

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

КУБИК Лілія Ярославівна

**Нейромережеві методи опрацювання даних
експериментальних досліджень / Neural Network Methods
for Experimental Research Data Processing**

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

Кваліфікаційна робота

Виконала студентка групи
КНзм-21
Л.Я. Кубик

Науковий керівник:
к.т.н., професор В.В. Кочан

Кваліфікаційну роботу
допущено до захисту:

«___» _____ 20___ р.

В.о. завідувача кафедри

_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ - 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Ступінь вищої освіти «магістр»
спеціальність: 122 – «Комп'ютерні науки»
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
« ____ » _____ 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

КУБИК Лілії Ярославівні

(прізвище, ім'я, по батькові)

1. Тема роботи

**Нейромережеві методи опрацювання даних експериментальних досліджень
/ Neural Network Methods for Experimental Research Data Processing**

керівник роботи к.т.н., професор В.В. Кочан

затверджені наказом по університету від 20 грудня 2024 року № 938.

2. Строк подання студентом кваліфікаційної роботи 1 грудня 2025 р.

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

4. Основні питання, які потрібно розробити

- провести аналіз штучного інтелекту та штучних нейронних мереж;
- провести аналіз нейромережевих методів прогнозування часових рядів;
- провести аналіз методів підвищення точності нейромережевого прогнозу;
- провести дослідження можливості підвищення точності прогнозу за рахунок штучного збільшення навчальної вибірки прогнозуючої ШНМ;
- провести аналіз отриманих результатів.

5. Перелік графічного матеріалу у роботі

- архітектури нейронних мереж MLP та LSTM;
- результати прогнозування кількості вступників за допомогою стандартних функцій активації;
- результати нейромережевого прогнозування при штучному збільшенні кількості даних.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 20.05.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 28.10. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 11.11.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 25.11.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту.	до 1.12.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 04.12.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.12. 2025 р.	

Студентка _____ Л.Я. Кубик
(підпис)Керівник роботи _____ к.т.н., професор В.В. Кочан
(підпис)

РЕЗЮМЕ

Кваліфікаційна робота на тему «Нейромережеві методи опрацювання даних експериментальних досліджень» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом 84 сторінки, 17 ілюстрацій, 9 таблиць, 3 додатки, 45 використаних джерела.

Метою кваліфікаційної роботи є підвищення точності прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень.

Методи досліджень: узагальнення інформації з літературних джерел; нейромережеві методи (архітектури MLP та LSTM) для прогнозування часових рядів; метод регресійного аналізу (лінійний, степеневий, експоненціальний, поліноміальний) для апроксимації експериментальних даних; методика штучного збільшення навчальної вибірки; статистичні методи оцінки якості моделей.

Результати дослідження: запропоновано методику підвищення точності прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень, яка полягає у формуванні розширеної навчальної вибірки шляхом попередньої апроксимації даних методом регресійного аналізу. Встановлено, що використання стандартних функцій активації на малих вибірках призводить до втрати узагальнюючих властивостей мережі. Доведено, що найкращу ефективність для апроксимації та подальшого навчання LSTM-мережі забезпечують квадратична та кубічна моделі, що дозволило отримати стійкий прогноз.

Результати роботи можна використовувати для вирішення задач прогнозування динаміки процесів в умовах дефіциту вхідних даних, економії ресурсів і часу при проведенні високовартісних або тривалих експериментальних досліджень.

Ключові слова: ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, ЧАСОВІ РЯДИ, ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ПРОГНОЗ, ПОХИБКА.

ABSTRACT

Qualification work on the topic «Neural Network Methods for Experimental Research Data Processing» for the Master's degree in specialty 122 «Computer Science», educational and professional program «Computer Science», is written on 84 pages and contains 17 figures, 9 tables, 3 appendices, and 45 sources.

The purpose of the qualification work is to increase the accuracy of time series forecasting with a limited number of experimental research results.

Research methods: generalization of information from literary sources; neural network methods (MLP and LSTM architectures) for time series forecasting; regression analysis method (linear, power, exponential, polynomial) for approximating experimental data; method of artificially increasing the training sample; statistical methods for assessing the quality of models.

Research results: a method for increasing the accuracy of time series forecasting with a limited number of experimental research results is proposed, which consists in forming an expanded training sample by preliminary data approximation using the regression analysis method. It is established that the use of standard activation functions on small samples leads to the loss of generalizing properties of the network. It is proved that the best efficiency for approximation and further training of the LSTM network is provided by the quadratic and cubic models, which allowed obtaining a stable forecast.

The results of the work can be used to solve the problems of forecasting the dynamics of processes in conditions of input data shortage, saving resources and time when conducting high-cost or long-term experimental research.

Keywords: ARTIFICIAL NEURAL NETWORKS, TIME SERIES, EXPERIMENTAL DATA, FORECAST, ERROR.

ЗМІСТ

Перелік умовних позначень.....	8
Вступ	9
1 Теоретичні основи нейронних мереж	12
1.1 Базові поняття нейронних мереж та систем штучного інтелекту	12
1.2 Базовий елемент штучних нейронних мереж	19
1.3 Функції активації штучних нейронів.....	24
1.4 Постановка задачі дослідження.....	26
Висновки до розділу 1.....	27
2 Нейромережеві методи прогнозування часових рядів.....	28
2.1 Типові архітектури нейронних мереж.....	28
2.2 Модель MLP для прогнозування часових рядів нейромережевими методами	37
2.3 Модель LSTM для прогнозування часових рядів нейромережевими методами	40
Висновки до розділу 2.....	44
3 Дослідження можливості підвищення точності нейромережевого прогнозу	45
3.1 Методика нейромережевого прогнозу при обмеженому обсязі.....	45
3.2 Підготовка вхідних даних та вибір інструментальних засобів.....	47
3.3 Прогноз кількості осіб, які будуть зараховані, за допомогою ШНМ з функцією активації гіперболічний тангенс	49
3.4 Прогноз кількості осіб, які будуть зараховані, за допомогою ШНМ з функцією активації лінійних елементів виправлення	50
3.5 Прогноз кількості осіб, які будуть зараховані, при штучному збільшенні кількості результатів експериментальних досліджень	52
Висновки до розділу 3	57
Висновки.....	59
Список використаних джерел.....	60

Додаток А Копії публікацій.....	64
Додаток Б Формування розширеного набору даних	77
Додаток В Програмна реалізація прогнозу за допомогою штучної нейронної мережі	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШІ – штучний інтелект

ШНМ – штучні нейронні мережі

ШН – штучний нейрон

НМ – нейронна мережа

БНМ – багатошарова нейронна мережа

MLP – багатошаровий перцептрон

LSTM – рекурентна нейронна мережа

ВСТУП

Актуальність теми. Сучасний етап розвитку інформаційних технологій характеризується стрімким впровадженням методів ШІ в різноманітні сфери людської діяльності. Завдяки здатності до навчання, узагальнення та адаптації, ШНМ стали потужним інструментом для розв'язання складних задач, зокрема апроксимації функцій, класифікації об'єктів та обробки експериментальних даних. Такий значний науковий та практичний інтерес до інтелектуальних систем зумовив появу широкого кола фундаментальних досліджень у цій галузі.

Значний внесок у розвиток теорії ШІ зробили такі вчені, як Rumelhart D. E., Zurada J. M., Bernstein N., Widrow B, Kalman R. E. Вагомими також є напрацювання у сфері ШНМ науковців – С.Аlippi, А.Ferrero, V.Piuri, Dayhoff J. E., Hopfield J. J., Kroese B., Сігеру Омату, Головка В. А., Савіцкий Ю. В.

Однак на сьогодні залишаються слабо дослідженими або і не дослідженими цілий ряд аспектів як теоретичного застосування НМ різної архітектури, так і практичного застосування НМ для вирішення різноманітних задач.

Дана кваліфікаційна робота магістра присвячена дослідженню можливостей підвищення точності прогнозу часових рядів в умовах обмеженої кількості результатів експериментальних досліджень. Адже експериментальні дослідження, спрямовані на отримання великого обсягу даних мають високу трудомісткість та найчастіше вимагають значних затрат. Тому дослідження методів підвищення точності прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень є актуальною задачею, яка дасть змогу економії ресурсів і часу при експериментальних дослідженнях.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи є підвищення точності прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) провести аналіз штучного інтелекту та ШНМ;
- 2) провести аналіз нейромережових методів прогнозування часових рядів;
- 3) провести аналіз методів підвищення точності нейромережевого прогнозу;
- 4) провести дослідження можливості підвищення точності прогнозу за рахунок штучного збільшення навчальної вибірки прогнозуючої ШНМ;
- 5) провести аналіз отриманих результатів.

Об'єктом дослідження є процес нейромережевого опрацювання даних експериментальних досліджень.

Предметом дослідження є методи нейромережевого прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень.

Наукова новизна. Отримали подальший розвиток методи нейромережевого прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень, які передбачають формування навчальної вибірки прогнозуючої нейронної мережі шляхом штучного збільшення кількості цих результатів за рахунок їх попередньої апроксимації методом регресійного аналізу, що дало змогу підвищити точність і стійкість прогнозу.

Практична цінність отриманих результатів полягає у тому, що:

- 1) проведено дослідження можливості штучного збільшення кількості результатів експериментальних досліджень за рахунок їх попередньої апроксимації;
- 2) досліджено метод регресійного аналізу для апроксимації результатів експериментальних досліджень, який показав кращу ефективність порівняно з аналогом;
- 3) проведені дослідження дали змогу вибрати найбільш перспективні методи апроксимації для подальшого використання на практиці.

Публікації та апробація. За результатами проведених досліджень опубліковано двоє тез доповідей на конференціях:

– 5th International Scientific and Practical Conference «Modern Perspectives on Global Scientific Solutions» (October 13-15, 2025, Bergen, Norway).

