття рішень і підвищення точності в різноманітних програмах. Машинне навчання – це процес навчання комп'ютерів навчанню на основі даних, а нейронні мережі – це форма алгоритму машинного навчання, натхненного структурою людського мозку. Машинне навчання та нейронні мережі зробили глибокий вплив на зниження

витрат, покращення взаємодії з користувачами та вдосконалення процесів прийняття рішень. Використовуючи великі обсяги даних і складні алгоритми, методи машинного навчання дозволяють компаніям автоматизувати процеси, оптимізувати розподіл ресурсів і визначати можливості економії коштів. Крім того, нейронні мережі революціонізували залучення користувачів, створюючи персоналізовані рекомендації, цільову рекламу та інтерактивні чат-боти, покращуючи таким чином загальну взаємодію з користувачем. Крім того, ці технології дозволяють приймати рішення на основі даних, надаючи організаціям цінну інформацію, прогностичну аналітику та дієві рекомендації, які сприяють більш обґрунтованому та ефективному вибору, що веде до покращення результатів і конкурентних переваг у сучасному швидкоплинному світі, що керується даними.

Відповідно до включених досліджень, машинне навчання та нейронні мережі все частіше використовуються для вирішення складних проблем і прийняття більш точних та ефективних рішень. Ці технології можуть автоматизувати роботу, яку виконує людина, наприклад ідентифікацію зображень, мовний переклад і навіть водіння автомобіля. Згідно з кількома представленими документами, машинне навчання та нейронні мережі також використовуються для підвищення безпеки та безпеки різних систем. Дослідження Karayığit et al. [39] розглядав використання машинного навчання для виявлення образливих коментарів в Instagram, тоді як інший розглядав використання нейронних мереж для створення системи виявлення вторгнень для хмарних середовищ.

Інші дослідження стверджують, що машинне навчання та нейронні мережі можуть допомогти покращити результати охорони здоров'я. Дослідження Комацу та ін. [42] охоплювало використання машинного навчання для прогнозування успіху психіатричних ліків, тоді як інше дослідження [63] досліджувало використання нейронних мереж для прогнозування ризику захворювання та підвищення безпеки пацієнтів. Однак реалізація машинного навчання та нейронних мереж не позбавлена труднощів. Пісня та ін. [50]

досліджував необхідність кількісної оцінки невизначеності в моделях машинного навчання, тоді як Ялур [61] досліджував тему «природності» в машинному навчанні та RNN. Ці назви означають, що, хоча машинне навчання та нейронні мережі мають потенціал змінити багато дисциплін, необхідно враховувати ключові міркування та обмеження.

Підсумовуючи, включені дослідження демонструють, як машинне навчання та нейронні мережі можна використовувати для покращення прийняття рішень, безпеки, результатів охорони здоров'я та інших елементів сучасного життя. Однак не менш важливо знати про обмеження цих технологій, а також ретельно досліджувати їх використання в різноманітних контекстах.

За останні п'ять років значно зросла кількість публікацій, у яких обговорюються машинне навчання та нейронні мережі, особливо щодо соціальних мереж, медіа та пов'язаних факторів (рис. 1.6).

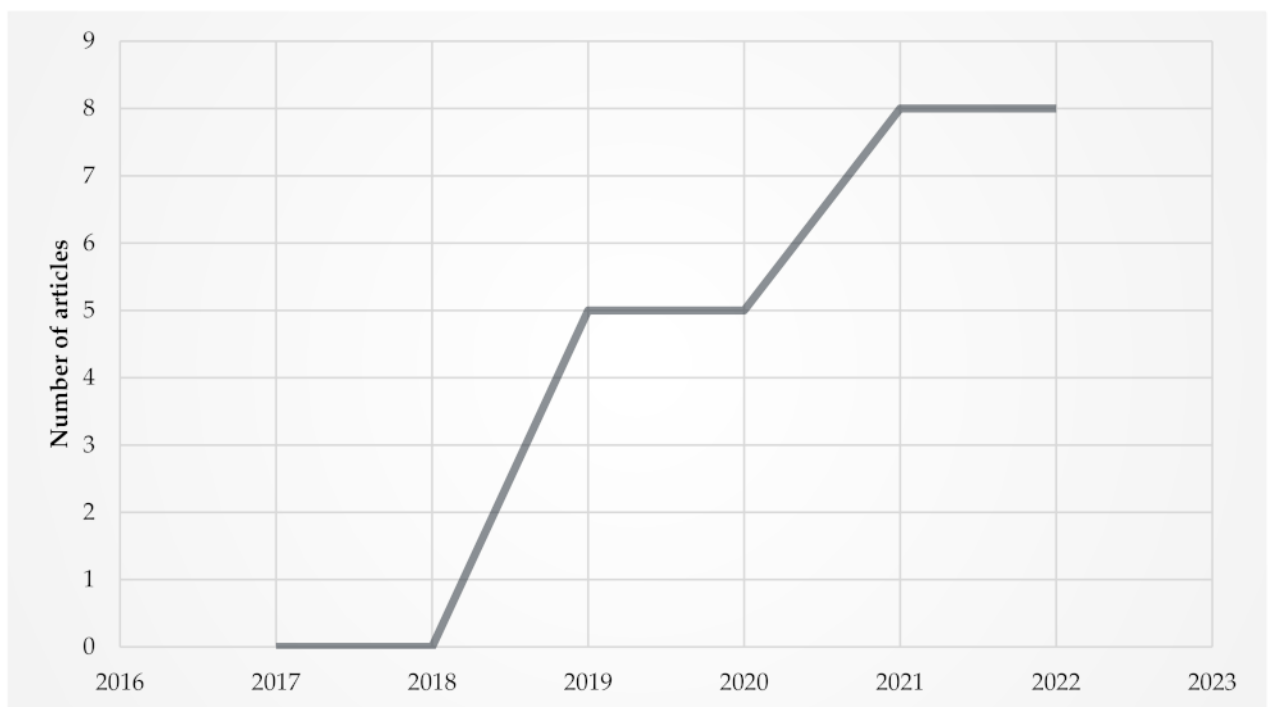


Рисунок 1.6 – Кількість включених робіт за останні роки (2017–2022).

Ця тенденція почалася з нестачі публікацій у 2017 та 2018 роках, але набула поширення в наступні роки. Цю тенденцію до зростання можна пояснити кількома факторами. По-перше, зростання інтересу та інвестицій у

штучний інтелект та машинне навчання вивели ці теми в мейнстрім. Як наслідок, існує більший попит на інформацію та публікації, які заглиблюються у складність цих тем. У цей час прогрес у машинному навчанні та нейронних мережах також був важливим. Примітно, що прогрес у цих дисциплінах відбувся у 2019 році та пізніше завдяки впровадженню нових моделей і алгоритмів, таких як конструкції на основі трансформаторів. ЗМІ агресивно повідомляли про ці досягнення, що, безсумнівно, сприяло збільшенню кількості публікацій. Крім того, увагу привернуло зростання використання машинного навчання та нейронних мереж у різних сферах, включаючи соціальні мережі та ЗМІ. Крім того, на шаблон міг вплинути поширення знань і попит на обізнаність.

Розподіл країн авторів у включених статтях про машинне навчання та нейронні мережі є значущим, оскільки він надає цінну інформацію про глобальну участь та внесок у ці галузі досліджень (рис. 1.7). Цей розподіл проливає світло на географічний розподіл дослідників, виділяючи регіони, які активно залучені до розвитку машинного навчання та нейронних мереж. Розуміння цього розподілу може сприяти обміну знаннями, співпраці, міжкультурним поглядам і прогресу в цих сферах.

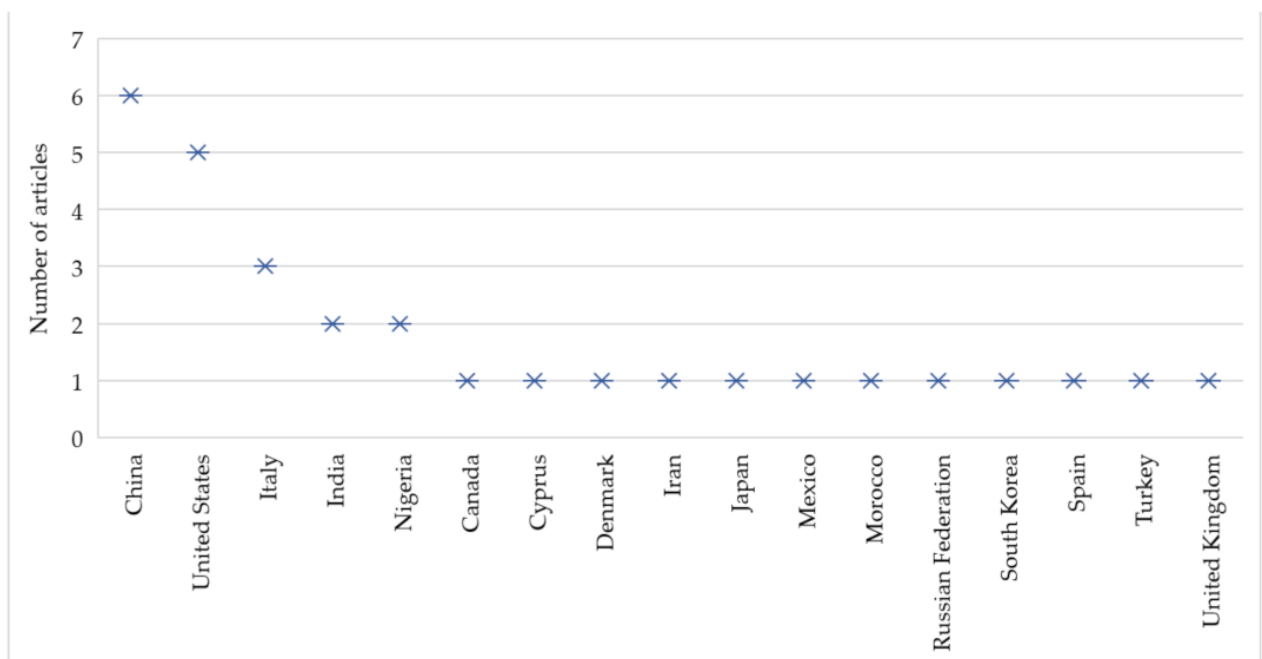


Рисунок 1.7 – Розподіл авторів по країнах

Кілька факторів можуть впливати на розподіл країн авторів у включених статтях про машинне навчання та нейронні мережі. На цей розподіл суттєво впливають дослідницькі та дослідницькі установи, такі як у Сполучених Штатах, Китаї та Великобританії. Ці країни зарекомендували себе як центри розвитку технологічних інновацій із добре фінансованими установами та технологічними компаніями, які активно вносять свій внесок у дослідницьку літературу. Крім того, країни з відомими академічними установами, які приділяють увагу дослідженням та інноваціям, мають більше шансів опублікувати більше робіт. На розподіл країн авторів у сфері машинного навчання та нейронних мереж також впливають такі фактори, як наявність фінансування, дослідницькі ресурси та міжнародна співпраця.

1.7. Напрямки сучасних досліджень

Вивчені статті демонструють різноманітні застосування машинного навчання та нейронних мереж у різноманітних сферах, підкреслюючи зростаючий інтерес до використання цих технологій для вирішення складних проблем. Нами проаналізовано підібрану добірку з 26 статей, з наголосом на основних темах і моделях, які випливають із статей. З включених документів випливають такі теми та напрями досліджень, які представлено на рис. 1.8:

1. машинне навчання та нейронні мережі – застосування алгоритмів машинного навчання, зокрема нейронних мереж, у різних сферах, таких як освіта, фінанси, охорона здоров'я та дослідження навколишнього середовища, є предметом численних публікацій;

2. гібридні моделі – у кількох публікаціях обговорюється використання гібридних моделей, які включають різні методи машинного навчання, такі як нейронні мережі з опорними векторними машинами, випадкові ліси та процеси Гауса, для підвищення продуктивності та точності;



Рисунок 1.8 – Теми та напрями досліджень у напрямі розвитку соціальних проектів

3. прогнозне моделювання – поширеною темою є використання машинного навчання для завдань прогнозування та класифікації, таких як прогнозування академічної успішності, оцінка параметрів, картографування схильності до зсувів, розпізнавання емоцій і прогнозування ризиків для безпеки;

4. міждисциплінарні програми – ці статті демонструють міждисциплінарну природу машинного навчання із застосуванням у таких галузях, як соціологія, нанотехнології, соціальні науки, прогнозування злочинності, звуковий дизайн і прогнозування молекулярної активності.

1.8. Завдання кваліфікаційної роботи

Доцільність виконання роботи зумовлена забезпеченням об'єктивного та прозорого процесу відбору соціальних проектів, спрямованих на розвиток громади, в умовах обмеженого фінансування та збільшення кількості

конкурсних заявок. Використання нейромережевих моделей дозволяє підвищити точність прогнозування конкурсних балів на основі багатофакторного аналізу, що знижує суб'єктивність оцінки та сприяє ефективному розподілу ресурсів. Застосування таких моделей дозволяє автоматизувати процес прогнозування, що прискорює оцінку заявок, забезпечує справедливість у виборі найбільш значущих проектів і завершує довіру учасників до результатів конкурсу.

Для досягнення мети дослідження передбачено виконання таких завдань:

- ✓ Аналіз існуючих методів оцінки конкурсного балу соціальних проектів. Огляд традиційних підходів та оцінка їх переваг і недоліків у контексті об'єктивності та точності оцінювання;

- ✓ Розробка концепції нейромережевої моделі для прогнозування конкурсного балу. Визначення основних етапів і компонентів моделі, вибір відповідної структури нейронної мережі з урахуванням специфіки завдання;

- ✓ Збір та підготовка даних. Вибір та обробка ключових факторів, що впливають на конкурсну оцінку проектів, формування навчальної вибору для моделей;

- ✓ Розробка, навчання та налаштування нейромережевої моделі. Побудова моделі на основі нейронних мереж, її навчання на зібраних даних і гіперналаштування параметрів для досягнення максимальної точності прогнозування;

- ✓ Аналіз результатів та оцінка ефективності моделі. Порівняння результатів отриманих результатів із традиційними методами оцінки, розробка рекомендацій для практичного застосування моделі в процесі конкурсного відбору.

РОЗДІЛ 2.

РОЗРОБКА КОНЦЕПЦІЇ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ ТА ПІДГОТОВКА ДАНИХ

2.1. Розробка концепції побудови нейромережових моделей

Розробка концепції побудови нейромережових моделей для прогнозування конкурсного балу соціальних проєктів розвитку громад є вагомим етапом, що забезпечує ефективність і точність моделі. Концепція включає декілька етапів: аналіз задачі, підготовка даних, вибір архітектури нейромереж, побудова, навчання та тестування моделей (рис. 2.1).



Рисунок 2.1 – Основні етапи побудови нейромережових моделей

Основною задачею є побудова моделі, яка здатна прогнозувати конкурсний бал проєкту на основі набору вхідних ознак. Вхідні дані включають такі характеристики проєктів, які представлені на рис. 2.2.



Рисунок 2.2 – Основні вхідні дані для створення нейромережевої моделі прогнозування конкурсного балу та підготовка даних

Задача належить до регресії, де нейромережа повинна передбачити числове значення (бал) на основі вхідних ознак.

Для якісного прогнозування необхідна ретельна підготовка даних:

- 1) Очищення даних – видалення пропусків, дублювання та некоректних значень;
- 2) Масштабування ознак – нормалізація даних для рівномірного внеску всіх ознак у процесі навчання;
- 3) Розподіл даних – розділення на тренувальну, тестову та валідаційну вибірки для запобігання переобученню.

Для побудови моделі побудовані архітектури нейронних мереж, які можуть бути адаптовані до різних задач (рис. 2.3).