– 5th International Scientific and Practical Conference «Scientific Exploration: Bridging Theory and Practice» (October 20-22, 2025. Berlin, Germany).

Структура та обсяг роботи. Дана кваліфікаційна робота складається із змісту, вступу, трьох розділів, висновків, списку використаних джерел і додатків.

1 ТЕОРЕТИЧНІ ОСНОВИ НЕЙРОННИХ МЕРЕЖ

1.1 Базові поняття нейронних мереж та систем штучного інтелекту

Головне завдання Міністерства цифрової трансформації України – не просто адаптуватися нашій державі до світових тенденцій, а й стати лідером у розвитку технологій штучного інтелекту (ШІ) [1]. При цьому, згідно до заяви Міністра цифрової трансформації України, головною ціллю від створення міністерства і до сьогодні є побудова найзручнішої цифрової держави у світі [1].

За словами М. Федорова, віцепрем'єр-міністра з інновацій, розвитку освіти, науки та технологій: «Майбутнє України – у розвитку цифрової економіки та інновацій. ШІ – важлива частина цього шляху. Сьогодні ШІ в Україні вже використовують у різних напрямках – від військових і цивільних технологій до повсякденного життя. Бачимо важливий фокус у розвитку технологій ШІ також у сферах освіти, охорони здоров'я, економіки, урбаністичного планування та багатьох інших. Це дасть змогу нашій державі не просто адаптуватися до світових тенденцій, а й стати лідером у цій галузі» [1].

Результати проведеного онлайн-опитування Kantar Panel Ukraine [2] для українців, щодо сфер застосування ШІ зображено на рис. 1.1. Найактивніше інструменти ШІ впроваджуються для забезпечення охорони, у діяльність фінансових установ, творчі професії та при наданні послуг підтримки користувачів.

З огляду на інтенсивну інтеграцію технологій ШІ у вітчизняний бізнес-сектор, пріоритетним викликом для Міністерства цифрової трансформації стає формування нормативно-правової бази для регламентації цієї сфери. З метою визначення підходів до регулювання ШІ, у червні 2024 року було оприлюднено «Білу книгу» [1]. Вона надає компаніям чіткі інструкції щодо створення безпечних для суспільства технологій. Дія відповідного Закону поширюватиметься на перелічені нижче категорії учасників ринку (рисунок 1.1) [3]:

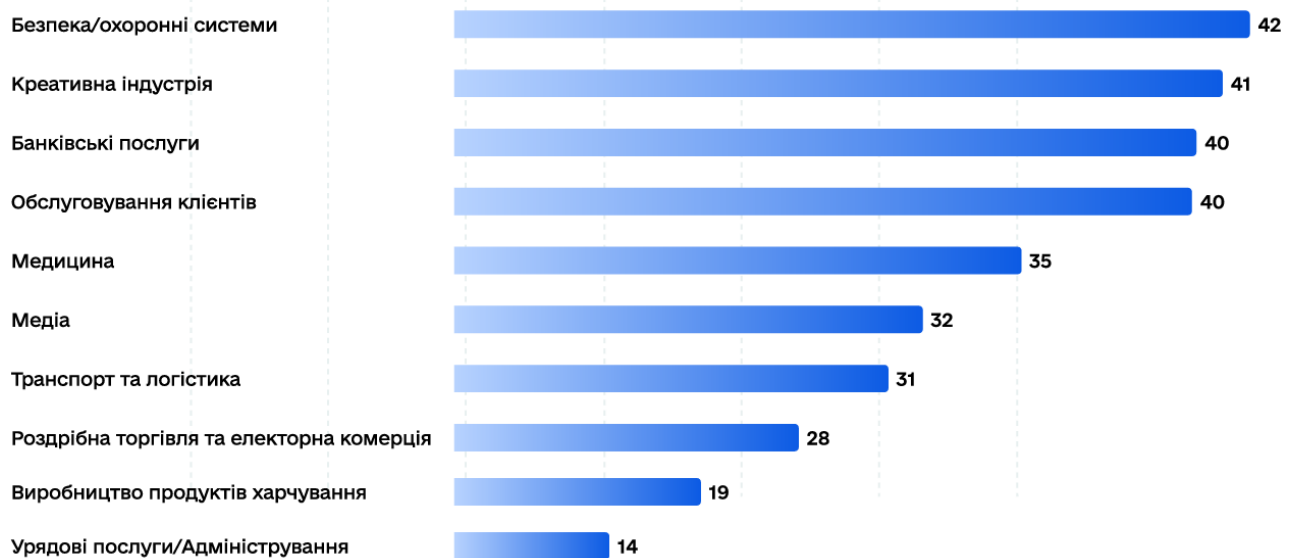


Рисунок 1.1 – Сфери, в яких українці помічають застосування ШІ

1. Розробники інтелектуальних систем – до цієї категорії належать компанії та незалежні фахівці, що займаються створенням технологій ШІ. На них покладається відповідальність за дотримання встановлених норм на всіх етапах життєвого циклу продукту – від архітектурного задуму до моніторингу функціонування.

2. Постачальники ШІ-рішень – до цієї групи входять учасники ринку, які здійснюють продаж або дистрибуцію інтелектуальних продуктів на території Європейського Союзу. Вони несуть відповідальність за гарантування повної сумісності своїх систем із положеннями Закону про штучний інтелект (AI Act).

3. Користувачі ШІ – до цієї групи належать організації та приватні особи, які інтегрують інтелектуальні системи у свої робочі процеси. Обсяг регуляторних обмежень та зобов'язань щодо моніторингу й звітності для них варіюється залежно від рівня потенційного ризику, який несе конкретна технологія.

Законодавство передбачає різні правила для різних рівнів (ступенів) ризику, встановлюючи обов'язки для розробників, постачальників та користувачів в залежності від характеру потенційного ризику технологій штучного інтелекту.

На сьогодні дефініція поняття «штучний інтелект» є доволі умовним та багатограним. Ринок перенасичений технікою з мікрочіпами, яку виробники позиціонують як таку, що має ШІ. Втім, у більшості випадків це є лише технічним відтворенням поведінкових патернів людини, основною метою якого є автоматизація рутинних задач та зниження витрат.

Термін «інтелект» (Intelligence) бере свій початок від латинського поняття *intellectus*, що перекладається як розум [4]. Сутність штучного інтелекту (Artificial Intelligence – AI) полягає у здатності технічних систем перебирати на себе людські функції: аналізувати зовнішні впливи, використовувати попередній досвід та на основі нього генерувати оптимальні рішення [5]. Фундаментом будь-якого інтелекту вважається діяльність [6].

Процес мислення розглядається як безпосередній результат функціонування мозку [7]. Інтелект та мислення тісно переплітаються через спільний спектр завдань, зокрема ідентифікацію обставин, логічну обробку даних та стратегічне планування дій. Ключовими атрибутами інтелекту вважаються вміння навчатися, узагальнювати інформацію, акумулювати досвід та адаптуватися до змінних умов під час виконання поставлених задач.

Аналіз самої дефініції ШІ породжує фундаментальне питання для розробників: наскільки можливим є моделювання мислення, ідентичного мисленню людини як дорослої, так і дитини [8].

Механізми людського мислення донині залишаються недостатньо вивченими та становлять загадку для науковців. Сучасні обчислювальні машини виконують переважно формальні операції над даними у регістрах пам'яті, незалежно від їхньої природи. Відповідно, на відміну від людини, якій притаманне оперування осмисленими поняттями, комп'ютери не здатні до справжнього усвідомлення семантичного змісту інформації.

Фундаментальна різниця між повноцінними системами штучного інтелекту та стандартними додатками полягає у здатності перших оперувати візуальними патернами, тобто «мислити» образами [9]. Саме цей принцип уможливив реалізацію низки сучасних технологій, серед яких ефективно

стиснення даних, біометрична ідентифікація, оптимізація кольоропередачі, а також інтелектуальний пошук, семантичний аналіз та автоматизована класифікація зображень [6].

На сучасному етапі проблематика штучного інтелекту посіла центральне місце у світовому дискурсі [5]. Така пильна увага зумовлена стрімкою еволюцією технологій та їхньою інтеграцією у повсякденність, що дозволило широкому загалу безпосередньо взаємодіяти з інтелектуальними системами [10]. Інструментарій ШІ пропонує інноваційні шляхи для оптимізації рутинних процесів, а також володіє потужним потенціалом для модернізації різних галузей суспільного життя та цивілізаційного поступу загалом [11].

Високий потенціал технології зумовив стрімке зростання кількості нових розробок у сфері ШІ та його динамічну інтеграцію в діяльність приватних підприємств. Разом з тим, системи штучного інтелекту стають невіддільним інструментом у повсякденній та професійній діяльності людей по всьому світу [12].

Однак розвиток ШІ вимагає не лише досліджень ШІ як такого, але і дослідженням методів, які лежать у його основі [13] – штучних нейронних мереж (ШНМ) [14], генетичних алгоритмів [15], нечіткої логіки [16] і теорії ігор [17]. При цьому слід відзначити, що можливості ШІ у значній мірі визначаються саме можливостями його методів [13].

Такий стан речей вимагає також всебічного вивчення можливостей методів ШІ, зокрема, ШНМ, при вирішенні різноманітних практичних задач з врахуванням їх особливостей.

У статтях, присвячених комп'ютерним технологіям, часто згадують ШІ та НМ як синоніми, хоча так вважати є зовсім некоректно. ШІ – це широке поняття, яке охоплює багато різних технологій, що дозволяють машинам імітувати інтелект людей. НМ – це лише один із методів (елементів) ШІ [18].

ШНМ представляють собою клас математичних або апаратно-програмних моделей, архітектура яких спроектована за аналогією з організацією нервової системи живих істот [19, 20].

Взаємодія між біологічними нейронами [19, 20] здійснюється шляхом генерації електричних імпульсів, що трансформуються у хімічні сигнали.

Саме ці складні процеси забезпечують здатність людини до мислення та сенсорного сприйняття.