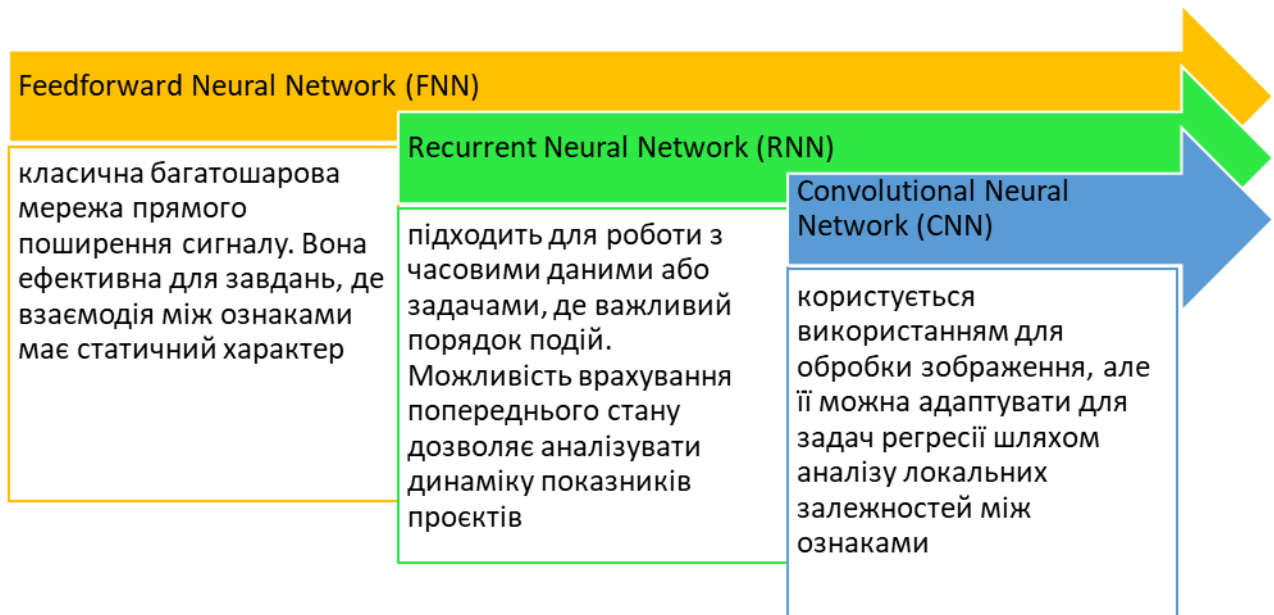


Рисунок 2.3 – Основні вхідні дані для створення нейромережевої моделі прогнозування конкурсного балу та підготовка даних

На основі підготовлених даних і обраної архітектури створено побудову нейромережі. При цьому виконується вибір кількості шарів і нейронів на кожній шарі. Також використовується функції активації, такі як ReLU, Sigmoid або Tanh, залежно від задачі. Проводиться налаштування оптимізаторів (Adam, SGD) для ефективного навчання. Здійснюється вибір функцій втрат, наприклад, `mean_squared_error` або `huber_loss`, для мінімізації помилок у прогнозах. У подальшому проводиться навчання моделей на поточних даних із перевіркою точності на валідаційній вибірці.

Після навчання виконується тестування моделей на окремій тестовій вибірці. Для оцінки якості моделей використовують наступні метрики:

- 1) середня квадратична помилка (MSE) – для оцінки середньої квадратичної помилки;
- 2) середня абсолютна похибка (MAE) – для аналізу середньої абсолютної помилки;
- 3) R-квадрат (R^2) – для вимірювання якості відповідності даним моделі;

4) середньоквадратична помилка (RMSE) – для оцінки розміру середньої помилки в одиницях вимірювання.

На основі результатів тестування моделі можуть бути оптимізовані. Для цього використовується регуляризація для запобігання перенавчанню. Також можливе додавання Dropout-шарів для підвищення стійкості моделі. Виконується пошук оптимальних гіперпараметрів (кількість шарів, нейронів, швидкість навчання) за допомогою GridSearch або RandomSearch.

Розробка концепції побудови нейромережових моделей є багатоетапним процесом, що включає аналіз задачі, підготовку вибору даних, оптимальної архітектури та тестування. Правильний підхід до реалізації кожного з етапів дозволяє отримати точну, адаптовану до потреби користувача модель, здатну прогнозувати конкурсний бал соціальних проєктів із високою точністю.

2.2. Вибір та підготовка даних

Для побудови нейромережових моделей прогнозування конкурсного балу соціальних проєктів розвитку громад Львівської області було зібрано дані про результати проведених конкурсів проєктів місцевих ініціатив за період 2020-2023 років. Ці дані включають показники реалізації соціальних проєктів, які фінансувалися із обласного бюджету. Загалом таблиця містить 16 атрибутів для 629 записів, які наведені у таблиці 2.1.

Таблиця 2.1 – Структура даних про реалізацію соціальних проєктів

№	Назва стовпця	Тип даних	Опис
1	2	3	4
1	Registration_number	int64	Реєстраційний ном
2	Project_name	object	Назва проєкту.
3	Territorial	object	Назва територіальної громади.
4	Settlement	object	Тип населеного пункту (місто, село тощо).

Продовження табл. 2.1

1	2	3	4
5	Project_budget	int32	Загальний бюджет проекту (грн).
6	Regional_budget	int64	Фінансування з обласного бюджету (грн).
7	District_budget	int32	Фінансування з районного
8	Public_budget	int32	Фінансування з
9	Sponsorship_funds	int32	Кошти від спонсорів (грн).
10	Financial_contrib	int32	Фінансовий внесок громади (грн).
11	Nonfinancial_cont	int32	Нефінансовий внесок громади (грн).
12	%_non- budgetary_contrib	float64	Частка позабюджетного фінансування (%)
13	Taxcapacity_index	float64	Індекс податкової
14	Score_taxability_index	float64	Бал за рівень податкової спроможності.
15	Final_score	float64	Підсумковий конкурс
16	Results	object	Результат конкурсу (переміг чи ні).

Для забезпечення точності та коректності роботи нейромережових моделей проведено підготовку даних. Насамперед виконано оцінку повноти даних. Аналіз даних показав, що всі атрибути мають повні записи без пропущених значень, про що свідчить результат методу `df.info()` (усі стовпці мають 629 записів).

Нами проведено аналіз типів даних. Для цього вид даних кожного атрибута перевірено за допомогою `df.dtypes`:

- ✓ числові дані – бюджет, внески, індекси (типи `int32`, `int64`, `float64`);
- ✓ категоріальні дані – назви громад, населених пунктів, назва проекту, результати (тип `object`).


```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 629 entries, 0 to 628
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Registration_number                    629 non-null    int64
1   Project_name                           629 non-null    object
2   Territorial_community                  629 non-null    object
3   Settlement                              629 non-null    object
4   Project_budget                         629 non-null    int32
5   Regional_budget                        629 non-null    int64
6   District_budget                       629 non-null    int32
7   Public_budget                         629 non-null    int32
8   Sponsorship_funds                     629 non-null    int32
9   Financial_contrib                      629 non-null    int32
10  Nonfinancial_contrib                   629 non-null    int32
11  %_non-budgetary_contrib                629 non-null    float64
12  Taxcapacity_index                     629 non-null    float64
13  Score_taxability_index                 629 non-null    float64
14  Final_score                            629 non-null    float64
15  Results                                629 non-null    object
dtypes: float64(4), int32(6), int64(2), object(4)
memory usage: 64.0+ KB

```

Рисунок 2.3 – Результати виведення інформації про дані

Під час очищення даних не було виявлено помилок чи пропусків. Однак для подальшої роботи з категоріальними даними їх було перекодовано у числовий формат.

Для нормалізації числових даних та покращення стабільності роботи нейронних мереж усі числові ознаки було масштабовано за допомогою `StandardScaler`. Це забезпечує приведення всіх значень до єдиного масштабу з середнім значенням 0 і стандартним відхиленням 1.

Категоріальні стовпці (`Territorial_community`, `Settlement`, `Results`) перекодовано в числовий формат за допомогою методу `one-hot encoding`, щоб надати моделям можливість враховувати якісні дані.

Після цього виконано розподіл на дві вибірки:

- ✓ тренувальну вибірку (80%) для навчання моделі;
- ✓ тестову вибірку (20%) для оцінки її точності.

Для зменшення розмірності даних і підвищення точності моделі були враховані результати кореляційного аналізу. Атрибути, що мали низьку

кореляцію з підсумковим балом (District_budget і Score_taxability_index), були виключені.

Зібрані дані забезпечують широкий спектр показників для аналізу ефективності реалізації соціальних проєктів. Проведені підготовчі дії гарантують, що дані є якісними, масштабованими та готовими до використання в нейромережових моделях для прогнозування конкурсного балу.

2.3. Результати аналізу взаємозв'язків між даними

Аналіз взаємозв'язків між даними є ключовим етапом для розуміння структури та поведінки соціальних проєктів, а також для вибору основних ознак, що впливають на прогнозування конкурсного балу. Результати аналізу представлені у вигляді графіків та кількісних оцінок.

На основі даних у стовпці Results, було побудовано кругову діаграму (рис. 2.4-2.5), яка відображає розподіл результатів проєктів.

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

# Створення палітри кольорів
colors = sns.color_palette('pastel')

# Побудова кругової діаграми для стовпця 'Results'
result_counts = df['Results'].value_counts()

# Заміна українських підписів англійськими (динамічний підхід)
labels = ['Winning proposals', 'Rejected proposals'] if len(result_counts) == 2 else result_counts.index

# Візуалізація кругової діаграми
plt.figure(figsize=(8, 8))
wedges, texts, autotexts = plt.pie(
    result_counts,
    labels=labels,
    autopct='%1.1f%%',
    colors=colors,
    startangle=90,
    textprops={'fontsize': 12, 'color': 'black'}
)
```

Рисунок 2.4 – Фрагмент коду для побудови розподілу результатів реалізації соціальних проєктів

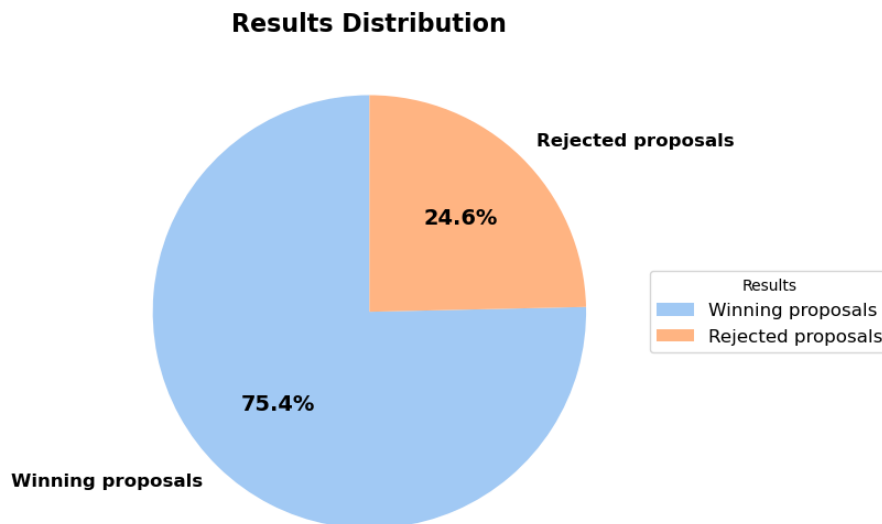


Рисунок 2.5 – Кругова діаграму, яка відображає розподіл результатів реалізації соціальних проєктів

Встановлено, що 75,4% проєктів є переможцями конкурсів (Winning proposals), а 24,6% проєктів було відхилено (Rejected proposals). Такий високий відсоток успішних проєктів свідчить про значну конкуренцію між громадами, а також про високий рівень підготовки заявок.

Нами виконано розподілу бюджетів проєктів (рис. 2.6-2.7).

```
# Додавання відомостей про графік
plt.title('Гістограма Project_budget зі Середнім значенням та СКВ', fontsize=16, weight='bold')
plt.xlabel('Project_budget', fontsize=14)
plt.ylabel('Частота', fontsize=14)

# Додавання лінії для середнього значення
mean_value = df['Project_budget'].mean()
plt.axvline(mean_value, color='red', linestyle='dashed', linewidth=2, label=f'Середнє = {mean_value:.2f}')

# Додавання ліній для середньоквадратичного відхилення
std_deviation = df['Project_budget'].std()
plt.axvline(mean_value - std_deviation, color='green', linestyle='dashed', linewidth=2, label=f'Середнє - СКВ = {mean_value - std_deviation:.2f}')
plt.axvline(mean_value + std_deviation, color='green', linestyle='dashed', linewidth=2, label=f'Середнє + СКВ = {mean_value + std_deviation:.2f}')

# Додавання текстових підписів на лініях
plt.text(mean_value, max(n) * 0.9, 'Середнє', color='red', fontsize=12, ha='center', weight='bold')
plt.text(mean_value - std_deviation, max(n) * 0.7, '-СКВ', color='green', fontsize=12, ha='center', weight='bold')
plt.text(mean_value + std_deviation, max(n) * 0.7, '+СКВ', color='green', fontsize=12, ha='center', weight='bold')

# Вивід легенди
plt.legend(fontsize=12)

# Оформлення графіка
plt.tight_layout()

# Відображення графіка
plt.show()
```

Рисунок 2.6 – Фрагмент коду для побудови розподілу бюджетів соціальних проєктів

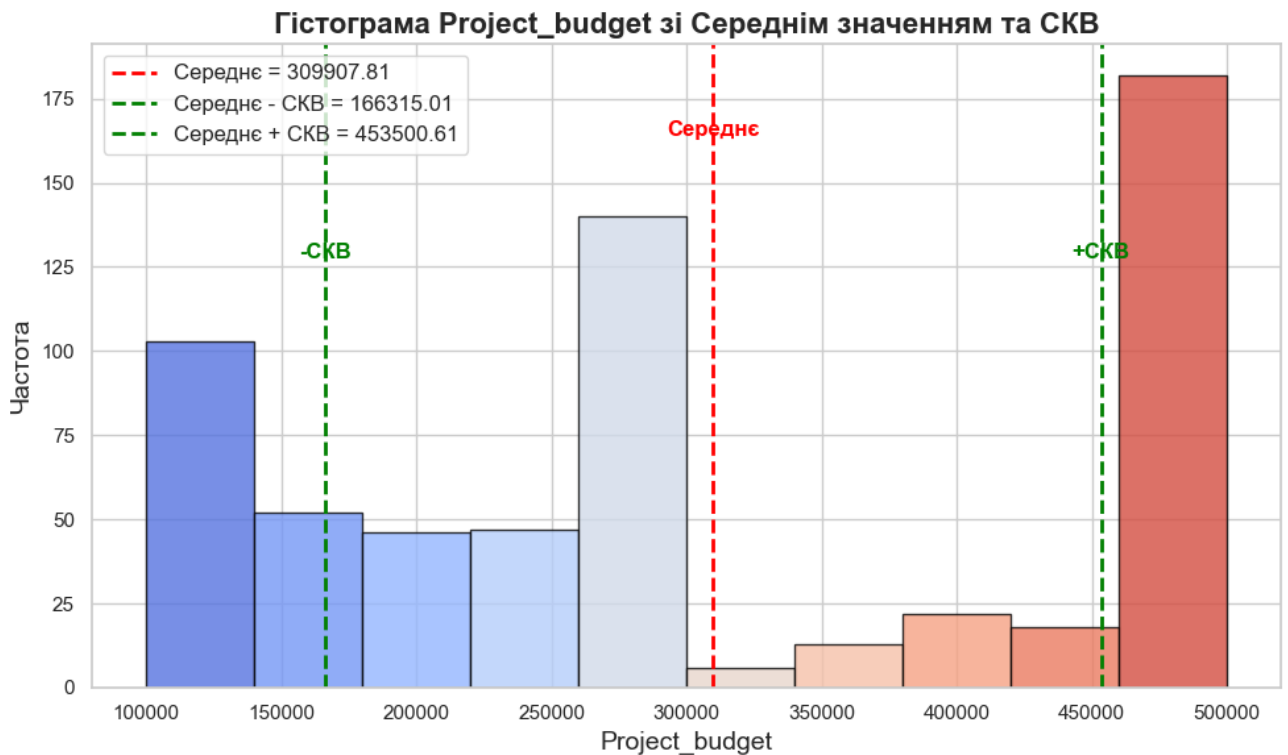


Рисунок 2.7 – Розподіл бюджетів соціальних проєктів

Отриманий розподіл бюджетів проєктів соціальних (рис. 2.7) показує, що середнє значення бюджету проєкту становить 315000 грн. Стандартне відхилення бюджету складає 152980 грн, що вказує на значну варіативність фінансування проєктів. Більшість проєктів мають бюджети в діапазоні 250000 – 500000 грн, з помітним піком у межах 500000 грн. Ця інформація є корисною для визначення середніх тенденцій у фінансуванні проєктів.

На гістограмі (рис. 2.8) представлені 20 територіальних громад, які найчастіше брали участь у конкурсах. Найбільшу кількість проєктів подали громади: Львівська громада – 53 проєкти; Дрогобицька громада – 33 проєкти; Червоноградська – 23 проєкти. Ці громади проявляють високу активність у реалізації соціальних ініціатив, що може бути зумовлено їхніми організаційними можливостями та фінансовою спроможністю.

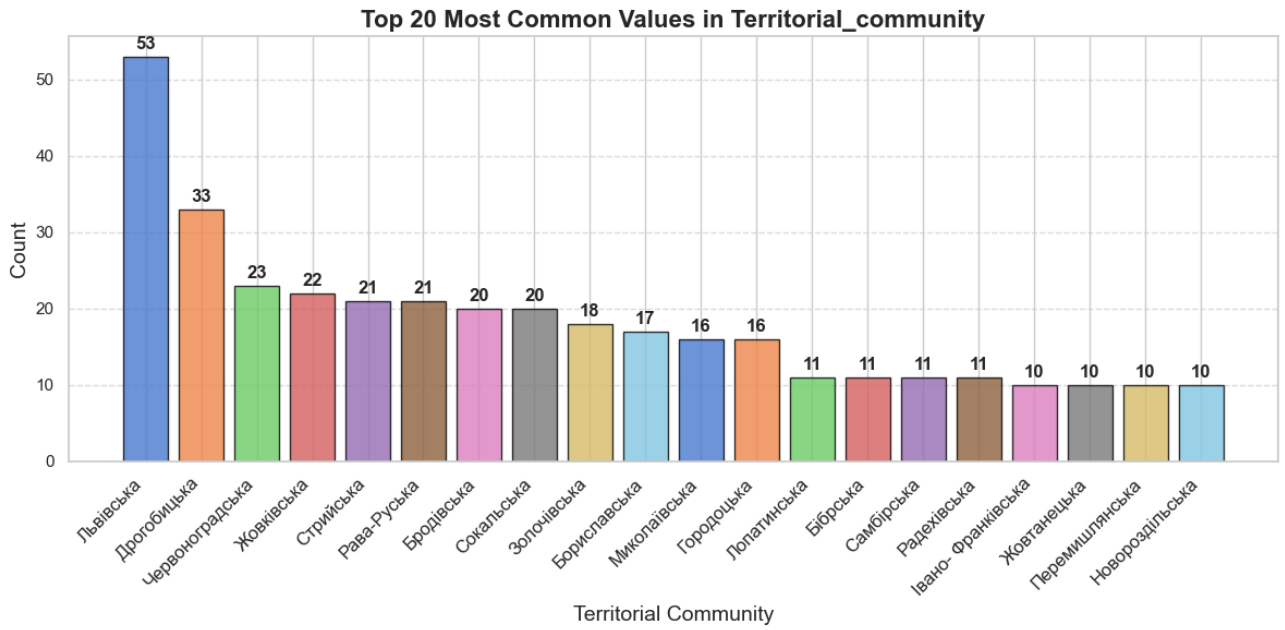


Рисунок 2.8 – Розподіл територіальних громад, які найчастіше брали участь у конкурсах

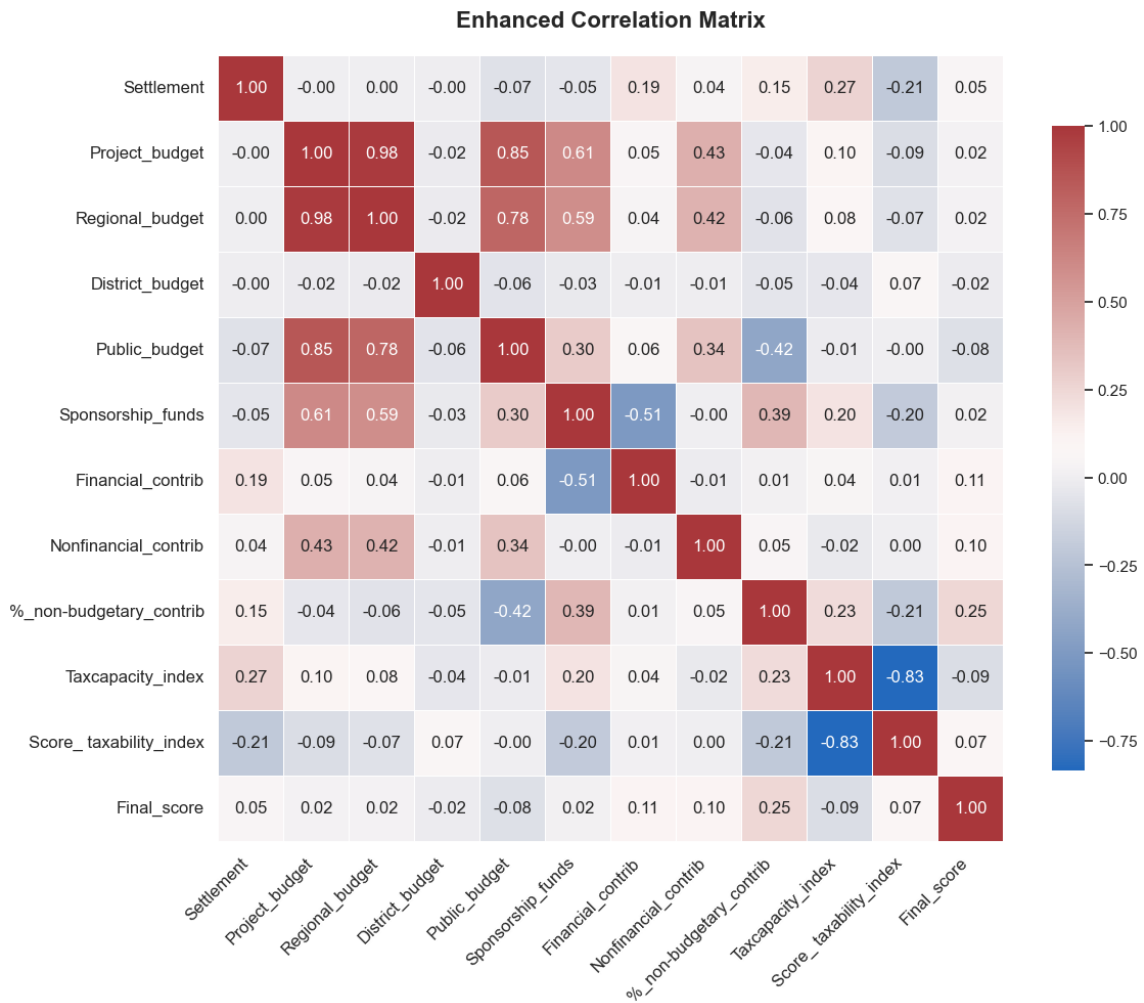


Рисунок 2.9 – Кореляційна матриця між показниками реалізації соціальних проєктів

На основі кореляційної матриці (рис. 2.9) було виявлено ключові залежності між змінними. Загальний бюджет проєкту має високу кореляцію з такими показниками:

- ✓ обласний бюджет (Regional_budget) – 0.98;
- ✓ частка позабюджетного фінансування (%_non-budgetary_contrib) – 0.43;
- ✓ рівень податкової спроможності громади (Taxcapacity_index) демонструє слабкий зв'язок із конкурсним балом (-0.09), що свідчить про незалежність підсумкового балу від податкового потенціалу.

Ці результати дозволяють зробити висновки щодо основних чинників, які слід враховувати під час побудови моделей прогнозування.

Результати аналізу показали, що основними чинниками, які впливають на конкурсний бал, є загальний бюджет проєкту, обласне фінансування та частка позабюджетного фінансування. Отримані залежності дозволяють зосередити увагу на цих показниках під час розробки нейромережових моделей прогнозування.

2.4. Показники оцінки точності моделей

Для оцінки точності нейромережових моделей прогнозування конкурсного балу використовуються різноманітні показники, які відображають якість роботи моделі з точки зору відповідності передбачень фактичним значенням. Нами наведено основні показники, які використовуються для аналізу точності.

Середньоквадратична помилка Mean Squared Error (MSE) визначається за формулою:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.1)$$

де y_i – фактичне значення; \hat{y}_i – прогнозне значення; n – кількість зразків.

MSE вимірює середній квадрат різниці між передбаченими та фактичними значеннями. Чим менше значення MSE, тим точніша модель. Однак ця метрика є чутливою до викидів через квадратування відхилень.

Середня абсолютна помилка Mean Absolute Error (MAE) визначається за формулою:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.2)$$

MAE вимірює середню абсолютну різницю між прогнозованими та фактичними значеннями. Вона менш чутлива до викидів, ніж MSE, і надає інтуїтивно зрозумілу оцінку середнього розміру помилки.

Коефіцієнт детермінації R-squared (R^2) визначається за формулою:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (2.3)$$

де \bar{y} – середнє значення фактичних даних.

R^2 показує, яка частка варіації у фактичних даних пояснюється моделлю. Значення R^2 близьке до 1 свідчить про високу якість моделі. Негативні значення вказують, що модель працює гірше, ніж проста середня.

Квадратний корінь із середньоквадратичної помилки Root Mean Squared Error (RMSE) визначається за формулою:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (2.4)$$

RMSE є інтерпретованою формою MSE, оскільки її результат має ті ж одиниці вимірювання, що й цільова змінна. Вона більш чутлива до великих помилок, ніж MAE.

Обернений квадратний корінь середньоквадратичної помилки Inverse RMSE (Inv RMSE) визначається за формулою:

$$\text{Inv RMSE} = \frac{1}{\text{RMSE}}. \quad (2.5)$$

Inv RMSE використовується для порівняння моделей: чим вище значення, тим краще модель. MSE і RMSE дозволяють виявити великі помилки моделі. Вони є особливо корисними у задачах, де важливі точність і мінімізація великих відхилень. MAE краще підходить для задач, де потрібно мінімізувати середню помилку незалежно від її масштабу. R^2 показує, наскільки добре модель описує дані. Це метрика високого рівня, яка допомагає оцінити якість моделі в цілому. Inv RMSE є додатковим критерієм, що дозволяє швидко оцінити точність моделі.

Застосування різних метрик дозволяє детально оцінити якість моделі з різних точок зору. У випадку задачі прогнозування конкурсного балу соціальних проєктів MAE та RMSE є основними метриками для оцінки помилок, оскільки вони відображають абсолютну точність та масштаб помилки. Значення R^2 використовується для загальної оцінки пояснювальної здатності моделі. Таким чином, поєднання цих метрик забезпечує комплексну оцінку ефективності нейромережових моделей.

2.5. Використання алгоритмів оптимізації для покращення моделей

Оптимізація є невід'ємною складовою процесу навчання нейромережових моделей. Вона дозволяє зменшити значення функції втрат, покращити якість передбачень та забезпечити узгодженість роботи моделей. Для цієї задачі використовуються різні алгоритми оптимізації, кожен із яких має свої переваги й недоліки.

Стохастичний градієнтний спуск (SGD) визначається за формулою:

$$w_{t+1} = w_t - \eta \cdot \nabla_w L(w_t). \quad (2.6)$$

де w_t – значення ваги на ітерації t ; η – швидкість навчання (learning rate); $\nabla_w L(w_t)$ – градієнт функції втрат L по відношенню до ваг.

Цей показник простий у реалізації, чутливий до вибору швидкості навчання (η) та може застрягти в локальних мінімумах. Використання динамічного η , що змінюється залежно від епохи, дозволяє уникати проблеми перенавчання.

Адаптивний градієнтний алгоритм (Adagrad) визначається за формулою:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t + \varphi}} \cdot \nabla_w L(w_t). \quad (2.7)$$

де G_t – накопичена сума квадратів градієнтів до ітерації t ; φ – мала константа для уникнення ділення на нуль.

Адаптивний градієнтний алгоритм (Adagrad) втоматично адаптує швидкість навчання для кожного параметра. Добре працює при великій кількості рідкісних ознак. Може «зупинитися» через дуже низький темп навчання на пізніх етапах.

Адаптивний моментний алгоритм (Adam) визначається за формулою:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla_w L(w_t) \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_w L(w_t))^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ w_{t+1} &= w_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varphi}} \end{aligned} \quad (2.8)$$

де m_t – експоненційно згладжена середня градієнтів (момент); v_t – експоненційно згладжена середня квадратів градієнтів; β_1, β_2 – параметри згладжування.

Особливостями адаптивного моментного алгоритму (Adam) є те, що він комбінує переваги моментного градієнта і Adagrad. Він є стабільний у задачах з високою варіативністю даних. Використання оптимізатора AdamW (Adam з L2-регуляризацією), зменшує потребу переобладнання.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ СТВОРЕННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ ПРОЄКТІВ РОЗВИТКУ ГРОМАД

3.1. Архітектури нейромережєвих моделей для прогнозування конкурсного балу соціальних проєктів

У нашій роботі розглянуто три основні архітектури нейромережєвих моделей, які використовуються для прогнозування конкурсного балу соціальних проєктів – Feedforward Neural Network (FNN), Recurrent Neural Network (RNN) та Convolutional Neural Network (CNN). Кожна з цих архітектур має свої переваги та використовується залежно від структури вхідних даних і задачі моделювання.

3.1.1. Feedforward Neural Network (FNN)

Архітектура Feedforward Neural Network є найпростішою архітектурою штучної нейронної мережі (рис. 3.1).

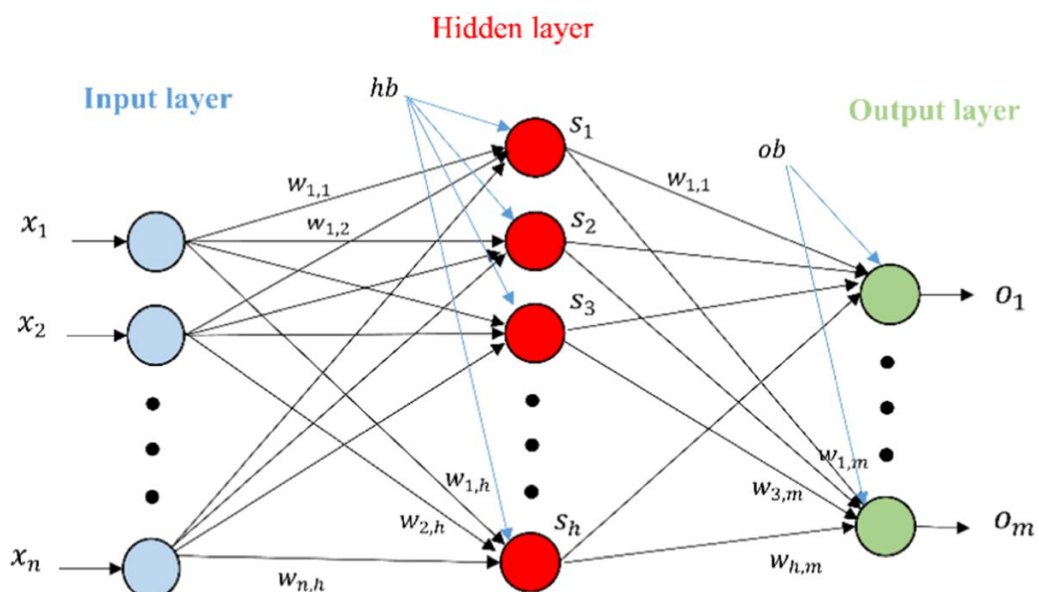


Рисунок 3.1 – Архітектура Feedforward Neural Network

Вона складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Кожен нейрон у шарі підключений до нейронів наступного шару без зворотних зв'язків. Формула активації нейрона має вигляд:

$$\begin{aligned} z^{(l)} &= W^{(l)} \cdot a^{(l-1)} + b^{(l)} \\ a^{(l)} &= f(z^{(l)}) \end{aligned} \quad (3.1)$$

де $W^{(l)}$ – матриця ваг для l -го шару; $a^{(l-1)}$ – вихід попереднього шару; $b^{(l)}$ – вектор зсуву; $f(z)$ – активаційна функція (наприклад, ReLU або Sigmoid).

Використовується для задач регресії та класифікації. Модель FNN обробляє лише табличні дані без часової залежності.

3.1.2. Recurrent Neural Network (RNN)

Архітектура Recurrent Neural Network є моделлю, яка враховує часову залежність у даних (рис. 3.2).

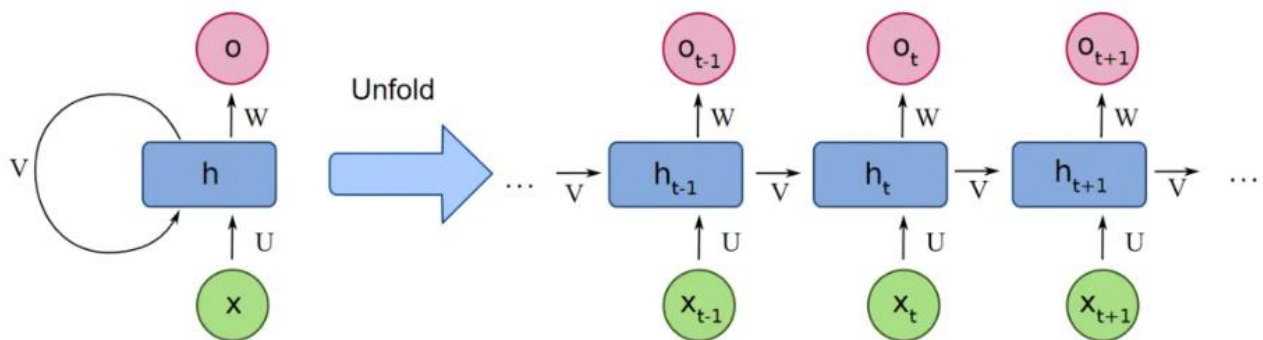


Рисунок 3.2 – Архітектура Recurrent Neural Network

Основна особливість RNN – зворотний зв'язок між нейронами, що дозволяє використовувати інформацію з попередніх елементів послідовності. Формула роботи нейрона має вигляд:

$$h_t = f(W_x \cdot x_t + W_h \cdot h_{t-1} + b). \quad (3.2)$$

де h_t – стан прихованого шару в момент часу t ; W_x , W_h – ваги для вхідного сигналу та прихованого стану відповідно; x_t – вхідні дані в момент часу t ; b – вектор зсуву.

Особливістю Recurrent Neural Network є те, що така архітектура підходить для обробки послідовних даних, таких як часові ряди. Чутлива до проблеми затухання градієнтів, яку вирішують покращені архітектури (LSTM, GRU).

3.1.3. Convolutional Neural Network (CNN)

Архітектура Convolutional Neural Network найчастіше використовується для обробки зображень, але також застосовується для табличних даних у вигляді згорток. CNN витягує локальні залежності між ознаками, що забезпечує високу точність при моделюванні складних взаємозв'язків. Формула згортки має вигляд:

$$z_{i,j}^{(l)} = \sum_m \sum_n X_{i+m,j+n} \cdot K_{m,n} + b. \quad (3.3)$$

де $X_{i+m,j+n}$ – матриця вхідних даних; $K_{m,n}$ – ядро згортки; b – вектор зсуву.

Особливостями цієї архітектури є те, що вона виявляє локальні патерни в даних. Використовується для задач регресії та класифікації.

Кожна з представлених архітектур має свої переваги. *FNN* є простою та ефективною для обробки табличних даних. *RNN* враховує часові залежності, що дозволяє моделювати вплив попередніх значень. *CNN* знаходить локальні залежності між ознаками, що забезпечує вищу точність для складних даних.

Вибір конкретної архітектури залежить від природи даних і поставленої задачі. У процесі навчання моделей використовувалися оптимізовані архітектури для досягнення найкращих результатів у прогнозуванні конкурсного балу соціальних проєктів.

3.2. Побудова нейромережових моделей для прогнозування конкурсного балу соціальних проєктів розвитку громад

3.2.1. Побудова Feedforward Neural Network (FNN)

Feedforward Neural Network (FNN) – це одна з найпростіших структур штучних нейронних мереж, яка широко застосовується для завдань прогнозування та класифікації. У цій моделі інформація передається в одному напрямку – від вхідного шару через закриті шари до вихідного. Модель не містить зворотних зв'язків, що забезпечують простоту її реалізації (рис. 3.3).

FNN у цьому дослідженні використовується для прогнозування конкурсного балу соціальних проєктів. На першому етапі виконується попередня обробка даних, яка включає нормалізацію вхідних значень для зменшення впливу масштабів ознак (рис. 3.3).