У свою чергу, галузь ШІ та когнітивного моделювання ставить за мету технічне відтворення ключових принципів роботи таких природних мереж.

Крім нейромереж, штучний інтелект використовує такі технології:

- машинне навчання для розробки алгоритмів, які можуть навчатися на великих масивах даних.

- логічне програмування для вирішення завдань, які пов'язані з плануванням, розмірковуванням та прогнозуванням.

- генетичне програмування для вирішення завдань, які потребують використання принципів генетики та біологічної еволюції.

- теорія ігор для розробки стратегій та прийняття рішень в умовах невизначеності [18].

Отже, основна відмінність між ШІ та нейромережею полягає в тому, що ШІ охоплює безліч різноманітних технологій, тоді як НМ – це одна з його функціональних структур (один із його методів).

ШНМ [19–21] являють собою комплекс взаємопов'язаних обчислювальних одиниць – нейронів, схематичне зображення яких наведено на рисунку 1.2.

Функціонування кожного окремого елемента зводиться до обробки вхідних потоків даних та подальшої передачі згенерованих сигналів до інших вузлів мережі.

При цьому кожен такий елемент оперує виключно тими даними, що надходять до нього, трансформуючи їх у вихідні імпульси для трансляції сусіднім нейронам [22].

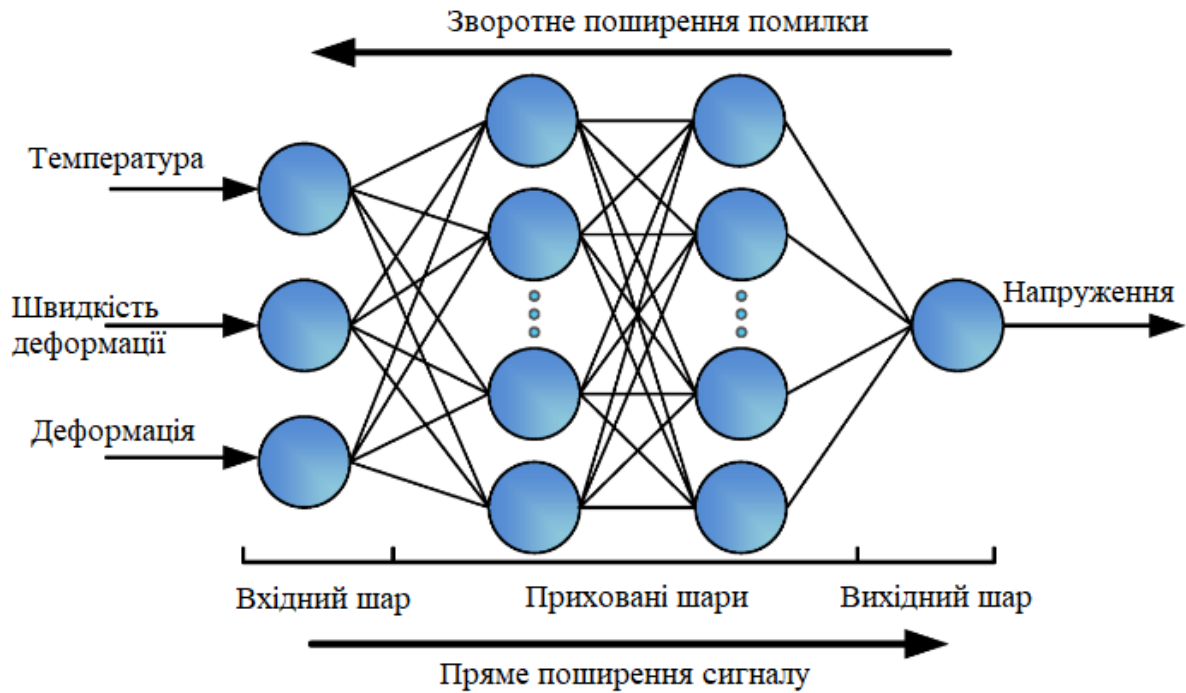


Рисунок 1.2 – Узагальнена модель ШНМ оцінки напруження у заданій механічній конструкції

ШНМ працює наступним чином [19, 21]: на входи нейронів її вхідного шару надходять сигнали, які надалі розподіляються між нейронами першого прихованого шару. У нейронах першого та наступних прихованих шарів ключову роль відіграє синаптична вага, яка визначає пріоритетність (ступінь впливу) кожного окремого вхідного сигналу. Для цього кожному виходу попереднього шару ШН присвоюється множник (коефіцієнт). При цьому зв'язки з додатними коефіцієнтами діють як збуджуючі, тоді як від'ємні значення відповідають гальмівним механізмам. На подальше опрацювання у кожному ШН поступає зважена за допомогою цих коефіцієнтів сума входів ШН.

ШН [21] поділяють на лінійні – на його вихід подаються згадана зважена сума, та нелінійні – згадана зважена сума подається на нелінійний перетворювач, вихід якого подається на вихід.

Передача сигналів від виходів одних нейронів до входів інших визначає підсумковий результат обчислень, фактично формуючи пам'ять та поведінкову модель системи. Для забезпечення здатності ШНМ вирішувати складні завдання застосовується процедура навчання. Принципова відмінність таких систем полягає у тому, що їхнє функціонування не базується на програмуванні алгоритмів у класичному розумінні, а реалізується через адаптивне навчання [22].

Ключовою перевагою нейромережових методів є нестандартний підхід до опрацювання даних. Інформація в таких системах не локалізується у відокремлених комірках пам'яті, а розподіляється через конфігурацію зв'язків та їхні вагові коефіцієнти. Відповідно, активність кожного окремого нейрона формується під впливом сукупності з'єднаних з ним елементів. Завдяки такій архітектурі руйнування або втрата окремих зв'язків не призводить до критичних збоїв у роботі системи, що гарантує її високу стабільність та надійність функціонування [23].

Притаманна нейромережам стійкість до завад та експлуатаційна надійність проявляються як при обробці зашумлених сигналів, так і у випадку несправності окремих обчислювальних вузлів. Завдяки цьому досягається висока швидкість і точність аналізу даних. Крім того, можливість оперативного донавчання системи дозволяє ефективно адаптуватися до змін у зовнішньому середовищі та розширювати спектр розв'язуваних задач.

Зазначені функціональні переваги обумовлюють доцільність використання нейромережових методів у широкому спектрі задач, зокрема:

- інтелектуальний аналіз та інтерпретація графічних даних (комп'ютерний зір);
- ідентифікація усного мовлення (незалежно від особливостей голосу спікера);
- опрацювання інтенсивних потоків цифрових даних у реальному часі;
- створення автоматизованих механізмів для оперативного пошуку інформації;

- миттєва категоризація та сортування вхідних даних (класифікація on-line);
- стратегічне планування та управління ресурсами у масштабних системах;
- знаходження розв'язків для складних багатофакторних оптимізаційних задач;
- побудова систем адаптивного контролю та прогнозування динаміки процесів [23].

Функціонування кожного нейрона базується на нелінійному перетворенні сукупності вхідних сигналів за допомогою функції активації зазвичай порогового типу. У тому випадку, коли результат цієї функції долає визначену межу – поріг, який відображає сукупну важливість отриманої інформації, відбувається збудження нейрона. Як наслідок, він генерує вихідний сигнал, який далі передається іншим елементам мережі [23].

У даному підрозділі розглянуто базові поняття ШМ і систем штучного інтелекту, структуру типової НМ, функції її складових, згадано про види нейронних мереж та основні сфери застосування.

1.2 Базовий елемент штучних нейронних мереж

Фундаментальна перевага ШНМ та нейрокомп'ютерів криється в універсальності підходу до розв'язання нетипових проблем [19, 21]. Цінність технології полягає не в абсолютній першості над вузькоспеціалізованими машинами в окремих задачах, а в мультифункціональності: одна й та ж нейросистема здатна вирішувати різні типи завдань майже з тією ж ефективністю. Це усуває необхідність проектування нового обладнання під кожену проблему. Процес класичного програмування тут замінюється навчанням: замість детального опису алгоритмів фахівець готує набори даних для тренування. Тобто праця програміста стає працею педагога, який створює умови для самостійної адаптації системи. Це робить нейрокомп'ютери

незамінними у сферах, де відсутні явні методи розв'язання і потрібен аналог інтуїтивного мислення [22].

У широкому розумінні під ШНМ прийнято розглядати систему обчислювальних елементів (штучних нейронів), що об'єднані між собою за допомогою синаптичних контактів. Саме сукупність цих вузлів та конфігурація зв'язків між ними визначають структуру мережі [19, 21].

У контексті обчислювальних механізмів нейромережі імітують біологічні процеси людського мозку, проте залишаються його суттєво спрощеною математичною абстракцією. Їхнє функціонування зводиться до трансформації вхідних даних у набір вихідних сигналів шляхом динамічної зміни внутрішніх станів системи. Зазвичай ці операції виконуються над цифровими величинами. Архітектура зв'язків визначає тип мережі: канали передачі даних від входу до виходу називають прямими, а у зворотному напрямку – зворотними. Відповідно, системи лише з одностороннім рухом інформації класифікують як мережі прямого поширення, тоді як наявність петель (зворотних шляхів) дозволяє віднести їх до класу рекурентних мереж [24].

Архітектура з'єднань між окремими нейронами визначає топологію мережі, яку класифікують на повнозв'язну та неповнозв'язну. Самі нейрони, з яких складається НМ, виконують роль елементарних процесорів: їхня обчислювальна робота регламентується правилами агрегації вхідних даних та функцією активації, що формує вихідний результат. Передача цього сигналу до інших вузлів системи відбувається через синаптичні канали, кожному з яких присвоєно певний ваговий коефіцієнт (вагу зв'язку) [24].