```
# Завантаження даних та розділення на ознаки та цільову змінну
X = df.drop('Final_score', axis=1).values
y = df['Final_score'].values

# Масштабування ознак
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Розділення на тренувальний та тестовий набори даних
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Функція для обчислення метрик
def evaluate_model(y_test, y_pred, model_name):
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    rmse = np.sqrt(mse)
    inv_rmse = 1 / rmse if rmse != 0 else 0
    print(f"\nModel: {model_name}")
    print("Mean Squared Error (MSE):", mse)
    print("Mean Absolute Error (MAE):", mae)
    print("R^2 Score:", r2)
    print("Root Mean Squared Error (RMSE):", rmse)
    print("Inverse RMSE:", inv_rmse)
    return mse, mae, r2, rmse, inv_rmse
```

Рисунок 3.3 – Фрагмент коду для завантаження даних та розділення на ознаки та цільову змінну, а також обчислення метрик

Насамперед нами виконано завантаження даних та розділення на ознаки та цільову змінну. Оцінка точності проводиться за допомогою метрик, таких як MSE, MAE, R^2 , RMSE, Inv_RMSE.

Побудова архітектури нейронної мережі складається з трьох шарів – вхідного, двох прихованих і вихідного (рис. 3.4)

```
# 1. Feedforward Neural Network (FNN)
model_fnn = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(1)
])
model_fnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_fnn = model_fnn.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=16, verbose=0)
y_pred_fnn = model_fnn.predict(X_test).flatten()
metrics_fnn = evaluate_model(y_test, y_pred_fnn, "FNN")

# Вивід інформації про модель FNN
print("\nFNN Model Summary:")
model_fnn.summary()
```

Рисунок 3.4 – Фрагмент коду для навчання моделі
Feedforward Neural Network (FNN)

Навчання моделі виконується на тренувальній вибірці з оцінкою результатів на тестовій вибірці. На цьому етапі дані розділяються на тренувальний та тестовий набори, а також нормалізуються для усунення масштабних відмінностей між ознаками. Модель має дві закриті шари (рис. 3.5).

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 64)	512
dense_7 (Dense)	(None, 32)	2,080
dense_8 (Dense)	(None, 1)	33

Total params: 7,877 (30.77 KB)

Trainable params: 2,625 (10.25 KB)

Non-trainable params: 0 (0.00 B)

optimizer params: 5,252 (20.52 KB)

Рисунок 3.5 – Архітектура нейронної мережі Feedforward Neural Network (FNN)

Перший шар складається з 64 нейронів із цією функцією активації ReLU. Другий шар із 32 нейронів із тією ж функцією активації. Модель навчається протягом 50 епох з розміром батчу 16. Валідаційні дані можуть відстежувати

точність під час навчання. На цьому етапі прогнозування на вибірці тестів та оцінки точності виконується за метриками MSE, MAE. Візуалізація навчання представлена на рис. 3.6.

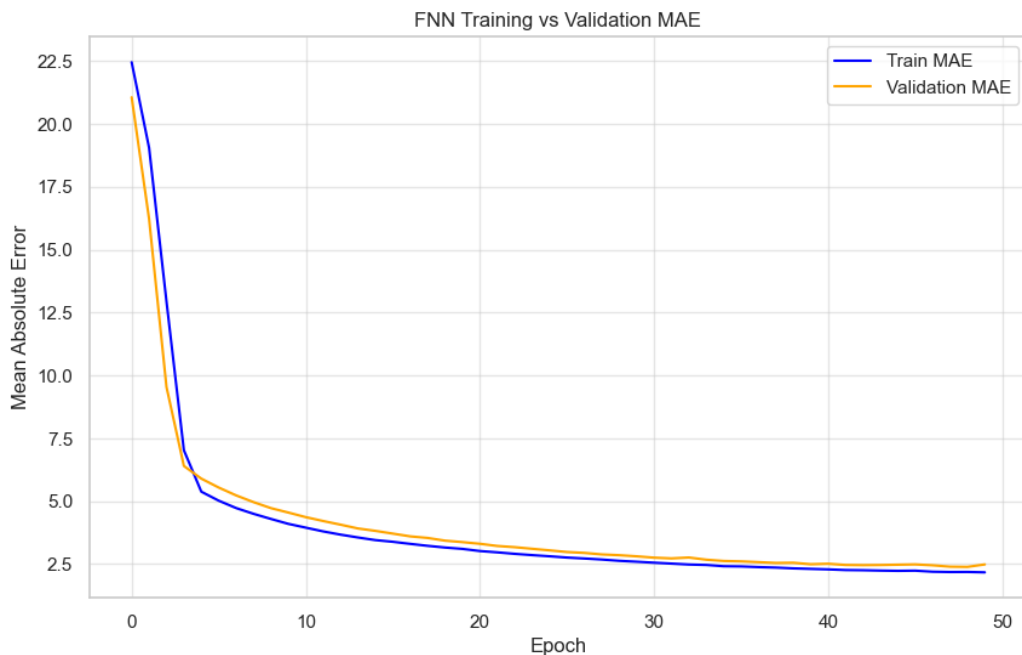


Рисунок 3.6 – Тенденції зміни середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних впродовж навчання нейронної мережі Feedforward Neural Network (FNN)

Графік демонструє зміну середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних в епоху навчання. Отже, Feedforward Neural Network є базовою моделлю, яка показує стабільні результати в задачах прогнозування. Це забезпечує легкість у налаштуваннях та швидке навчання. На основі отриманих метрик модель демонструє адекватну точність, що робить її одним із варіантів для подальших експериментів.

3.2.2. Побудова Recurrent Neural Network (RNN)

Рекурентна нейронна мережа (RNN) являє собою тип нейронних мереж, які враховують взаємозв'язок між даними в часі або конфліктах. Завдяки

зворотним зв'язкам у архітектурі RNN дозволяється зберегти інформацію про попередні стани, що робить її ефективною для аналізу наступних даних.

Нами використовується RNN для прогнозування конкурсного балу соціальних проєктів. Основні етапи побудови включають:

Побудова архітектури мережі передбачає один прийнятий шар із функцією активації ReLU (рис. 3.7).

```
# 2. Recurrent Neural Network (RNN)
X_train_rnn = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test_rnn = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
model_rnn = Sequential([
    SimpleRNN(64, activation='relu', input_shape=(X_train_rnn.shape[1], X_train_rnn.shape[2])),
    Dense(1)
])
model_rnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_rnn = model_rnn.fit(X_train_rnn, y_train, validation_data=(X_test_rnn, y_test), epochs=50, batch_size=16, verbose=0)
y_pred_rnn = model_rnn.predict(X_test_rnn).flatten()
metrics_rnn = evaluate_model(y_test, y_pred_rnn, "RNN")

# Вивід інформації про модель RNN
print("\nRNN Model Summary:")
model_rnn.summary()

# Схема моделі RNN
plot_model(model_rnn, to_file='model_rnn.png', show_shapes=True, show_layer_names=True)
```

Рисунок 3.7 – Фрагмент коду для навчання моделі
Recurrent Neural Network (RNN)

Recurrent Neural Network потребує тривимірного формату вхідних даних: (кількість прикладів, кількість часових кроків, кількість ознак).

Модель має один прихований шар із 64 нейронами, які задають функцію активації ReLU. Вихідний шар містить один нейрон для прогнозування конкурсного балу (рис. 3.8).

Layer (type)	Output Shape	Param #
simple_rnn_1 (SimpleRNN)	(None, 64)	4,608
dense_9 (Dense)	(None, 1)	65

Total params: 14,021 (54.77 KB)

Trainable params: 4,673 (18.25 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 9,348 (36.52 KB)

Рисунок 3.8 – Архітектура нейронної мережі Recurrent Neural Network (RNN)

Мережа навчається протягом 50 епох із розміром батчу 16. У процесі навчання використовують як тренувальні, так і тестувальні дані.

Візуалізація навчання представлена на рис. 3.9.

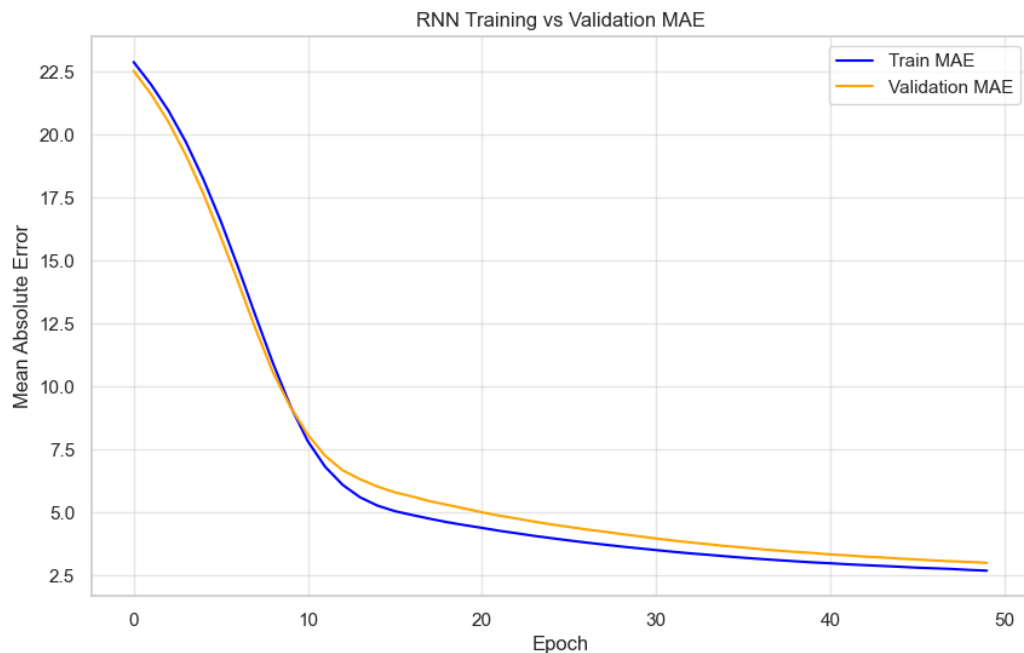


Рисунок 3.9 – Тенденції зміни середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних впродовж навчання нейронної мережі Recurrent Neural Network (RNN)

Встановлено, що Recurrent Neural Network ефективна для прогнозування конкурсного балу соціальних проєктів з можливістю завершення наступного взаємодії між даними. Використання RNN забезпечує точні результати, а її структура легко адаптується до інших завдань із тимчасовими або послідовними залежностями.

3.2.3. Побудова Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) є потужним інструментом у сфері машинного навчання, який спеціалізується на виявленні та обробці локальних залежностей у даних. Хоча CNN часто використовується для обробки

зображень, її можливості ефективні й для роботи з табличними даними після перетворення їх у відповідний формат. Розглянемо побудову CNN для прогнозування конкурсного балу соціальних проєктів.

Для роботи CNN дані мають бути перетворені у формат, придатний для згорткових операцій. У нашому випадку це тривимірний формат, де кожен приклад представляється як масив із формою (кількість ознак, 1) (рис. 3.10).

```
# 3. Convolutional Neural Network (CNN)
X_train_cnn = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test_cnn = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
model_cnn = Sequential([
    Conv1D(32, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1], 1)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(1)
])
model_cnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_cnn = model_cnn.fit(X_train_cnn, y_train, validation_data=(X_test_cnn, y_test), epochs=50, batch_size=16, verbose=0)
y_pred_cnn = model_cnn.predict(X_test_cnn).flatten()
metrics_cnn = evaluate_model(y_test, y_pred_cnn, "CNN")

# Вивід інформації про модель CNN
print("\nCNN Model Summary:")
model_cnn.summary()

# Схема моделі CNN
plot_model(model_cnn, to_file='model_cnn.png', show_shapes=True, show_layer_names=True)
```

Рисунок 3.10 – Фрагмент коду для навчання моделі
Convolutional Neural Network (CNN)

Насамперед нами виконано перетворення даних у тривимірний формат. Цей крок важливий, оскільки згорткові шари працюють із багатовимірними масивами, а табличні дані за замовчуванням є одномірними.

Основними елементами CNN є згорткові шари (Conv1D), що виявляють локальні особливості даних, як-от взаємозв'язки між сусідніми ознаками, шар Flatten, що перетворює багатовимірний вихід згорткового шару в одномірний вектор, придатний для Dense-шарів та повнозв'язні шари (Dense), які завершують обробку даних і формують кінцевий результат.

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 5, 32)	128
flatten_1 (Flatten)	(None, 160)	0
dense_10 (Dense)	(None, 64)	10,304
dense_11 (Dense)	(None, 1)	65

Total params: 31,493 (123.02 KB)

Trainable params: 10,497 (41.00 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 20,996 (82.02 KB)

Рисунок 3.11 – Архітектура нейронної мережі
Convolutional Neural Network (CNN)

Згортковий шар Conv1D(32, kernel_size=3) має 32 фільтри та ядро розміром 3, що виділяє локальні патерни. Прихований шар Dense(64) із 64 нейронами обробляє виділені особливості. Вихідний шар Dense(1) із одним нейроном для прогнозування конкурсного балу. Для оптимізації ваг використовується оптимізатор adam, а функція втрат – mean_squared_error.

Модель навчається на тренувальних даних із використанням заданої кількості епох і розміру батчу (рис. 3.12).

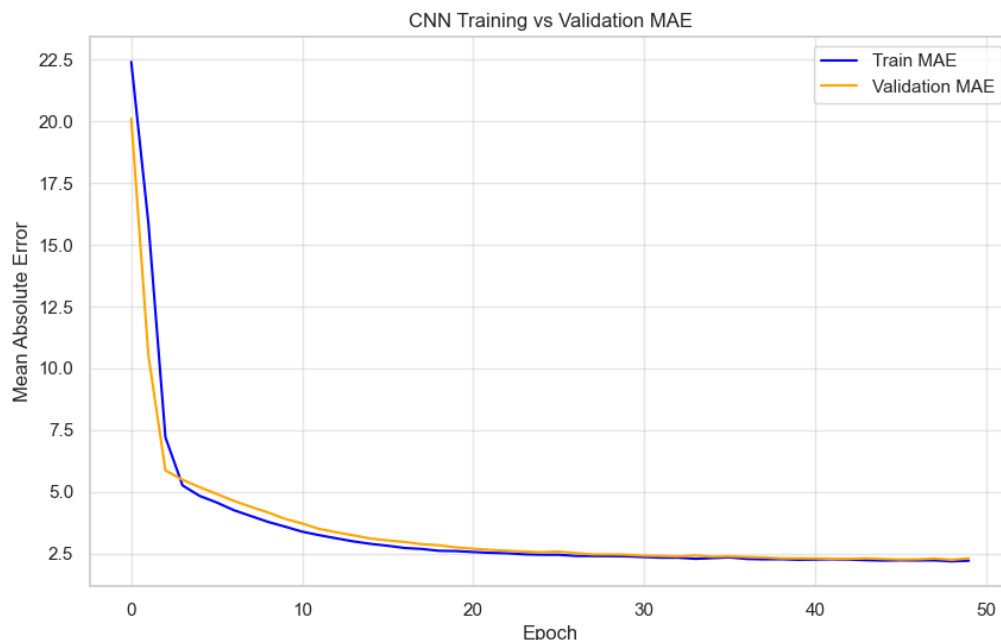


Рисунок 3.12 – Тенденції зміни середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних впродовж навчання нейронної мережі
Convolutional Neural Network (CNN)

Після навчання та оцінки моделі CNN було виявлено, що вона демонструє високі результати точності завдяки своїй здатності ефективно виділяти локальні особливості вхідних даних. Графіки показали, що модель добре навчається на даних без значного перенавчання, а також має конкурентоспроможні метрики точності.

3.3. Порівняння результатів та аналіз точності

Для оцінки ефективності побудованих моделей (FNN, RNN, Convolutional CNN) проведено порівняння їхніх результатів за основними метриками точності. Моделі оцінювалися за наступними показниками:

- ✓ середня квадратична помилка (MSE);
- ✓ середня абсолютна помилка (MAE);
- ✓ коефіцієнт детермінації R^2 Score;
- ✓ корінь середньоквадратичної помилки RMSE;
- ✓ обернений показник RMSE Inverse RMSE ($1/\text{RMSE}$).

Результати порівняння показників цих моделей представлено у таблиці 3.1.

Таблиця 3.1 – Результати оцінки точності створених нейромережевих моделей

Модель	MSE	MAE	Оцінка R^2	RMSE	Інв RMSE
FNN	9,087	2,483	0,795	3,015	0,332
RNN	14,165	2,998	0,311	3,764	0,266
CNN	7,823	2,305	0,921	2,797	0,358

Тенденції зміни середньо квадратичної помилки (MSE) представлено на рис. 3.13, а. Найменшу MSE показала модель CNN (7.823), що працює про її здатність краще прогнозувати конкурсні бали. Модель FNN також показала конкурентоспроможний результат (9,087), хоча й поступилася CNN. Модель RNN має найвищу MSE (14.165), що вказує на високу точність.

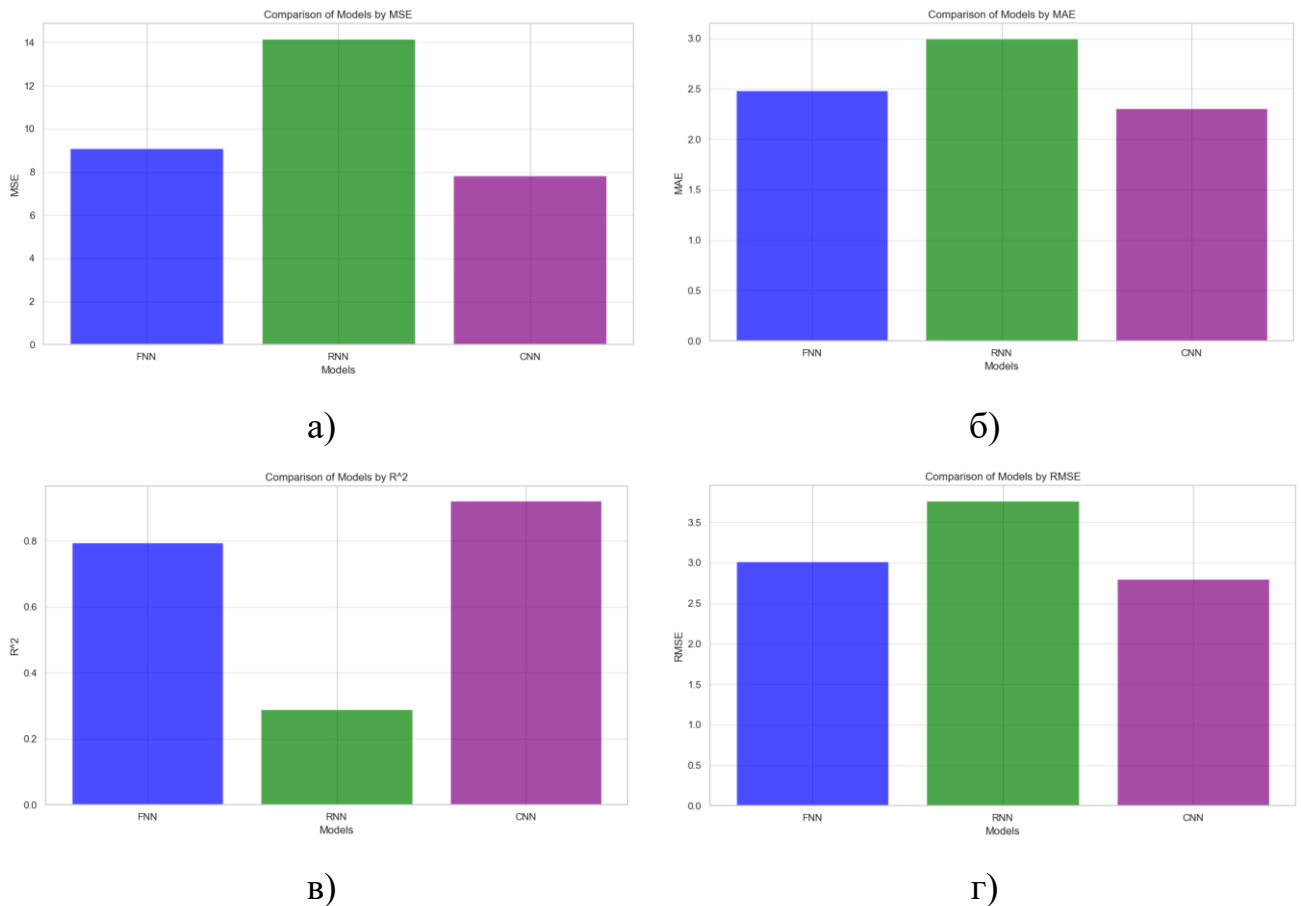


Рисунок 3.13 – Тенденції зміни показників точності створених нейромережевих моделей

Тенденції зміни середньої абсолютної похибки (MAE) представлено на рис. 3.13. б. Найнижче MAE досягнуто для CNN (2,305), що підтверджує її перевагу в точності прогнозів. FNN дає друге місце з MAE 2.483, тоді як RNN має найбільший результат – 2.998.

Оцінка R^2 представлена на рис. 3.13. в. Найвищий коефіцієнт визначення отримано для CNN (0,921), що свідчить про сильну залежність між прогнозованими та реальними значеннями. FNN також має високий R^2 (0,795), що свідчить про її конкурентоспроможність. Для RNN коефіцієнт R^2 значно нижчий (0,311), що свідчить про її слабку узгодженість з реальними даними.

Тенденції зміни середньоквадратичної помилки (RMSE) представлено на рис. 3.13. г. Найнижчий RMSE у CNN (2.797), що підтверджує її точність. FNN має значення RMSE 3.015, що трохи вище за CNN. Найвищий RMSE у RNN (3.764).

Отже модель CNN продемонструвала найкращі результати за всіма метриками, що робить її найкращим вибором для прогнозування конкурсних балів. FNN є конкурентоспроможною моделлю та може бути використана як альтернатива. RNN має слабкі показники точності, що недостатньо про необхідність її подальшої оптимізації або застосування інших підходів. Згідно з проведеним аналізом, CNN є найбільш ефективною моделлю для поставленої задачі прогнозування.

3.4. Оптимізація моделі Convolutional Neural Network (CNN)

Для підвищення точності прогнозування та зниження помилок моделі Convolutional Neural Network (CNN) було застосовано кілька методів оптимізації, які включають регулярне додавання, використання покращеного функціоналу втрат, оптимізатора з меншою швидкістю навчання та більш тривалого навчання.

Під час оптимізації архітектури базової CNN було внесено такі зміни:

- 1) регуляризація Dropout – Для зменшення переобучення додано шар Dropout з ймовірністю 0.3, що випадковим чином вимикає нейрони під час навчання;
- 2) додатковий згортковий шар – Додано ще один шар згортки (Conv1D) із меншою кількістю фільтрів (32), що забезпечує глибше вилучення позначки;
- 3) регуляризація L2 – Використано регуляризацію L2 у повнозв'язному шарі для уникнення сильного узагальнення;
- 4) покращений оптимізатор – Оптимізатор Adam з меншою швидкістю навчання (0,001) забезпечує більш точну мінімізацію функції втрат;

5) функціональний втрат Huber Loss – Функціональний втрат Huber є менш чутливою до викидів, завдяки стандартній середньоквадратичній помилці.

Код реалізації для створення оптимізованої архітектури CNN представлено на рис. 3.14.

```
# Поліпшена модель CNN
model_cnn_improved = Sequential([
    Conv1D(64, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1], 1)),
    Dropout(0.3), # Dropout для регуляризації
    Conv1D(32, kernel_size=3, activation='relu'),
    Flatten(),
    Dense(64, activation='relu', kernel_regularizer=l2(0.01)), # L2 регуляризація
    Dense(1)
])

# Оптимізатор Adam з меншою швидкістю навчання
optimizer = Adam(learning_rate=0.001)

# Використання Huber Loss
loss_function = Huber(delta=1.0) # delta можна налаштувати залежно від проблеми

model_cnn_improved.compile(optimizer=optimizer, loss=loss_function, metrics=['mae'])

# Навчання моделі
history_cnn_improved = model_cnn_improved.fit(
    X_train_cnn, y_train,
    validation_data=(X_test_cnn, y_test),
    epochs=100,
    batch_size=16,
    verbose=1
)