За аналогією з біологічними структурами, вхідні канали нейрона називають «дендритами», а вихідний шлях – «аксоном». Класифікація самих нейронів базується на їхньому розташуванні в архітектурі мережі. Елементи, що забезпечують первинний прийом даних із зовнішнього середовища, визначаються як вхідні, тоді як ті, що транслюють фінальний результат назовні – як вихідні. Всі інші вузли, що знаходяться між входом і виходом, належать до категорії прихованих і формують відповідні внутрішні шари мережі [25].

Базовим структурним елементом будь-якої НМ виступає штучний (формальний) нейрон. Він являє собою елементарний обчислювальний модуль або його математичну модель. Архітектурно такий елемент характеризується наявністю кількох вхідних каналів для прийому сигналів та єдиного виходу для передачі результату.

Схематичне зображення класичного штучного нейрона наведено на рисунку 1.3. Первинна обробка вхідних даних у ньому здійснюється шляхом підсумовування: обчислюється зважена сума всіх сигналів, до якої додається спеціальна постійна величина – зсув (bias).

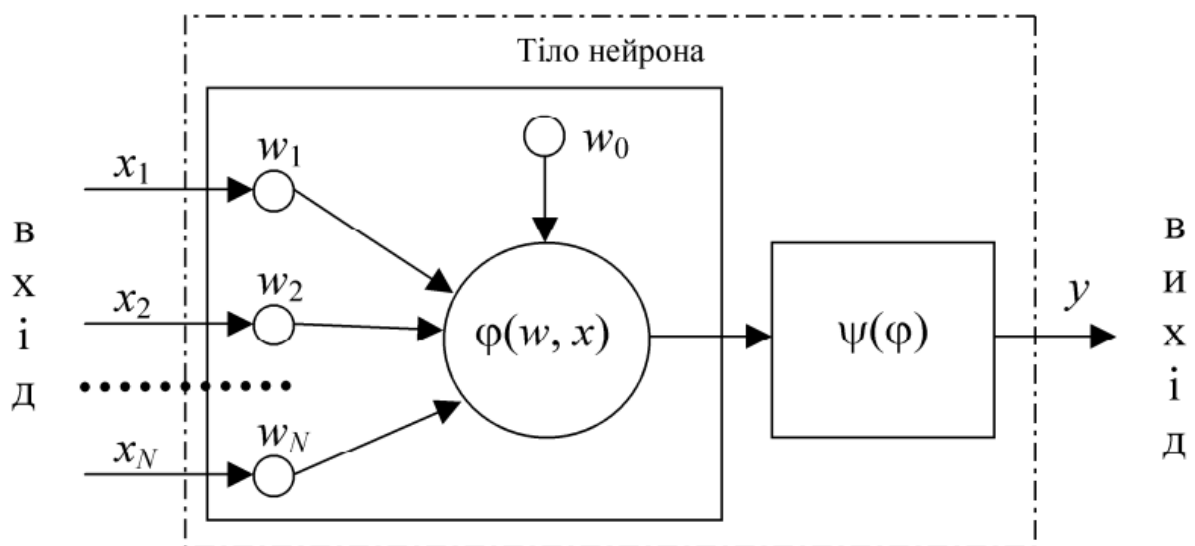


Рисунок 1.3 – Формальна модель штучного нейрона

Робота одношарового елемента починається з отримання набору сигналів, що формують вхідний вектор $x = \{x_j\}, j = 1, 2, \dots, N$, де N – кількість входів.

Далі відбувається зважування (масштабування) кожного вхідного значення x_j за допомогою відповідної ваги зв'язку – вагового коефіцієнта w_j , який присвоюється кожному компоненту цього вектора. Така математична операція є аналогом роботи біологічного синапсу, відтворює процес перетворення сигналу в синаптичних з'єднаннях [19, 21].

Функція зважування (дискримінантна функція) φ виконує підсумовування вхідних сигналів з урахуванням їхніх ваг, формуючи так званий постсинаптичний потенціал. Ця величина виступає аргументом для функції активації (передатної функції) ψ , яка трансформує її у кінцеві скалярні значення на виході. З математичної точки зору, такий елемент здійснює відображення векторного входу в скалярний вихід, що є аналогом біологічних процесів обробки сигналів у синапсах та тілі клітини.

Величини вхідних сигналів та вагових множників, як правило, є дійсними числами, конкретний діапазон яких залежить від специфіки розв'язуваної проблеми. При цьому полярність ваги визначає тип взаємодії:

- зв'язки з негативними коефіцієнтами діють як гальмівні;
- зв'язки з позитивними значеннями відповідають збуджуючим каналам.

Функціональне призначення блоку активації полягає у безпосередньому обчисленні вихідного значення нейрона [25, 26].

У підсумку, математична формалізація роботи штучного нейрона визначається рівнянням:

$$y = \psi(\varphi(w, x)),$$

де x – вектор вхідних змінних (сигналів);

y – результуючий вихідний сигнал;

ψ – функція активації (передатна функція);

φ – функція зважування (дискримінантна);

$w = \{w_j\}$ – вектор налаштувань, який об'єднує синаптичні ваги w_j та величину зсуву (поріг) w_0 .

Саме конфігурація вагових коефіцієнтів w формує пам'ять нейрона, який виконує роль запам'ятовуючого елемента. Водночас, з функціональної точки зору, нейрон виступає як елементарний процесор. Його робота полягає в обчисленні результату функції активації, який базується на попередньо отриманому значенні від дискримінантної функції (зваженої суми входів) [25].

Підсумовуючи, вихідний сигнал нейрона формується шляхом застосування функції активації до результату лінійної комбінації вхідних

даних: входів X і ваг W , то вихід нейрона дорівнює сумі добутків вхідних сигналів на їхні ваги $Y = f(z) = f(X_1 * W_1 + \dots + X_n * W_n)$, що є лінійною функцією. Оскільки робота суматора базується на лінійній комбінації векторів входу X та ваг W , цей процес описується класичним лінійним рівнянням $y = kx + b$, де b – це зміщення, яке виступає в якості вільного члена, що дозволяє зсувати графік функції активації [27].

Нейрон зсуву (або bias-нейрон) – це третій різновид нейронів, який знаходить застосування у більшості нейронних мереж. Характерна риса цього типу елементів полягає в тому, що їхні вхідні та вихідні значення завжди дорівнюють 1, при цьому вхідні синапси у них повністю відсутні. У структурі ШНМ нейрони зміщення можуть або розташовуватися по одному на кожному шарі, або бути відсутніми зовсім – інших варіантів не передбачено. Схема їх з'єднання ідентична звичайним нейронам: вони зв'язуються з усіма елементами наступного рівня. Єдиний виняток – неможливість утворення синапсу між двома bias-нейронами. Таким чином, їх доцільно розміщувати на вхідному та прихованих шарах, але ніколи на вихідному, оскільки там їм просто ні з чим буде встановити зв'язок [27].

У графічному представленні архітектури мереж нейрони зсуву часто візуально не відображають, обмежуючись врахуванням їхнього впливу безпосередньо в розрахунках. Оскільки вихідний сигнал такого нейрона завжди дорівнює одиниці, це дозволяє умовно розглядати його як додатковий синапс із вагою, яку додають до загальної суми без явної згадки самого нейрона. Відтак, підсумовуючи сказане, математичну модель нейрона можна подати у вигляді [19, 25]:

$$Y = f(z) = f(X_1 * W_1 + X_2 * W_2 + \dots + X_n * W_n + b) = f(\sum w_i \cdot x_i + b) \quad (1.1)$$

де $f(z)$ – функція активації нейрона;

$X_1 \dots X_n$, x_i – значення, які поступають на входи нейрона;

$W_1...W_n$, w_i – вагові коефіцієнти нейрона;

b – зміщення нейрона.

Варто зауважити, що процес опрацювання даних у формальній моделі нейрона (див. рисунок 1.3) має суттєві розбіжності з принципами роботи його біологічного аналога. Головні відмінності зводяться до такого [19, 27]:

- відсутній інструмент для врахування часової затримки при формуванні вихідного сигналу;
- не застосовується модуляція інтенсивності вхідного сигналу через щільність нервових імпульсів;
- у переважній більшості НМ ігнорується ефект синхронізації активності нейронів;
- немає зовнішньої системи керування (аналога гормональної регуляції), яка б координувала роботу мережі в цілому;
- не передбачено механізму динамічної адаптації порогу активації та вагових коефіцієнтів безпосередньо під час функціонування мережі;
- взаємодія між нейронами обмежується виключно збуджуючими та гальмівними зв'язками.

Внаслідок зазначених відмінностей, застосування НМ для моделювання динамічних процесів вимагає залучення додаткових компонентів, що не є інтегрованими в структуру самої мережі. Також слід враховувати той факт, що гнучкість НМ та їхні адаптивні можливості у змінних умовах значно слабші, ніж у біологічних аналогів [19, 24].

У даному підрозділі розглянуто основний елемент, на якому будуються ШНМ – штучний нейрон, вивчено його внутрішню структуру, модель та принципи його функціонування.

1.3 Функції активації штучних нейронів

Параметри та властивості найпопулярніших функцій активації, характерних для класичних архітектур НМ, систематизовано в таблиці 1.1. Слід

зауважити, що прийнято такі умовні позначення: θ – це поріг активації (або зсув), a – параметр крутизни, тоді як σ – визначає радіус функції Гаусса.