# Прогнозування
y_pred_cnn_improved = model_cnn_improved.predict(X_test_cnn).flatten()

# Оцінка покращеної моделі
metrics_cnn_improved = evaluate_model(y_test, y_pred_cnn_improved, "Improved CNN")

# Графік навчання покращеної моделі
plot_training(history_cnn_improved, "Improved CNN Training vs Validation MAE")
```

Рисунок 3.14 – Фрагмент коду для оптимізації архітектури CNN

Оптимізована модель навчалася протягом 100 епох із використанням невеликого розміру пакета даних (16). Валідація виконувалася на основі тестових даних. Для оцінки точності покращеної моделі використовували метрики MSE, MAE, R^2 , RMSE та обернений RMSE. Результати показали суттєве покращення. Зокрема, MSE=7.45, MAE=2.3, $R^2=0.957$, RMSE=2.73 та Inverse RMSE=0.36.

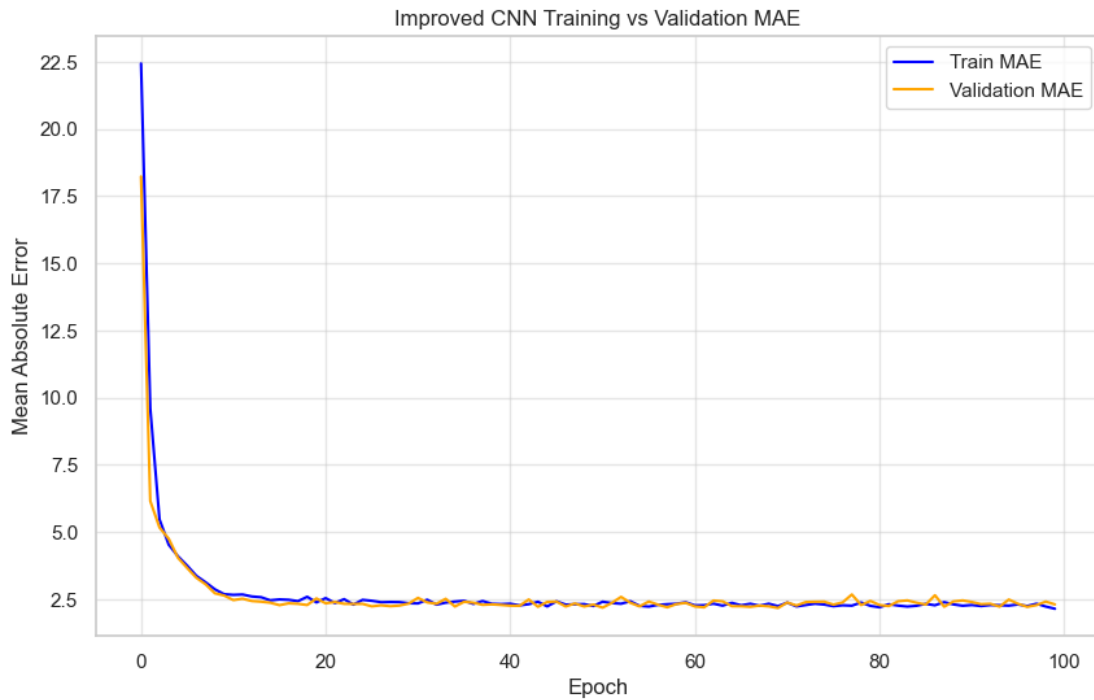


Рисунок 3.15 – Тенденції зміни середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних впродовж навчання оптимізованої нейронної мережі Convolutional Neural Network (CNN)

Покращена модель продемонструвала зниження MSE і MAE порівняно з базовою CNN. Використання Huber Loss зробило модель менш чутливою до викидів у дані. Регуляризація Dropout і L2 допомогли уникнути перенавчання, що видно з менших розривів між навчанням і валідацією. Значення R^2 для покращеної CNN зросло, що свідчить про більш точне моделювання залежностей.

Оптимізація моделі CNN значно покращила її точність і стабільність. Запропонована архітектура є ефективною для прогнозування конкурсного балу соціальних проєктів, демонструючи найкращі результати серед усіх тестованих моделей.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпечних чинників та розробка заходів із покращення умов праці

Розробка заходів із покращення умов праці передбачає детальний аналіз небезпечних чинників, які можуть впливати на працівників, а також визначення ефективних заходів для зниження ризиків. Нами розглянуто основні небезпечні чинники на виробництві, оцінено їхній вплив та запропонуємо заходи для їх мінімізації.

Для проведення аналізу було зібрано інформацію про найбільш поширені небезпечні чинники на робочих місцях. Ці дані представлені в таблиці 4.1.

Таблиця 4.1. Небезпечні чинники та заходи для їх усунення

№ п/п	Небезпечний чинник	Опис впливу	Потенційні наслідки	Запропоновані заходи
1	Високий рівень шуму	Постійний вплив на шум понад 85дБ	Погіршення слуху, стрес	Використання шумопоглинаючих матеріалів; видання захисних навушників
2	Недостатнє освіта	Низький рівень освіти в робочих зонах	Зниження продуктивності, травматизм	Встановлення LED-світильників, регулярна перевірка освітлення
3	Підвищений рівень пилу	Наявність великої кількості пилу у повітрі	Респіраторні захворювання	Використання системи пиловловлення, регулярне прибирання
4	Хімічні речовини	Контакт із токсичними речовинами	Інтоксикація, шкірні захворювання	використання захисного одягу; контроль вентиляції
5	Висока температура	Робота в умовах температури понад 30°C	Перегрів, зниження концентрації	Організація системи кондиціонування повітря, перерви для відпочинку
6	Небезпечне обладнання	Наявність незахищених частин обладнання	Травматизм, ампутації кінцівок	встановлення захисних екранів; регулярне технічне обслуговування
7	Високі фізичні навантаження	Постійне підняття важких вантажів	Проблеми з опорно-руховим апаратом	Організація механізмів для підйому вантажів; навчання ергономіці

На основі аналізу небезпечних чинників встановлено, що найпоширенішими проблемами є високий рівень шуму та недостатнє освітлення. Ці чинники призводять до значного погіршення умов праці, зниження продуктивності та підвищення ризику травм. Особливої уваги потребують робочі місця, де працівники мають контакт із хімічними речовинами або працюють за високих температур.

Для кожного небезпечного чинника запропоновано відповідні заходи:

- 1) шумопоглинання – установка шумозахисних бар'єрів, видача засобів індивідуального захисту слуху (навушники, вкладиші).
- 2) покращення освітлення – замінити лампи на світлодіодні з високою інтенсивністю, забезпечити додаткове локальне освітлення.
- 3) пиловловлювання – використовувати пиłosоси промислового типу, проводити регулярне зволоження повітря.
- 4) контроль температури – встановити кондиціонери або тепловентилятори для підтримки комфортного мікроклімату.
- 5) регулярний технічний огляд обладнання – планові перевірки та своєчасний ремонт для зменшення ризику травм.

Аналіз небезпечних чинників дозволив виявити основні проблеми та розробити заходи для покращення умов праці. Реалізація цих заходів знизить ризику для здоров'я працівників, підвищить їхню продуктивність та загальний рівень безпеки на виробництві.

4.2. Розробка логічно-імітаційної моделі травматизму під час монтажу інформаційної системи

Проаналізувавши кожну із логічних моделей процесів формування та можливого виникнення травмонебезпечних та аварійних ситуацій, завжди можна знайти подію з якої починається небезпечний процес ще до виникнення небезпечних наслідків.

Методикою оцінки рівня небезпеки робочих місць, машин, виробничих процесів та окремих виробництв передбачено пошук об'єктивного критерію рівня небезпеки для конкретного об'єкта. Таким показником вибрана ймовірність виникнення аварії, травми залежно від досліджуваного явища.

Для оцінки рівня небезпеки певного об'єкта чи явища можна застосувати метод обчислення ймовірності виникнення будь-якого випадкового явища, який широко застосовують в зарубіжній інженерній практиці. Основні його принципи полягають в тому, що на основі обстеження робочого місця чи окремої машини виявляють виробничі небезпеки, можливі аварійні або травматичні ситуації. При оцінці ситуацій визначають події, які можуть стати головною подією при побудові логічно-імітаційної моделі травми. Після цього будують модель «дерева відмов і помилок оператора». При цьому важливе значення має правильний вибір головної події.

Головну подію (травма), модель якої нам необхідно побудувати, вибирають виходячи з оцінки відповідного об'єкта, виробництва чи окремої одиниці обладнання і змісту його найбільш небезпечного явища, яке за певних умов виробництва може виникнути.

Після вибору головного випадкового явища (події) розпочинаємо побудову моделі («дерева»). Використовуючи оператора «і» та «або», використовуємо набір ситуацій (відомих до цього), які можуть призвести до подій, вибраної як головна.

Після визначення відповідних травмонебезпечних ситуацій та їх кількості, визначаємо інші події, що входять до кожної такої ситуації, логічним аналізом із застосуванням операторів «і», «або» та інших. Процес побудови моделі триває, поки не будуть знайдені усі базові події, що визначають межу моделі.

Слід мати на увазі, що кожна випадкова подія, до якої входять базові події, може формуватися й виникати при входженні у неї двох, трьох і більше базових подій за допомогою відповідних операторів.

Повністю побудована і перевірена модель підлягає математичній обробці для визначення ймовірності кожної випадкової події, що увійшла до моделі, починаючи з базових і закінчуючи головною.

Ймовірність базових подій визначаємо за даними виробництва. Наприклад, базова подія «стан контролю з охорони праці». Для визначення ймовірності ми повинні встановити, наскільки (у відсотках) від ідеального рівня здійснюється відповідний контроль на об'єкті. Якщо буде встановлено, що такий рівень контролю становить 50% або 30%, то ймовірність відповідно дорівнює 0,5 і 0,3. При відсутності контролю ймовірність «не здійснення контролю» становитиме 1, якщо контроль ідеальний, то відповідно ймовірність дорівнює 0.

Після обчислення ймовірності всіх подій, розміщених у ромбах, і базових подій, починаючи з лівої нижньої гілки «дерева», позначаємо номерами всі випадкові події, що увійшли до моделі.

На цьому можна вважати, що певна модель підготовлена до математичних обчислень ймовірностей випадкових подій логічно-імітаційної моделі

Отже, для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми під час монтажу інформаційної системи прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей складемо список базових подій. Вони лежатимуть у основі даної моделі. Кожному пункту списку присвоюємо певне значення ймовірності виникнення. Нижче подано сам список:

- | | |
|--|---------------------------------|
| 1. Стан контролю з охорони праці | $P_1 = 0,2;$ |
| 2. Несерйозне відношення до проходження ТО інструменту | $P_2 = 0,1;$ |
| 3. Відсутність комплектуючих установки..... | $P_3 = 0,2;$ |
| 4. Невисока міцність | $P_4 = 0,03;$ |

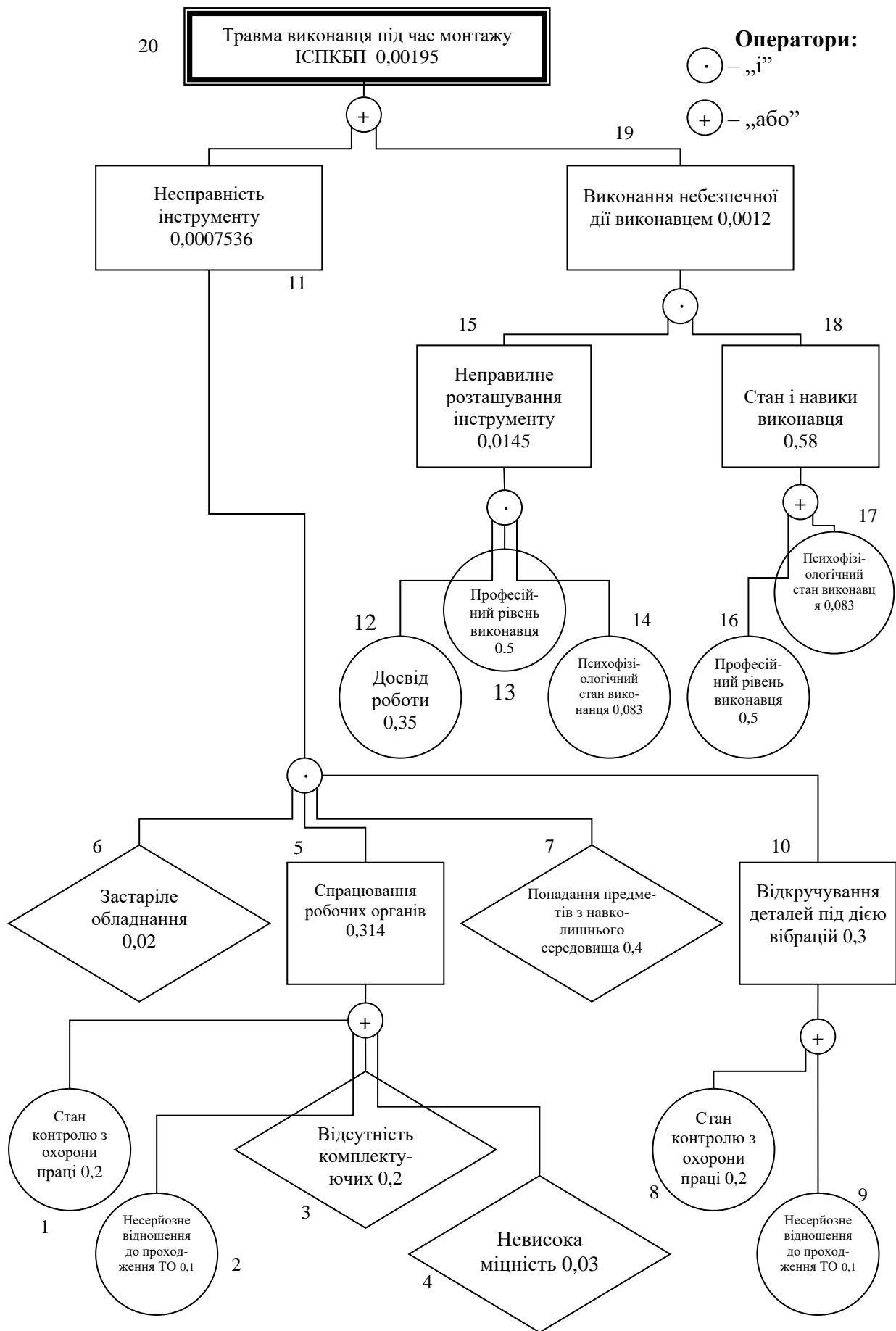


Рис. 4.1. Логіко-імітаційна модель процесу формування та виникнення аварії та травми під час монтажу інформаційної системи прогнозування конкурсного балу соціальних проєктів розвитку громад

- | | |
|---|-------------------|
| 1. Використання застарілого обладнання..... | $P_6 = 0,02;$ |
| 2. Попадання сторонніх предметів | $P_7 = 0,4;$ |
| 3. Досвід роботи виконавця | $P_{12} = 0,35.$ |
| 4. Професійний рівень виконавця | $P_{13} = 0,5;$ |
| 5. Психофізіологічний стан виконавця..... | $P_{14} = 0,083;$ |

На основі даного списку будуємо матрицю логічних взаємозв'язків між окремими пунктами, графічне представлення якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що входять у дану логіко-імітаційну модель процесу монтажу інформаційної системи прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережевих моделей (на прикладі ймовірності отримання травми виконавця).

Ймовірність виникнення події P_5 визначаємо наступним чином:

$$P_5 = 0,2 + 0,1 + 0,2 + 0,003 - 0,2 \cdot 0,1 - 0,2 \cdot 0,03 - 0,2 \cdot 0,03 - 0,1 \cdot 0,2 - 0,1 \cdot 0,03 - 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,1 \cdot 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,2 \cdot 0,1 \cdot 0,03 - 0,2 \cdot 0,1 \cdot 0,2 \cdot 0,03 = 0,314$$

Ймовірність виникнення події P_{10} визначаємо так:

$$P_{10} = 0,2 + 0,1 = 0,3.$$

Ймовірність виникнення події P_{11} визначаємо:

$$P_{11} = 0,02 \cdot 0,314 \cdot 0,4 \cdot 0,3 = 0,00075.$$

Ймовірність виникнення події P_{15} визначаємо наступним чином:

$$P_{15} = 0,35 \cdot 0,5 \cdot 0,083 = 0,0145.$$

Ймовірність події P_{18} :

$$P_{18} = 0,5 + 0,083 = 0,58.$$

Ймовірність події P_{19} :

$$P_{19} = 0,0145 \cdot 0,083 = 0,0012.$$

Ймовірність події P_{20} :

$$P_{20} = 0,00075 + 0,0012 = 0,00195.$$

Ймовірність травми рівна ймовірності виникнення аварії, бо остання можлива лише за умови монтажу інформаційної системи прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережевих моделей людиною.

Логіко-імітаційні моделі аварій і травм допомагають зменшити ймовірність виникнення аварійних та травмонебезпечних ситуацій. Якщо необхідно оцінити рівень небезпеки будь-якого робочого місця, слід уважно вивчити і побудувати логічні моделі можливих небезпечних ситуацій, які охоплюють як стан обладнання і самого робочого місця, так і поведінку працюючого і обчислити ймовірність виникнення травми.

Після аналізу результатів моделювання ймовірність виникнення травми можна звести до дуже малої величини – достатньо зменшити вплив ймовірностей вихідних факторів, які до неї призводять.

4.3. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Надзвичайні ситуації (НС) виникають через природні катаклізми, техногенні аварії або соціальні конфлікти. Важливим елементом підготовки до НС є розробка заходів, які забезпечують мінімізацію ризиків для здоров'я та життя працівників, збереження матеріальних цінностей і швидке відновлення нормального функціонування підприємства.

Заходи щодо захисту цивільного населення плануються проводяться по населених пунктах де розміщені підприємства і охоплюють населення навколишніх сіл. Водночас характер та зміст захисних засобів встановлюються від ступеня загрози, місцевих умов з урахуванням важливості виробництва для безпеки населення і інших економічних і соціальних чинників.

Основні заходи щодо захисту населення плануються та здійснюються завчасно і мають випереджувальний характер, це стосується насамперед підготовки, підтримання у постійній готовності індивідуальних та колективних

засобів захисту, їх накопичення, а також підготовки до проведення евакуації населення із зон підвищеного ризику.

Також раз в три роки проводяться навчання по підготовці близьких до військових дій, що в разі небезпеки могло би не дістати людину зненацька. Керівництво докладає максимум зусиль, щоб працівники підприємств були хоча би мінімально захищенні в разі будь-якої небезпеки пов'язаної з тими чи іншими обставинами.

РОЗДІЛ 5.
ВИЗНАЧЕННЯ ПОКАЗНИКІВ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД
ПРОГНОЗУВАННЯ КОНКУРСНОГО БАЛУ СОЦІАЛЬНИХ
ПРОЄКТІВ РОЗВИТКУ ГРОМАД НА ОСНОВІ
НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ

Економічна ефективність прогнозування оцінюється через низку показників. Зокрема приріст ефективності фінансування E_f визначається за формулою:

$$E_f = \frac{\Delta B}{\Delta C}, \quad (5.1)$$

де ΔB – приріст вигод від реалізації проектів завдяки прогнозуванню; ΔC – витрати на розробку та впровадження прогнозних моделей.

Ефективність використання бюджетних коштів E_b визначається за формулою:

$$E_b = \frac{N_{yc}}{N_3}, \quad (5.2)$$

де N_{yc} – кількість успішних проектів з виконаним прогнозом, од; N_3 – загальна кількість проектів, од.

Рентабельність інвестицій ROI визначається за формулою:

$$ROI = \frac{\Delta B - \Delta C}{\Delta C} \times 100\%. \quad (5.3)$$

Скорочення витрат завдяки оптимізації S_c визначається за формулою:

$$S_c = \frac{\Delta C_{on}}{\Delta C_{bn}} \times 100\%. \quad (5.4)$$

де ΔC_{on} – витрати на реалізацію проектів з використанням прогнозування; ΔC_{bn} – витрати без прогнозування.

Виконаємо розрахунок показників економічної ефективності із за наступними даними. Загальні вигоди від проектів без прогнозування становлять

$B_{\bar{on}}=10000$ тис. грн. Загальні вигоди з прогнозування становлять $B_{np}=12500$ тис. грн. Витрати на реалізацію проектів без прогнозування $C_{\bar{on}}=8000$ тис. грн. Витрати з використанням прогнозування $C_{np}=7500$ тис. грн. Загальна кількість проектів становить 100 од. Успішно реалізованих проектів з прогнозуванням наявно 80 од.

Підставивши відповідні значення у формулу (5.1) отримаємо приріст ефективності фінансування E_f :

$$\Delta B = B_{np} - B_{\bar{on}} = 12500 - 10000 = 2500 \text{ тис.грн.}$$

$$\Delta C = C_{\bar{on}} - C_{np} = 8000 - 7500 = 500 \text{ тис.грн.}$$

$$E_f = \frac{\Delta B}{\Delta C} = \frac{2500}{500} = 5.0.$$

Підставивши відповідні значення у формулу (5.2) отримаємо ефективність використання бюджетних коштів E_b :

$$E_b = \frac{N_{yc}}{N_3} = \frac{80}{100} = 0.8(80\%).$$

Підставивши відповідні значення у формулу (5.3) отримаємо рентабельність інвестицій ROI :

$$ROI = \frac{\Delta B - \Delta C}{\Delta C} \times 100\% = \frac{2500 - 500}{500} \times 100\% = 400\%.$$

Підставивши відповідні значення у формулу (5.4) отримаємо скорочення витрат завдяки оптимізації S_c :

$$S_c = \frac{\Delta C_{\bar{on}} - \Delta C_{np}}{\Delta C_{\bar{on}}} \times 100\% = \frac{8000 - 7500}{8000} \times 100\% = 6.25\%.$$

Отримані результати розрахунків показників економічної ефективності від прогнозування конкурсного балу соціальних проектів розвитку громад на основі нейромережових моделей подано у таблиці 5.1.

Отримані результати показують, що використання прогнозних моделей дозволяє значно підвищити ефективність фінансування соціальних проектів.

Таблиця 5.1 – Результати розрахунків показників економічної ефективності від прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей

Показник	Одиниця виміру	Значення
Загальні вигоди від проєктів без прогнозування становлять $V_{\text{он}}$	тис. грн.	10000
Загальні вигоди з прогнозуванням становлять $V_{\text{пр}}$	тис. грн.	12500
Витрати на реалізацію проєктів без прогнозування $C_{\text{он}}$	тис. грн.	8000
Витрати з використанням прогнозування $C_{\text{пр}}$	тис. грн.	7500
Загальна кількість проєктів становить N_3	100	од.
Успішно реалізованих проєктів з прогнозуванням $N_{\text{ус}}$	80	од.
Приріст ефективності фінансування (E_f)	–	5.0
Ефективність використання бюджетних коштів (E_b)	%	80
Рентабельність інвестицій (ROI)	%	400
Скорочення витрат завдяки оптимізації (S_c)	%	6.25

Зокрема, приріст вигод $E_f = 5.0$ вказує на суттєвий економічний ефект від використання прогнозів, а ROI на рівні 400% свідчить про високу рентабельність інвестицій. Ефективність використання бюджетних коштів $E_b = 80\%$ демонструє, що більшість проєктів реалізовано успішно завдяки прогнозуванню. Скорочення витрат на $S_c = 6.25\%$ підтверджує доцільність застосування моделі.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Використання нейромережевих моделей для прогнозування проєктів конкурсного балу соціальних пропозицій значно полегшує процес відбору, зменшує суб'єктивний вплив і забезпечує справедливий розподіл фінансування між найбільш перспективними ініціативами. Розробка такої моделі є актуальним напрямком досліджень, вона здатна покращити процеси прийняття рішень у галузі соціального розвитку громад.

Застосування машинного навчання та нейронних мереж у соціальних сферах відкриває нові можливості для підвищення ефективності вирішення суспільно значущих проблем. Завдяки автоматизації процесів і персоналізації підходів, машинне навчання сприяє більш точному визначенню потреб суспільства, що дозволяє значно покращити якість послуг і забезпечити рівний доступ до них для різних верств населення. Такі інновації сприяють сталому розвитку соціальної інфраструктури та підвищують загальний рівень добробуту.

Аналіз наукових праць у досліджуваному напрямі свідчить про значний інтерес до впровадження інноваційних технологій у різних галузях науки і практики. Незважаючи на значні досягнення, окремі аспекти, такі як адаптивність моделей до динамічних змін середовища та інтеграція методів у комплексні системи, потребують подальшого опрацювання. Це підкреслює актуальність і перспективність подальших досліджень у зазначеному напрямі.

Розробка концепції побудови нейромережевих моделей є багатоетапним процесом, що включає аналіз задачі, підготовку вибору даних, оптимальної архітектури та тестування. Правильний підхід до реалізації кожного з етапів дозволяє отримати точну, адаптовану до потреби користувача модель, здатну прогнозувати конкурсний бал соціальних проєктів із високою точністю.

Для побудови нейромережевих моделей прогнозування конкурсного балу соціальних проєктів розвитку громад Львівської області було зібрано дані про результати проведених конкурсів проєктів місцевих ініціатив за період 2020-

2023 років. Ці дані включають показники реалізації соціальних проєктів, які фінансувалися із обласного бюджету. Загалом таблиця містить 16 атрибутів для 629 записів, які наведені у таблиці 2.1. Під час очищення даних не було виявлено помилок чи пропусків. Для нормалізації числових даних та покращення стабільності роботи нейронних мереж усі числові ознаки було масштабовано за допомогою StandardScaler.

Нами виконано аналіз взаємозв'язків між даними. На основі даних у стовпці Results, було побудовано кругову діаграму (рис. 2.4-2.5), яка відображає розподіл результатів проєктів. Встановлено, що 75,4% проєктів є переможцями конкурсів (Winning proposals), а 24,6% проєктів було відхилено (Rejected proposals). Такий високий відсоток успішних проєктів свідчить про значну конкуренцію між громадами, а також про високий рівень підготовки заявок.

Нами виконано розподілу бюджетів проєктів (рис. 2.6-2.7). Отриманий розподіл бюджетів проєктів соціальних (рис. 2.2) показує, що середнє значення бюджету проєкту становить 315000 грн. Стандартне відхилення бюджету складає 152980 грн, що вказує на значну варіативність фінансування проєктів. Більшість проєктів мають бюджети в діапазоні 250000 – 500000 грн, з помітним піком у межах 500000 грн. Ця інформація є корисною для визначення середніх тенденцій у фінансуванні проєктів.

На гістограмі (рис. 2.8) представлені 20 територіальних громад, які найчастіше брали участь у конкурсах. Найбільшу кількість проєктів подали громади: Львівська громада – 53 проєкти; Дрогобицька громада – 33 проєкти; Червоноградська – 23 проєкти. Ці громади проявляють високу активність у реалізації соціальних ініціатив, що може бути зумовлено їхніми організаційними можливостями та фінансовою спроможністю.

На основі кореляційної матриці (рис. 2.9) було виявлено ключові залежності між змінними. Загальний бюджет проєкту має високу кореляцію з такими показниками як обласний бюджет (Regional_budget) – 0.98 та частка позабюджетного фінансування (%_non-budgetary_contrib) – 0.43. Результати аналізу показали, що основними чинниками, які впливають на конкурсний бал,

є загальний бюджет проєкту, обласне фінансування та частка позабюджетного фінансування. Отримані залежності дозволяють зосередити увагу на цих показниках під час розробки нейромережових моделей прогнозування.

Застосування різних метрик дозволяє детально оцінити якість моделі з різних точок зору. У випадку задачі прогнозування конкурсного балу соціальних проєктів MAE та RMSE є основними метриками для оцінки помилок, оскільки вони відображають абсолютну точність та масштаб помилки. Значення R^2 використовується для загальної оцінки пояснювальної здатності моделі. Таким чином, поєднання цих метрик забезпечує комплексну оцінку ефективності нейромережових моделей.

Оптимізація є невід’ємною складовою процесу навчання нейромережових моделей. Для цієї задачі використовуються різні алгоритми оптимізації – стохастичний градієнтний спуск (SGD), Адаптивний градієнтний алгоритм (Adagrad), Адаптивний моментний алгоритм (Adam). Особливостями адаптивного моментного алгоритму (Adam) є те, що він комбінує переваги моментного градієнта і Adagrad. Він є стабільний у задачах з високою варіативністю даних. Використання оптимізатора AdamW (Adam з L2-регуляризацією), зменшує потребу переобладнання.

У нашій роботі розглянуто три основні архітектури нейромережових моделей, які використовуються для прогнозування конкурсного балу соціальних проєктів – Feedforward Neural Network (FNN), Recurrent Neural Network (RNN) та Convolutional Neural Network (CNN). Кожна з цих архітектур має свої переваги та використовується залежно від структури вхідних даних і задачі моделювання.

Архітектура Feedforward Neural Network є найпростішою архітектурою штучної нейронної мережі (рис. 3.1). Використовується для задач регресії та класифікації. Модель FNN обробляє лише табличні дані без часової залежності.

Архітектура Recurrent Neural Network є моделлю, яка враховує часову залежність у даних (рис. 3.2). Особливістю Recurrent Neural Network є те, що така архітектура підходить для обробки послідовних даних, таких як часові ряди.

Чутлива до проблеми затухання градієнтів, яку вирішують покращені архітектури (LSTM, GRU).

Архітектура Convolutional Neural Network найчастіше використовується для обробки зображень, але також застосовується для табличних даних у вигляді згорток. CNN витягує локальні залежності між ознаками, що забезпечує високу точність при моделюванні складних взаємозв'язків.

Нами написано код для побудови архітектури нейронної мережі Feedforward Neural Network (FNN), яка складається з трьох шарів – вхідного, двох прихованих і вихідного (рис. 3.4) Перший шар складається з 64 нейронів із цією функцією активації ReLU. Другий шар із 32 нейронів із тією ж функцією активації. Модель навчається протягом 50 епох з розміром батчу 16. Валідаційні дані можуть відстежувати точність під час навчання. Візуалізація навчання представлена на рис. 3.6. Графік демонструє зміну середньої абсолютної помилки (MAE) на тренувальних і валідаційних даних в епоху навчання. Отже, Feedforward Neural Network є базовою моделлю, яка показує стабільні результати в задачах прогнозування.

Нами написано код для побудови рекурентної нейронної мережі (RNN), яка являє собою тип нейронних мереж, які враховують взаємозв'язок між даними в часі або конфліктах. Побудова архітектури мережі передбачає один прийнятий шар із функцією активації ReLU (рис. 3.7). Модель має один прихований шар із 64 нейронами, які задають функцію активації ReLU. Вихідний шар містить один нейрон для прогнозування конкурсного балу (рис. 3.8). Використання RNN забезпечує точні результати, а її структура легко адаптується до інших завдань із тимчасовими або послідовними залежностями.

Нами написано код для побудови моделі CNN, що забезпечує перетворення даних у формат, придатний для згорткових операцій. У нашому випадку це тривимірний формат, де кожен приклад представляється як масив із формою (кількість ознак, 1) (рис. 3.10). Згортковий шар Conv1D(32, kernel_size=3) має 32 фільтри та ядро розміром 3, що виділяє локальні патерни. Прихований шар Dense(64) із 64 нейронами обробляє виділені особливості.

Вихідний шар Dense(1) із одним нейроном для прогнозування конкурсного балу. Для оптимізації ваг використовується оптимізатор adam, а функція втрат – mean_squared_error. Після навчання та оцінки моделі CNN було виявлено, що вона демонструє високі результати точності завдяки своїй здатності ефективно виділяти локальні особливості вхідних даних.

Модель CNN продемонструвала найкращі результати за всіма метриками, що робить її найкращим вибором для прогнозування конкурсних балів. FNN є конкурентоспроможною моделлю та може бути використана як альтернатива. RNN має слабкі показники точності, що недостатньо про необхідність її подальшої оптимізації або застосування інших підходів. Згідно з проведеним аналізом, CNN є найбільш ефективною моделлю для поставленої задачі прогнозування.

Для підвищення точності прогнозування та зниження помилок моделі Convolutional Neural Network (CNN) було застосовано кілька методів оптимізації, які включають регулярне додавання, використання покращеного функціоналу втрат, оптимізатора з меншою швидкістю навчання та більш тривалого навчання.

Покращена модель продемонструвала зниження MSE і MAE порівняно з базовою CNN. Використання Huber Loss зробило модель менш чутливою до викидів у дані. Регуляризація Dropout і L2 допомогли уникнути перенавчання, що видно з менших розривів між навчанням і валідацією. Значення R^2 для покращеної CNN зросло, що свідчить про більш точне моделювання залежностей.

Розроблені рекомендації з охорони праці сприяють забезпеченню безпечних умов роботи для фахівців, які займаються розробкою нейромережових моделей.

На підставі проведених розрахунків економічної ефективності від прогнозування конкурсного балу соціальних проєктів розвитку громад на основі нейромережових моделей встановлено, що приріст вигод $E_f = 5.0$ вказує на суттєвий економічний ефект від використання прогнозів. *ROI* становить на

рівні 400%, що свідчить про високу рентабельність інвестицій. Ефективність використання бюджетних коштів $E_b = 80\%$ демонструє, що більшість проектів реалізовано успішно завдяки прогнозуванню. Скорочення витрат на $S_c = 6.25\%$ підтверджує доцільність застосування моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Громадський простір. URL: <https://www.prostir.ua/> (дата звернення 12.10.2024).
2. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.
3. Карпа, Д. М., Цмоць, І. Г., & Опотяк, Ю. В. Нейромережеві засоби прогнозування споживання енергоресурсів. Науковий вісник НЛТУ України, 2018, 28(5), 140–146.
4. Кириленко, В., Шиманський, В., & Івасюк, Р. Інтелектуальна система прогнозування серцевої недостатності з використанням штучних нейронних мереж. Комп'ютерне моделювання та інформаційні технології. 2023, 104–106.
5. Класифікація в Python з Scikit-Learn та Pandas. URL: <https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/> (дата звернення: 14.09.2024).
6. Кулик, В. М., & Коротєєва, Т. О. Алгоритм прогнозування результатів футбольних матчів на основі нейронних мереж. Науковий вісник НЛТУ України, 2017, 27(9), 111–114.
7. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.
8. Мармуляк А., Маланчук О.М. Підхід до моніторингу процесу відбору проєктів підтримки освіти та медицини на території громад. Вісник Кременчуцького національного університету імені Михайла Остроградського, 2024, Вип. 1(144), 63-70.
9. Морозов, В., & Мезенцева, О. Використання навчальних нейронних мереж для прогнозування розвитку продукту ІТ-проєкту. Безпека інформаційних систем і технологій, 2020, (4), 3–10.

10. Павлиш В. А., Гліненко Л. К., Шаховська Н. Б. Основи інформаційних технологій і систем: підручник. Львів: Львівська політехніка, 2018. 620 с.

11. Ситнік Б.Т. Основи інформаційних систем і технологій: навч. посіб. Харків: УкрДУЗТ, 2018. 130 с.

12. Спільнота медіа рішень. URL: <https://spilnota.com.ua/> (дата звернення 11.08.2024).

13. Територіальна громада України. Вікіпедія. URL: <http://surl.li/kburen> (дата звернення 05.10.2024).

14. Тригуба А. М., Маланчук О. М., Мармуляк А. С., Паньків О. В., Шолудько Р. Я. Алгоритм та програмні модулі моніторингу процесу відбору соціальних проєктів із використанням веб-парсингу. Вчені Львівського національного аграрного університету виробництву: каталог інноваційних розробок; за заг. ред. В. І. Лопушняка, Б. І. Гулька. Вип. 24. Львів: Львів. нац. ун-т. природ., 2024. С. 27.

15. Тригуба А., Маланчук О., Тригуба І., Мармуляк А., Демчина В. Андрушків О., Олійник Р. Вплив сучасних інформаційних технологій на процеси ініціації та планування проєктів розвитку громад та регіонів. Вісник Львівського національного університету природокористування: агроінженерні дослідження. №24. Львів: Львів НУП, 2024. С. 123-130.

16. Тригуба А.М., Мармуляк А., Маланчук О.М., Придатко О.В. Модель та програмні модулі для моніторингу процесу відбору соціальних проєктів розвитку громад. Вісник Львівського державного університету безпеки життєдіяльності, 2024, 29, 152-167.

17. Тригуба А.М., Мармуляк А., Маланчук О.М., Тригуба І.Л. Вплив цифрової трансформації громад на ініціацію та планування соціальних проєктів. Управління розвитком складних систем. 2024, 57, 130-136.

18. Тригуба А.М., Тригуба І.Л., Мармуляк А.С., Маланчук О.М. Моніторингу відбору соціальних проєктів із використанням веб-парсингу. Управління проєктами у розвитку суспільства: Управління проєктами

післявоєнної розбудови України: тези доп. XXI -ї Міжн. конф., Київ: КНУБА, 2024. С.233-237.

19. Федоришин, О. В. Моделювання інтелектуальної системи класифікаційного прогнозування успішності соціальних проєктів. Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції. 2014, 176 с.

20. Федоришин, О. В., Карпуша, В. Д., & Теліженко, О. М. Інтелектуальна система класифікаційного прогнозування успішності соціальних проєктів. Вісник Сумського державного університету. Серія Економіка, 2014, (2), 63–71.

21. Чайковська, М., & Шкеда, О. Алгоритм машинного навчання штучної нейронної мережі для побудови моделі прийняття управлінських рішень при розробці маркетингової стратегії. Маркетинг і цифрові технології, 2023, 7(2), 137–146.

22. Almeida, J., & Silva, P. Predicting social project success using artificial neural networks. *Journal of Community Development*, 2020, 55(3), 345-360.

23. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In *Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon)*, Tallinn, Estonia, 30 May–1 June 2018; pp. 371–390.

24. Bhamare, D.; Salman, T.; Samaka, M.; Erbad, A.; Jain, R. Feasibility of supervised machine learning for cloud security. In *Proceedings of the 2016 International Conference on Information Science and Security (ICISS)*, Pattaya, Thailand, 19–22 December 2016; pp. 1–5.

25. Brown, L., & Green, M. Application of machine learning in evaluating community development projects. *International Journal of Social Research Methodology*, 2019, 22(4), 467-482.

26. Chen, Y., & Zhang, X. Neural network models for assessing the impact of social initiatives in urban areas. *Urban Studies Journal*, 2021, 58(7), 1345-1360.

27. Davis, R., & Thompson, S. Machine learning approaches to predict funding outcomes for community projects. *Nonprofit Management & Leadership*, 2020, 30(2), 213-229.
28. Dionísio, N.; Alves, F.; Ferreira, P.M.; Bessani, A. Cyberthreat detection from twitter using deep neural networks. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 14–19 July 2019; pp. 1–8.
29. Dong, C. The Evolution of Machine Learning. *TechCrunch*, 18 August 2018.
30. Evans, K., & Williams, J. Evaluating the effectiveness of social programs using deep learning techniques. *Social Science Computer Review*, 2019, 37(5), 599-614.
31. Garcia, M., & Lopez, R. Predictive analytics in social project management: A neural network approach. *Project Management Journal*, 2021, 52(1), 45-60.
32. Gkotsis, G.; Oellrich, A.; Velupillai, S.; Liakata, M.; Hubbard, T.J.; Dobson, R.J.; Dutta, R. Characterisation of mental health conditions in social media using Informed Deep Learning. *Sci. Rep.* 2017, 7, 45141.
33. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
34. Hernandez, L., & Martinez, P. Assessing community project proposals with machine learning models. *Community Development Journal*, 2020, 55(2), 225-240.
35. Jayatilake, S.M.D.A.C.; Ganegoda, G.U. Involvement of machine learning tools in healthcare decision making. *J. Healthc. Eng.* 2021, 2021, 6679512.
36. Jetha, A.; Shamaee, A.; Bonaccio, S.; Gignac, M.A.; Tucker, L.B.; Tompa, E.; Bültmann, U.; Norman, C.D.; Banks, C.G.; Smith, P.M. Fragmentation in the future of work: A horizon scan examining the impact of the changing nature of work on workers experiencing vulnerability. *Am. J. Ind. Med.* 2021, 64, 649–666.

37. Johnson, D., & Smith, A. Utilizing artificial intelligence to forecast social project success rates. *AI & Society*, 2019, 34(3), 567-580.
38. Kaliyar, R.K.; Goswami, A.; Narang, P. DeepFakE: Improving fake news detection using tensor decomposition-based deep neural network. *J. Supercomput.* 2021, 77, 1015–1037.
39. Karayığit, H.; İnan Acı, Ç.; Akdağlı, A. Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods. *Expert Syst. Appl.* 2021, 174, 114802.
40. Kim, S., & Lee, J. Application of deep learning in evaluating community development initiatives. *Journal of Urban Affairs*, 2021, 43(4), 567-583.
41. Kitchin, R. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*; Sage: Thousand Oaks, CA, USA, 2014. [Google Scholar]
42. Komatsu, H.; Watanabe, E.; Fukuchi, M. Psychiatric neural networks and precision therapeutics by machine learning. *Biomedicines* 2021, 9, 403.
43. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Rudynets M., Grabovets V., Onyshchuk V. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. *Proceedings of the 3rd International Workshop on Modern Machine Learning Technologies and Data Science Workshop. Proc. 3rd International Workshop (MoMLeT&DS 2021)*. 2021; I, 96-206.
44. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, 521, 436–444.
45. Malekian, A.; Chitsaz, N. Concepts, procedures, and applications of artificial neural network models in streamflow forecasting. In *Advances in Streamflow Forecasting*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 115–147.
46. Nguyen, T., & Tran, H. Predicting the success of social projects using machine learning algorithms. *Social Indicators Research*, 2020, 150(2), 511-526.
47. Nilsson, N.J. *The Quest for Artificial Intelligence*; Cambridge University Press: Cambridge, UK, 2009.

48. Pramod, A.; Naicker, H.S.; Tyagi, A.K. Machine learning and deep learning: Open issues and future research directions for the next 10 years. In *Computational Analysis and Deep Learning for Medical Care: Principles, Methods, and Applications*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2021; pp. 463–490.

49. Russell, S.J. *Artificial Intelligence A Modern Approach*; Pearson Education, Inc.: London, UK, 2010.

50. Song, T.; Ding, W.; Liu, H.; Wu, J.; Zhou, H.; Chu, J. Uncertainty quantification in machine learning modeling for multi-step time series forecasting: Example of recurrent neural networks in discharge simulations. *Water* 2020, 12, 912.

51. Taherdoost, H. An Overview of Trends in Information Systems: Emerging Technologies that Transform the Information Technology Industry. *Cloud Comput. Data Sci.* 2023, 4, 1–16.

52. Taherdoost, H.; Madanchian, M. Artificial Intelligence and Sentiment Analysis: A Review in Competitive Research. *Computers* 2023, 12, 37.

53. Tryhuba A., Malanchuk O., Tryhuba I., Marmulyak, A. Decision support system for initiating projects of medical and social development in regions. *Proceedings of the 5nd International Workshop IT Project Management (ITPM 2024)*. Conference. Bratislava, Slovakia, May 22, 2024. pp.204-218.

54. Tryhuba A., Tryhuba I., Malanchuk O., Marmulyak, A. A deep neural network model for predicting the competitive score of social projects for community development. *Proceedings of the 6th International Workshop on Modern Machine Learning Technologies (MoMLLeT-2024)*. Vol. 3711. Lviv, Ukraine, May 31 - June 1, 2024. pp.55-74.

55. 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. *CEUR Workshop Proceedings*. 2021; 2631: 327–340.

56. 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, 44–52.

57. 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: 55–58.

58. Tryhuba, A., Vovk, M., Batyuk, B., Holomsha, O., Sava, A. Improving The Quality Of Management In The System Of Forecasting Milk Procurement In Communities Usage The Technology Of Neutron Networks. Journal of Hygienic Engineering and Design. 2022; 40, 201–209.

59. Tsai, S.-B.; Ma, H. A research on preparation and application of the monolithic catalyst with interconnecting pore structure. Sci. Rep. 2018, 8, 16605.

60. Veale, M.; Binns, R. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. Big Data Soc. 2017, 4, 2053951717743530.

61. Yalur, T. Interperforming in AI: Question of ‘natural’ in machine learning and recurrent neural networks. AI Soc. 2020, 35, 737–745.

62. Yavanoglu, O.; Aydos, M. A review on cyber security datasets for machine learning algorithms. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Beijing, China, 10–12 March 2017; pp. 2186–2193.

63. Zhang, H.; Li, Y.; Zhang, H. Risk early warning safety model for sports events based on back propagation neural network machine learning. Saf. Sci. 2019, 118, 332–336.

64. Zhang, J.; Williams, S.O.; Wang, H. Intelligent computing system based on pattern recognition and data mining algorithms. Sustain. Comput. Inform. Syst. 2018, 20, 192–202.

ДОДАТКИ

Додаток А.1

Код створення моделей штучних нейронних мереж

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, SimpleRNN, Conv1D, Flatten
from tensorflow.keras.utils import plot_model

# Завантаження даних та розділення на ознаки та цільову змінну
X = df.drop('Final_score', axis=1).values
y = df['Final_score'].values

# Масштабування ознак
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Розділення на тренувальний та тестовий набори даних
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Функція для обчислення метрик
def evaluate_model(y_test, y_pred, model_name):
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    rmse = np.sqrt(mse)
    inv_rmse = 1 / rmse if rmse != 0 else 0
    print(f"\nModel: {model_name}")
    print("Mean Squared Error (MSE):", mse)
    print("Mean Absolute Error (MAE):", mae)
    print("R^2 Score:", r2)
    print("Root Mean Squared Error (RMSE):", rmse)
    print("Inverse RMSE:", inv_rmse)
    return mse, mae, r2, rmse, inv_rmse

# 1. Feedforward Neural Network (FNN)
model_fnn = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(1)
])
model_fnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_fnn = model_fnn.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50,
batch_size=16, verbose=0)
y_pred_fnn = model_fnn.predict(X_test).flatten()
metrics_fnn = evaluate_model(y_test, y_pred_fnn, "FNN")

# Вивід інформації про модель FNN