Таблиця 1.1 – Функції активації штучних нейронів

№	Функція активації	Формула	Використання	Недоліки
1	Порогова	$f(x) = 1$, якщо $x \geq t$, інакше 0 , де t – порогове значення	Рідко	Безперервність, відсутність градієнта
2	Лінійна	$f(x) = x$	Прості моделі	Обмежене уявлення нелінійності
3	Гіперболічний тангенс	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Приховані шари	Згасаючий градієнт
4	Сигмоїда	$f(x) = \frac{1}{1 + \exp(-x)}$	Задачі класифікації	Згасаючий градієнт
5	ReLU	$f(x) = \max(0, x)$	Глибокі мережі	«Мертві нейрони» з негативним входом
6	Leaky ReLU	$f(x) = \max(0, 0.1x, x)$	Глибокі мережі	Додатковий параметр нахилу може бути неоптимально налаштований, що може впливати на продуктивність моделі.
7	Parametric ReLU (PReLU)	$f(x) = \max(ax, x)$, де a – параметр	Глибокі мережі	Обчислювальна складність
8	ELU (Exponential ReLU)	$f(x) = x$ за умови $x \geq 0$, $a(\exp(-1))$ за умови $x < 0$	Глибокі мережі	Наявність негативних значень може спричинити збільшення складності оптимізації моделі.
9	Maxout	$f(x) = \max(w1*x +b1, w2*x + b2)$, де $w1, w2$ – вагові коефіцієнти, $b1, b2$ – зміщення	Глибокі мережі	Збільшення обчислювальної складності та вимог більшого обсягу даних для навчання

У сучасних нейромережевих архітектурах набули поширення складні види функцій активації (див. таблицю 1.1) [26]. Зокрема, специфікою семантичних мереж є використання апарату нечіткої логіки. Натомість у

глибоких повнозв'язних та згорткових моделях найчастіше застосовують функції типу ReLU та Softmax. Окрім того, існують прецеденти використання гібридних (складених) функцій.

На практиці рішення про те, яку функцію активації використати, залежить від характеру завдання, швидкодії програмної реалізації та специфіки алгоритму навчання. Сьогодні не існує єдиного стандартного підходу до цього вибору, проте відомо про низку обмежень, які накладаються на використання певних видів функцій.

У даному підрозділі розглянуто типи та характеристики основних функцій активації ШН, які використовуються на практиці.

1.4 Постановка задачі дослідження

Важливим етапом у наукових та практичних розробках є проведення експериментальних досліджень. Однак отримання достовірних результатів часто пов'язане зі значними труднощами: високою вартістю обладнання, тривалістю проведення експериментів та значними трудовитратами. У результаті дослідники часто стикаються з проблемою обмеженого обсягу отриманих даних.

Сучасні інформаційні технології, зокрема методи штучного інтелекту та нейронні мережі, дозволяють ефективно опрацьовувати дані та будувати прогнози. Проте їх ефективність напряду залежить від обсягу навчальної вибірки. Для коректного навчання нейронних мереж та створення узагальнюючих моделей потрібні великі масиви даних.

В умовах дефіциту вхідних даних, при малих вибірках, виникають суттєві проблеми при застосуванні нейромережових методів. Невелика кількість точок для навчання призводить до того, що нейронна мережа не здатна виявити приховані закономірності процесу. Замість узагальнення і прогнозування, мережа просто «запам'ятовує» наявні приклади та перенавчається, що робить її

непридатною для прогнозування нових значень. Похибка прогнозу в таких випадках стає неприпустимо великою.

Оскільки нестача експериментальних даних обмежує використання ШНМ таких, як LSTM або MLP, тому виникає потреба в розробці спеціальних підходів до попереднього опрацювання вхідних масивів. Актуальним є підхід штучного збільшення навчальної вибірки за допомогою методів апроксимації регресійного аналізу, що дозволить покращити якість навчання нейронної мережі та підвищити точність прогнозу.

Тому, метою кваліфікаційної роботи є підвищення точності прогнозу часових рядів при обмеженій кількості результатів експериментальних досліджень.

Завдання кваліфікаційної роботи:

- провести аналіз ШІ та ШНМ;
- провести аналіз нейромережових методів прогнозування часових рядів;
- провести аналіз методів підвищення точності нейромережевого прогнозу;
- провести дослідження можливості підвищення точності прогнозу за рахунок штучного збільшення навчальної вибірки прогнозуючої ШНМ;
- провести аналіз отриманих результатів.

Висновки до розділу 1

1. Проведено аналіз базових поняття ШНМ і систем штучного інтелекту, показано, що ці поняття суттєво відрізняються.
2. Проведено огляд структури типової ШНМ, функцій її складових, основних видів ШНМ та основні сфери їх застосування.
3. Детально вивчено основний елемент, на якому будуються ШНМ – штучний нейрон, розглянуто його внутрішню структуру, модель та принципи його функціонування, а також основні функції активації, які використовуються на практиці.

2 НЕЙРОМЕРЕЖЕВІ МЕТОДИ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

2.1 Типові архітектури нейронних мереж

До кожного нейрона через систему нервових закінчень надходять сигнали від сусідніх клітин. За своєю природою ці впливи можуть бути стимулюючими або пригнічуючими, а їхня сукупність визначає поточний потенціал у тілі нейрона. Якщо ця величина перевищує встановлене порогове значення, нейрон переходить у стан збудження і транслює сигнал далі. Аналогічно, штучні нейрони можуть об'єднуватися між собою за різними схемами. Це дає змогу створювати різноманітні конфігурації ШНМ, які зображені на рисунку 2.1, зі своїми унікальними правилами навчання та сферами застосування [23]:

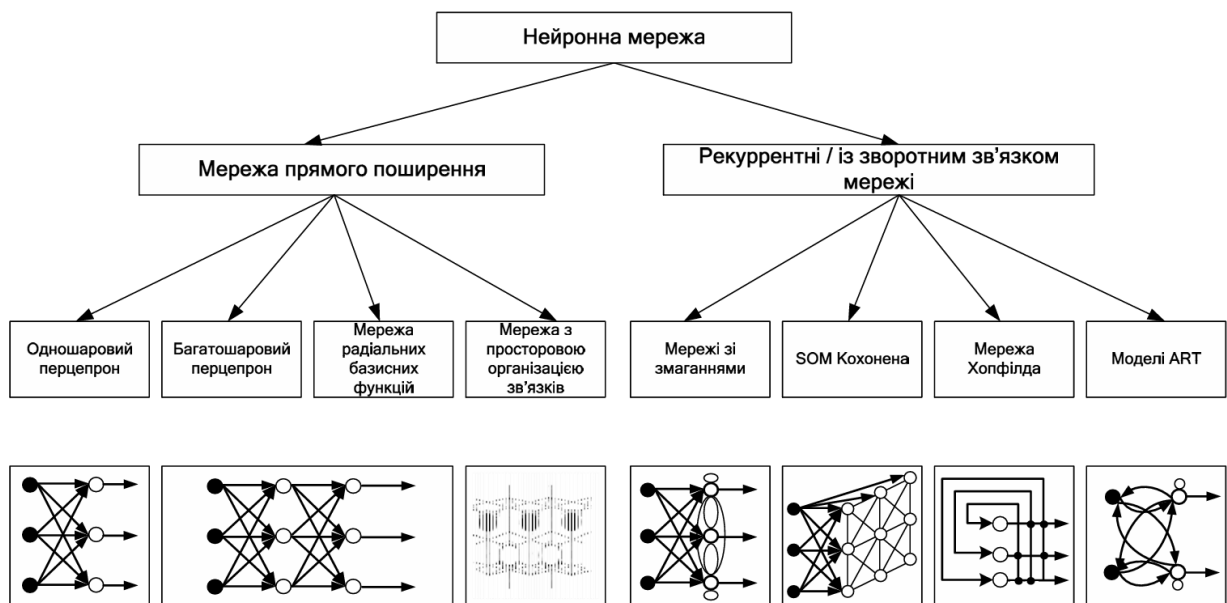


Рисунок 2.1 – Базова архітектура нейронних мереж

Формально ШНМ описують як спрямований граф, у якому вузлами є нейрони, а зв'язки мають певні ваги. За типом з'єднань розрізняють два основні класи мереж (див. рисунок 2.1): ациклічні (прямого розповсюдження) та циклічні (рекуррентні, або зі зворотним зв'язком). Найвідомішим представником першого класу є багатшаровий перцептрон (MLP). Його структура передбачає

сувору пошарову організацію нейронів з односпрямованою передачею сигналів від попереднього шару до наступного [23].

Мережі прямого поширення вважаються статичними, оскільки для конкретного вхідного набору вони генерують фіксований результат, який жодним чином не залежить від попереднього стану системи. Натомість рекурентні мережі є динамічними, адже завдяки наявності зворотних зв'язків вхідні сигнали нейронів модифікуються, що, у свою чергу, призводить до зміни поточного стану мережі.

Станом на сьогодні відомо про існування більш ніж 200 типів парадигм ШНМ (включно з детермінованими та ймовірнісними). Десятки варіантів нейроструктур вже втілено в «залізі» – у спеціальних кристалах і платах, на базі яких функціонують потужні обчислювальні станції та суперкомп'ютери. Сучасні технології перетнули межу, що дозволяє виготовляти системи з 3,4млрд. нейронів – саме такою кількістю оцінюється їх число в мозку людини [23].

Дослідницький інститут штучного інтелекту (Інститут Азімова) у Нідерландах, який публікує фундаментальні прориви в дослідженнях ШНМ, консультує щодо впливу даних та штучного інтелекту на суспільство та бізнес, а також досліджує зв'язок між глибоким навчанням та креативністю, опублікував 32 типи найпоширеніших ШНМ (таблиця 2.1) [27].

Таблиця 2.1 – Види та класифікація ШНМ

№	Нейронна мережа (укр.)	Нейронна мережа (англ.)	Позначення	Застосування
1	Перцептрон	Perceptron	P	Базова класифікація бінарних даних (лінійно роздільних).
2	Мережа прямого поширення	Feed forward	FF, FFNN	Класифікація, регресія, апроксимація функцій (найпростіша загальна мережа).