```

```

print("\nFNN Model Summary:")
model_fnn.summary()

# Схема моделі FNN
plot_model(model_fnn, to_file='model_fnn.png', show_shapes=True, show_layer_names=True)

# 2. Recurrent Neural Network (RNN)
X_train_rnn = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test_rnn = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
model_rnn = Sequential([
    SimpleRNN(64, activation='relu', input_shape=(X_train_rnn.shape[1], X_train_rnn.shape[2])),
    Dense(1)
])
model_rnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_rnn = model_rnn.fit(X_train_rnn, y_train, validation_data=(X_test_rnn, y_test), epochs=50,
batch_size=16, verbose=0)
y_pred_rnn = model_rnn.predict(X_test_rnn).flatten()
metrics_rnn = evaluate_model(y_test, y_pred_rnn, "RNN")

# Вивід інформації про модель RNN
print("\nRNN Model Summary:")
model_rnn.summary()

# Схема моделі RNN
plot_model(model_rnn, to_file='model_rnn.png', show_shapes=True, show_layer_names=True)

# 3. Convolutional Neural Network (CNN)
X_train_cnn = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test_cnn = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
model_cnn = Sequential([
    Conv1D(32, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1], 1)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(1)
])
model_cnn.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
history_cnn = model_cnn.fit(X_train_cnn, y_train, validation_data=(X_test_cnn, y_test),
epochs=50, batch_size=16, verbose=0)
y_pred_cnn = model_cnn.predict(X_test_cnn).flatten()
metrics_cnn = evaluate_model(y_test, y_pred_cnn, "CNN")

# Вивід інформації про модель CNN
print("\nCNN Model Summary:")
model_cnn.summary()

# Схема моделі CNN
plot_model(model_cnn, to_file='model_cnn.png', show_shapes=True, show_layer_names=True)

# Графіки навчання
def plot_training(history, title):
    plt.figure(figsize=(10, 6))
    plt.plot(history.history['mae'], label='Train MAE', color='blue')

```

```
plt.plot(history.history['val_mae'], label='Validation MAE', color='orange')
plt.title(title)
plt.xlabel('Epoch')
plt.ylabel('Mean Absolute Error')
plt.legend()
plt.grid(alpha=0.5)
plt.show()

plot_training(history_fnn, "FNN Training vs Validation MAE")
plot_training(history_rnn, "RNN Training vs Validation MAE")
plot_training(history_cnn, "CNN Training vs Validation MAE")

# Збір метрик для порівняння моделей
models = ['FNN', 'RNN', 'CNN']
metrics = ['MSE', 'MAE', 'R^2', 'RMSE', 'Inv RMSE']
values = [metrics_fnn, metrics_rnn, metrics_cnn]

# Побудова графіків точності моделей
for i, metric in enumerate(metrics):
    plt.figure(figsize=(10, 6))
    plt.bar(models, [v[i] for v in values], color=['blue', 'green', 'purple'], alpha=0.7)
    plt.title(f'Comparison of Models by {metric}')
    plt.xlabel('Models')
    plt.ylabel(metric)
    plt.grid(axis='y', alpha=0.5)
    plt.tight_layout()
    plt.show()
```

Додаток А.2

Код оптимізації архітектури CNN

```

from tensorflow.keras.losses import Huber

# Поліпшена модель CNN
model_cnn_improved = Sequential([
    Conv1D(64, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1], 1)),
    Dropout(0.3), # Dropout для регуляризації
    Conv1D(32, kernel_size=3, activation='relu'),
    Flatten(),
    Dense(64, activation='relu', kernel_regularizer=l2(0.01)), # L2 регуляризація
    Dense(1)
])

# Оптимізатор Adam з меншою швидкістю навчання
optimizer = Adam(learning_rate=0.001)

# Використання Huber Loss
loss_function = Huber(delta=1.0) # delta можна налаштувати залежно від проблеми

model_cnn_improved.compile(optimizer=optimizer, loss=loss_function, metrics=['mae'])

# Навчання моделі
history_cnn_improved = model_cnn_improved.fit(
    X_train_cnn, y_train,
    validation_data=(X_test_cnn, y_test),
    epochs=100,
    batch_size=16,
    verbose=1
)

# Прогнозування
y_pred_cnn_improved = model_cnn_improved.predict(X_test_cnn).flatten()

# Оцінка покращеної моделі
metrics_cnn_improved = evaluate_model(y_test, y_pred_cnn_improved, "Improved CNN")

# Графік навчання покращеної моделі
plot_training(history_cnn_improved, "Improved CNN Training vs Validation MAE")

from tensorflow.keras.utils import plot_model
from IPython.display import Image

# Виведення схеми моделі CNN у Jupyter Notebook
plot_model(model_cnn_improved, to_file='model_cnn.png', show_shapes=True,
show_layer_names=True, dpi=100)

# Відображення зображення в Jupyter
Image(filename='model_cnn.png')

```