Продовження таблиці 2.1

№	Нейронна мережа (укр.)	Нейронна мережа (англ.)	Позначення	Застосування
3	Мережа з радіальними базисними функціями	Radial basis network	RBF	Апроксимація функцій, класифікація, часові ряди.
4	Глибока мережа прямого поширення	Deep feed forward	DFF	Складні задачі класифікації та регресії, обробка даних великого об'єму.
5	Рекурентна нейронна мережа	Recurrent neural network	RNN	Обробка послідовностей (мови, текст, часові ряди), моделювання послідовностей.
6	Мережа довготривалої / короткочасної пам'яті	Long / Short term memory	LSTM	Розпізнавання мови, машинний переклад, прогнозування часових рядів (послідовності із залежностями).
7	Рекурентна мережа з керованими блоками	Gated recurrent unit	GRU	Обробка послідовностей, машинний переклад (спрощена версія LSTM).
8	Двонаправлена рекурентна нейронна мережа	BiRNN	BiRNN	Аналіз послідовностей, що потребують контексту з обох боків (наприклад, контекстуальний аналіз тексту).
9	Двонаправлена мережа довготривалої / короткочасної пам'яті	Bidirectional Long / Short term memory	BiLSTM	Розпізнавання іменованих сутностей, розуміння контексту в послідовностях.
10	Двонаправлена рекурентна одиниця	Bidirectional Gated recurrent unit	BiGRU	Схоже застосування до BiLSTM, але з меншою обчислювальною складністю.
11	Автоенкодер	Auto encoder	AE	Зниження розмірності, стиснення даних, виявлення аномалій, генерація.
12	Варіаційний автоенкодер	Variational AE	VAE	Генерація нових даних (зображень, тексту), навчання прихованих представлень.
13	Автоенкодер з шумозаглушенням	Denosing AE	DAE	Видалення шуму з даних (наприклад, із зображень).

Продовження таблиці 2.1

№	Нейронна мережа (укр.)	Нейронна мережа (англ.)	Позначення	Застосування
14	Розріджений автоенкодер	Sparse AE	SAE	Навчання розріджених (багато нулів) представлень, виділення ознак.
15	Ланцюг Маркова	Markov chain	MC, DTMC	Моделювання послідовностей, прогнозування станів системи.
16	Мережа Хопфілда	Hopfield network	HN	Асоціативна пам'ять, оптимізаційні задачі.
17	Машина Больцмана	Boltzmann machine	BM	Генеративне моделювання, навчання прихованих ознак.
18	Обмежена машина Больцмана	Restricted BM	RBM	Співпрацююча фільтрація, попереднє навчання глибоких мереж.
19	Мережа глибоких довір	Deep belief network	DBN	Класифікація, регресія (багатошарова архітектура з RBM).
20	Глибока згорткова нейронна мережа	Deep convolutional network	DCN	Обробка зображень та відео (класифікація, розпізнавання об'єктів), іноді текст.
21	Деконволюційна мережа	Deconvolutional network	DN	Сегментація зображень, візуалізація вивчених ознак.
22	Глибока згорткова інверсна графічна мережа	Deep convolutional inverse graphics network	DC-IGN	Відтворення 3D моделей з 2D зображень, маніпуляція атрибутами зображень.
23	Генеративно-змагальна мережа	Generative adversarial network	GAN	Генерація реалістичних зображень, відео, тексту; збільшення даних.
24	Машина рідких станів	Liquid state machine	LSM	Обробка часових рядів, обчислення у реальному часі.
25	Машина екстремального навчання	Extreme learning machine	ELM	Швидке навчання, класифікація та регресія з одним прихованим шаром.
26	Мережа з відлунням стану	Echo state network	ESN	Обробка часових рядів, прогнозування (тип рекурентної мережі).

Продовження таблиці 2.1

№	Нейронна мережа (укр.)	Нейронна мережа (англ.)	Позначення	Застосування
27	Глибока залишкова мережа	Deep residual network	DRN	Класифікація та розпізнавання зображень (дозволяє створювати дуже глибокі мережі).
28	Диференційований нейронний комп'ютер	Differentiable neural computer	DNC	Складне міркування, планування, використання зовнішньої пам'яті.
29	Нейронна машина Тюрінга	Neural turing machine	NTM	Алгоритмічні задачі, що вимагають пам'яті та символічних маніпуляцій.
30	Капсульна мережа	Capsule network	CN	Обробка зображень (краще розпізнавання ієрархії об'єктів та їх положення).
31	Мережа Кохонена (Самоорганізуюча карта ознак)	Kohonen network (Self-organizing feature map)	KN, SOM, SOFM	Візуалізація та кластеризація складних даних, зниження розмірності.
32	Мережа уваги	Attention network	AN	Машинний переклад, розуміння тексту, обробка зображень (фокусування на важливих частинах вхідних даних).

Кожна мережа має свої характеристики, які можна застосовувати у тому чи іншому випадку.

З запропонованих 32-ох моделей виділимо 16 типів архітектур НМ, які найчастіше використовуються. За архітектурою зв'язків згрупуємо їх у два великі класи:

- мережі прямого поширення (з односкерованими послідовними зв'язками);
- мережі зворотного поширення (з рекурентними зв'язками).

Мережі прямого поширення (Feedforward networks) характеризуються односкерованими послідовними зв'язками від входу до виходу, без циклів та зворотних зв'язків. Використовуються для класифікації, апроксимації та прогнозування. Найчастіше використовуються:

1. Перцептрон Розенблата (Perceptron) – класична одношарова мережа для розпізнавання образів. Основний блок для багатьох інших ШНМ. Використовується для бінарної класифікації.

2. Багатошаровий перцептрон (Multilayer perceptron, MLP): складається з декількох шарів перцептронів і використовується для різноманітних завдань, включаючи класифікацію та регресію.

3. Нейромережа зворотного поширення похибки (Back propagation network) – багатошаровий перцептрон (MLP) з навчанням методом градієнтного спуску. Використовується для навчання багатьох інших типів ШНМ, зокрема MLP.

4. Мережа Delta Bar Delta (Delta Bar Delta network) – feedforward-мережа з адаптивним коефіцієнтом навчання.

5. Мережа Extended Delta Bar Delta (Extended Delta Bar Delta network) – покращений варіант попередньої, також feedforward.

6. Мережа спрямованого випадкового пошуку (Directed random search network) – архітектурно feedforward, але оптимізує ваги стохастичним пошуком.

7. ШНМ вищого порядку або функціонально-зв'язана нейрона мережа (Higher-Order Neural network) – розширює MLP додатковими нелінійними комбінаціями входів.

8. Мережа Кохонена (Kohonen Self-Organizing Map, SOM) – топологічна карта без зворотних зв'язків, використовує конкурентне навчання. Застосовується для кластеризації та візуалізації даних.

9. Мережа квантування навчального вектора (Learning Vector Quantization, LVQ) – модифікація SOM для класифікації.

10. Мережа зустрічного поширення (Counterpropagation network) – поєднує SOM і асоціативне відображення, має два шари, але без повноцінних рекурентних циклів.

11. Ймовірнісна ШНМ (Probabilistic neural network, PNN) – статистична feedforward-архітектура, використовує метод Парзена для класифікації.

12. Мережа Хемінга (Hamming network) – також має архітектуру прямого поширення з подальшою нормалізацією відстаней.

Усі ці 12 моделей ШНМ мають односкеровані послідовні з'єднання (input → hidden → output), без глобальних циклів.

Рекурентні мережі (Recurrent networks) характеризуються зворотними зв'язками, наявністю циклів та здатністю до асоціативної пам'яті або моделювання динамічних систем.

1. Мережа Хопфілда (Hopfield network) – класична рекурентна асоціативна пам'ять, кожен нейрон з'єднаний з усіма іншими.

2. Машина Больцмана (Boltzmann machine) – стохастичний аналог Хопфілда з елементами термодинаміки. Використовуються для розв'язання завдань асоціативної пам'яті і оптимізації.

3. Двонаправлена асоціативна пам'ять (Bidirectional associative memory, BAM) – двошарова мережа з двосторонніми зворотними зв'язками.

4. Мережа адаптивної резонансної теорії (Adaptive Resonance Theory network, ART) – рекурентна архітектура для кластеризації та стабільного навчання в реальному часі.

Ці чотири мережі мають рекурентні з'єднання (цикли), що дозволяє їм зберігати стан системи, реалізовувати пам'ять і здійснювати оптимізацію.

ШНМ прямого розповсюдження є статичними в тому розумінні, що на певний вхід вони видають одну й ту ж сукупність вихідних значень, ігноруючи попередній стан мережі. Рекурентні ж мережі класифікують як динамічні: через зворотні зв'язки в них відбувається корекція входів нейронів, що викликає зміну стану самої мережі.

З розвитком ШНМ розроблено безліч різних типів, які переплітаються між собою для вирішення певних завдань у різних сферах. На даний момент складно класифікувати будь-яку мережу лише за однією ознакою. Це можна зробити за принципом застосування, типом вхідної інформації, характером навчання, характером зв'язків, сферою застосування тощо. В таблиці 1.2

проведено класифікацію різних видів ШНМ за принципом застосування, методом навчання та сферою застосування.

Функціонування більшості моделей НМ передбачає обов'язковий етап навчання, метою якого є оптимізація внутрішніх параметрів системи для ефективного розв'язання поставленої задачі. У загальному випадку процес навчання зводиться до ітеративного налаштування (розрахунку) вагових коефіцієнтів синаптичних зв'язків, тоді як сама структура НМ (число вузлів і наявність зв'язків) визначається ще до початку тренування.

У ході тренування мережі пропонують набір навчальних прикладів, де кожному відповідає певний вектор ознак. Ваги зв'язків змінюються так, щоб забезпечити максимальну відповідність НМ цим прикладам. Процес зміни коефіцієнтів реалізується відповідно до обраного алгоритму. Варто зазначити, що існують алгоритми (наприклад, «нейронний газ»), де крім модифікації ваг допускається зміна числа нейронів у структурі мережі [23].

«Багатошарова нейронна мережа (БНМ) прямого поширення (багатошаровий персептрон) складається з формальних нейронів і характеризується наступними параметрами і властивостями: M – кількість шарів мережі, N_μ – кількість нейронів у μ -му шарі, зв'язки між нейронами у шарі відсутні. Виходи нейронів μ -го шару, $\mu=1,2,\dots, M-1$, надходять на входи нейронів тільки наступного $\mu+1$ -го шару. Зовнішній векторний сигнал x надходить на входи нейронів тільки першого шару, виходи нейронів останнього M -го шару утворюють вектор виходів мережі $y^{(M)}$ » [26]. Схема мережі показана на рисунку 2.2.

«Кожен i -й нейрон μ -го шару ((μ, i) -й нейрон) перетворює вхідний вектор $x^{(\mu,i)}$ у вихідну скалярну величину $y^{(\mu,i)}$. Це перетворення складається з двох етапів: спочатку обчислюється дискримінантна функція $\varphi^{(\mu,i)}(w^{(\mu,i)}, x^{(\mu,i)})$, яка далі перетворюється на вихідну величину $y^{(\mu,i)} = \psi^{(\mu,i)}(\varphi^{(\mu,i)}(w^{(\mu,i)}, x^{(\mu,i)}))$, де $w^{(\mu,i)} = (w_0^{(\mu,i)}, w_1^{(\mu,i)}, \dots, w_{N_\mu}^{(\mu,i)})^T$ – вектор вагових коефіцієнтів нейрона, $x_j^{(\mu,i)}$ – j -та компонента N_μ – вимірного вхідного вектора $x^{(\mu,i)}$ » [26].

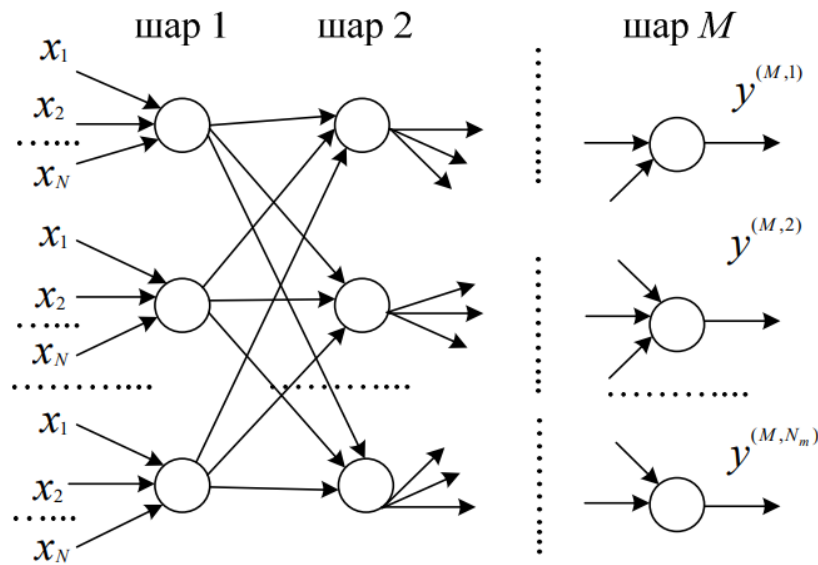


Рисунок 2.2 – Структура багатошарової нейронної мережі

Якість розв'язку задачі, який забезпечує БНМ, суттєво залежить від структурної організації мережі. Зокрема, на це впливають три основні параметри: число шарів, кількість нейронів у кожному з цих шарів, а також обсяг зв'язків між ними.

Для різних класів задач, що вирішуються за допомогою НМ, запропоновані евристичні оцінки кількості шарів і нейронів [26].

«Для забезпечення узагальнювальних властивостей необхідно, щоб кількість ваг і порогів БНМ $N_w \ll NS$, де N – розмірність вхідного сигналу (кількість ознак), S – кількість екземплярів у навчальній вибірці» [26].

Загальна кількість ваг і порогів у БНМ визначається формулою:

$$N_w = \sum_{\mu=1}^M N_{\mu} (N_{\mu-1} + 1), N_0 = N. \quad (1.2)$$

Виходячи з міркувань простоти, будемо вважати, що розмірність виходу БНМ дорівнює 1 (останній шар містить тільки один нейрон). Кількість шарів

при цьому варіюється від двох до трьох: менша кількість унеможливило вирішення нелінійних проблем, а понад три шари використовувати недоцільно, оскільки доведено, що тришарова мережа здатна апроксимувати будь-яку обчислювану функцію. Беручи до уваги також рівність $N \approx S$, маємо:

– для двошарової БНМ:

$$\begin{aligned} N_w &= N_1(N+1) + (N_1+1) = N_1(N+2) + 1 \ll NS, \\ N_1 &\ll \frac{NS-1}{N+2} \Rightarrow N_1 \ll N; \end{aligned} \quad (1.3)$$

– для тришарової БНМ:

$$N_w = N_1(N+1) + N_2(N_1+1) + (N_2+1) = N_1(N+1) + N_2N_1 + 2N_2 + 1 \ll NS. \quad (1.4)$$

У даному підрозділі розглянуто основні архітектури ШНМ – мережі прямого поширення та рекурентні мережі. Проаналізовані їх види та проведено класифікацію НМ, показано основну структуру багатошарової НМ.

2.2 Модель MLP для прогнозування часових рядів нейромережевими методами

Багатошаровий перцептрон (Multi-Layer Perceptron, MLP) [19, 28, 29] представляє собою тип нейромережі, що складається з повністю з'єднаних нейронів із нелінійною активацією (див. рисунок 2.2). Його широко застосовують для розв'язання задач, де дані неможливо розділити лінійними методами [30].

MLP знаходять використання у багатьох галузях, зокрема при розпізнаванні образів, обробці текстів та аналізі голосу. Їхня гнучка архітектура, а також здатність за певних умов апроксимувати будь-яку

функцію, перетворили їх на базовий будівельний блок для глибокого навчання. Схеми будови багатошарового перцептрона наведені на рисунку 2.3.

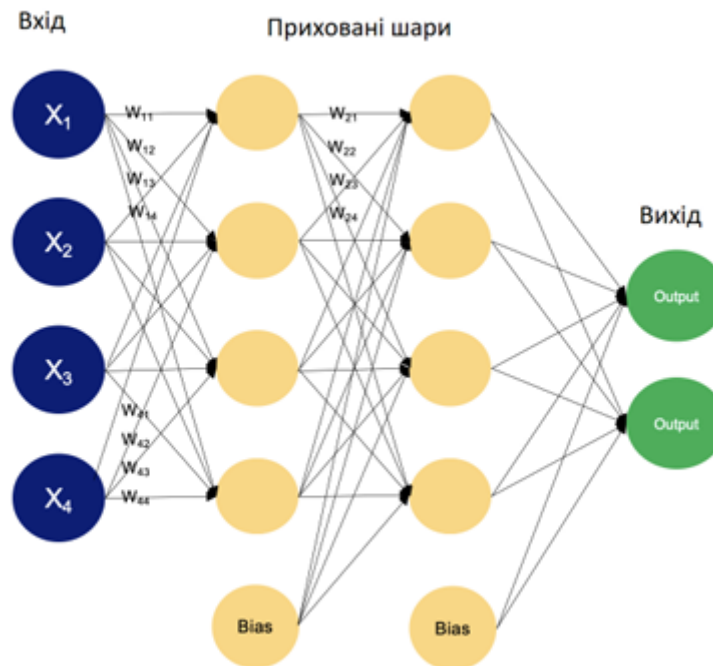


Рисунок 2.3 – Багатошаровий перцептрон (анг. multi-layer perceptron, MLP)

До складу вхідного шару входять вузли або нейрони, завданням яких є прийом початкових вхідних даних. Кожен такий елемент відповідає певній характеристиці чи виміру інформації, що надходить. Кількість нейронів у цьому шарі безпосередньо залежить від розмірності вхідних даних.

Простір між вхідним та вихідним рівнями мережі заповнюють приховані шари, кількість яких може варіюватися від одного до кількох (залежно від глибини архітектури). Топологія з'єднань у цих шарах зазвичай є повнозв'язною: кожен нейрон акумулює сигнали від усіх вузлів попереднього рівня (чи це вхідний масив, чи інший прихований шар) і транслює оброблений результат на наступну ланку. Параметри архітектури, такі як число прихованих шарів та кількість нейронів у них, належать до категорії гіперпараметрів і визначаються емпірично на етапі проєктування моделі.

Фінальною ланкою архітектури є вихідний шар, функція якого полягає у формуванні результуючого сигналу мережі. Число обчислювальних елементів на цьому рівні регламентується специфікою розв'язуваної задачі. Зокрема, для

бінарної класифікації зазвичай застосовують один нейрон або два нейрони, залежно від обраної функції активації. У випадку ж багатокласової класифікації вихідний шар формується з кількох нейронів.

Нейрони, розташовані на сусідніх рівнях, об'єднані між собою системою повних зв'язків. Для кожного такого з'єднання існує відповідна вага, яка визначає інтенсивність (силу) передачі сигналу. Ці значення не є статичними, їх оптимізація та налаштування відбуваються безпосередньо в ході ітеративної процедури навчання [30].

Окрім вхідних і прихованих елементів, у кожному шарі (окрім вхідного) зазвичай розміщують нейрон зміщення (англ. Bias Neurons). Він відповідає за подачу постійного вхідного значення на нейрони наступного шару. Кожен такий зв'язок характеризується власним ваговим коефіцієнтом, оптимізація якого відбувається нарівні з іншими параметрами під час процедури навчання.

Нейрон зсуву дозволяє ефективно зміщувати значення функції активації у наступному шарі, надаючи мережі здатність враховувати зсув границі прийняття рішень. Змінюючи ваги, відповідні цьому нейрону, багат шаровий перцептрон MLP вчиться регулювати поріг активації, що дозволяє йому краще підлаштовуватися під особливості навчальних даних.

Важливо підкреслити, що при описі MLP слово «зміщення» може стосуватися двох взаємопов'язаних, але відмінних сутностей: нейрона зміщення (описаного раніше) та загального поняття зміщення в машинному навчанні. В останньому випадку йдеться про помилку апроксимації, що з'являється при спробі описати реальну проблему за допомогою спрощеної моделі. Зсув слугує мірою того, наскільки точно модель здатна відтворити головні закономірності вхідних даних. Значне відхилення сигналізує про те, що обрана модель занадто проста і не відповідає даним, тоді як мале відхилення вказує на якісне виявлення основних закономірностей [19, 30].

Як правило, кожен нейрон у прихованих або вихідних шарах обробляє свою зважену суму входів за допомогою функції активації. До найпопулярніших видів належать сигмоїда, гіперболічний тангенс і ReLU

(випрямлена лінійна одиниця). Саме ці функції додають мережі нелінійних властивостей, надаючи їй здатність засвоювати та розпізнавати складні закономірності, приховані в даних.

У даному параграфі розглянуто модель MLP для прогнозування часових рядів нейромережевими методами, показано її переваги і недоліки.

2.3 Модель LSTM для прогнозування часових рядів нейромережевими методами

Мережі довгострокової короткочасної пам'яті (LSTM) – це спеціалізований клас архітектур, орієнтований на виявлення довготривалих часових залежностей. Вперше запропоновані Хохрайтером та Шмідхубером, ці моделі згодом були суттєво вдосконалені та набули значного поширення завдяки працям численних дослідників. Сьогодні вони демонструють високу ефективність у розв'язанні широкого спектра задач і є загальнозживаним інструментом аналізу даних [29, 30].

Основна мета розробки LSTM полягала в усуненні складнощів, пов'язаних із запам'ятовуванням довготривалих залежностей. Будь-яка рекурентна мережа має форму ланцюжка, що складається з повторюваних модулів. У традиційних рекурентних мережах ці модулі відрізняються простотою конструкції – наприклад, вони можуть містити лише один тангенціальний шар (tanh), що відображено на рисунку 2.4 [29].

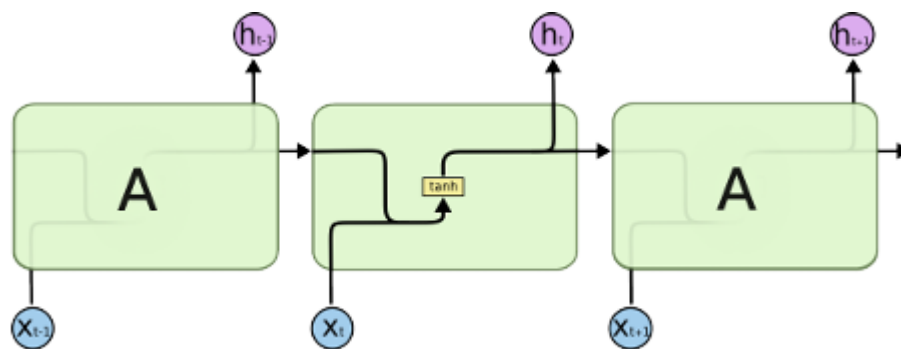


Рисунок 2.4 – Структурна організація повторюваного блоку зводиться до одного функціонального шару

Мережі LSTM базуються на загальній ланцюговій топології, характерній для рекурентних моделей, проте внутрішня архітектура їхніх повторюваних блоків є значно складнішою. На відміну від стандартної схеми з одним шаром, модуль LSTM включає чотири шари, об'єднані специфічним механізмом взаємодії, який зображений на рисунку 2.5 [29].

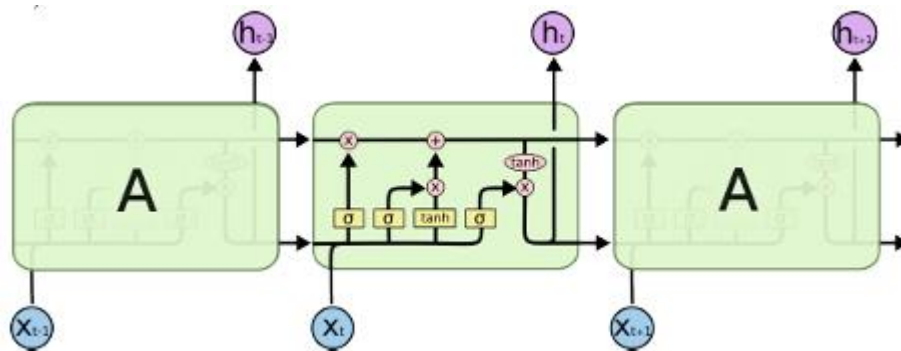


Рисунок 2.5 – Структура повторюваного блоку LSTM включає чотири взаємопов'язані шари

На схемі (див. рисунок 2.5) спрямовані лінії відображають маршрути передачі векторних сигналів між вузлами. Графічні примітиви інтерпретуються наступним чином: рожеві кола відповідають операціям поелементної обробки (наприклад, додаванню векторів), а жовті прямокутники – шарам нейромережі, параметри яких налаштовуються під час навчання. Злиття ліній символізує операцію конкатенації, тоді як їх розгалуження вказує на дублювання даних для паралельного використання. Фундаментальним елементом архітектури LSTM є стан комірки (Cell State), візуалізований як наскрізна горизонтальна вісь у верхній частині діаграми. Цей компонент функціонує за принципом конвеєра: інформація поширюється вздовж усього ланцюжка модулів, зазнаючи лише мінімальних лінійних перетворень, що забезпечує стабільність потоку даних [29, 30].

LSTM володіє здатністю видаляти стару або вносити нову інформацію до стану комірки; цей процес суворо контролюється структурами, відомими як

ворота (gates). Вони слугують інструментом для вибіркового (необов'язкового) пропускання даних. До складу воріт входить шар нейромережі з функцією сигмоїди, а також вузол, де виконується поелементне множення (pointwise multiplication operation) [29–31].

Слід також зазначити, що існують різні варіанти версій реалізації LSTM: класичний LSTM, двонаправлений LSTM (BiLSTM), закритий рекурентний блок / Gated Recurrent Unit (GRU), ConvLSTM (згортковий LSTM), LSTM з механізмом уваги.

Розглянемо метрики ефективності прогнозування часових рядів.

Середня абсолютна похибка (Mean Absolute Error, MAE) – це метрика, що являє собою суму абсолютних відхилень прогнозних значень від реальних, поділену на кількість спостережень.

$$MAE = \frac{1}{n} \sum_{i=0}^n |y_i - \hat{y}_i| \quad (2.1)$$

де y_i – фактичне значення в момент часу i ;

\hat{y}_i – прогнозоване значення в момент часу i ;

n – кількість спостережень.

Середня квадратична похибка (Mean Squared Error, MSE) – розраховується як середнє арифметичне квадратів різниць між прогнозованими та фактичними значеннями. Внаслідок того, що помилки підносяться до квадрата, значні відхилення отримують більшу вагу в підсумковій оцінці.

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i - \hat{y}_i)^2 \quad (2.2)$$

Середній квадрат відхилення (Root Mean Squared Error, RMSE) – обчислюється як квадратний корінь із середньої квадратичної похибки (MSE).

Це дозволяє представити значення помилки у тих самих одиницях виміру, що й реальні цільові значення.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n (y_i - \hat{y}_i)^2} \quad (2.3)$$

Середня абсолютна відсоткова похибка (Mean Absolute Percentage Error, MAPE) – характеризує точність прогнозування шляхом розрахунку середнього арифметичного абсолютних відхилень, нормованих на фактичні значення та виражених у відсотках.

$$MAPE = \frac{1}{n} \sum_{i=0}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| * 100\% \quad (2.4)$$

Моделювання ШНМ найкраще проводити у модулі Keras, Модуль Keras дуже гнучкий і добре підходить для інноваційних досліджень. Модель ШНМ в Keras складається з послідовності або окремого графа. Створювати моделі в Keras можна з допомогою повністю сконфігурованих модулів: нейронні рівні, оптимізатори, схеми ініціалізації, функції втрат, функції активації та ін. Деякі з цих настроюваних модулів можна об'єднати. Тобто, можна легко додавати нові функції як окремі модулі.

Існує два способи розробки моделі Keras: послідовний і функціональний. Послідовна модель API є найпростішою моделлю, яка містить лінійну кількість шарів, що дозволяє налаштовувати моделі шар за шаром. Функціональний API ідеально підходить для створення складних моделей, які вимагають більшої гнучкості. Це дозволяє визначати моделі, шари функцій яких пов'язані не лише з попереднім і наступним шарами.

Процес навчання ШНМ в Keras складається з таких етапів:

1. Визначення та побудова мережевої моделі за допомогою попередньо визначених шарів у Keras. Функція `Sequential()` дозволяє сформувати ШНМ з послідовності шарів.

2. Компілювання моделі за допомогою методу `model.compile()`.

3. Навчання моделі за допомогою `model.fit()`.

4. Передбачення результату за допомогою `model.predict()`.

Для встановлення оцінки якості навченості моделі використовуємо метод `model.evaluate()`, який повертає значення помилок та показників для моделі у тестовому режимі.

У даному параграфі розглянуто модель LSTM для прогнозування часових рядів нейромережевими методами, показано її переваги і недоліки.

Висновки до розділу 2

1. Розглянуто основні архітектури ШНМ – мережі прямого поширення та рекурентні мережі. Проаналізовані їх види та проведено класифікацію НМ, показано основну структуру багат шарової НМ.

2. Досліджено моделі MLP та LSTM для прогнозування часових рядів нейромережевими методами, показано їх переваги і недоліки.

3. Показано, що саме ці моделі найкраще підходять для прогнозування часових рядів нейромережевими методами.

3 ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ПІДВИЩЕННЯ ТОЧНОСТІ НЕЙРОМЕРЕЖЕВОВОГО ПРОГНОЗУ

3.1 Методика нейромережевого прогнозу при обмеженому обсязі результатів експериментальних досліджень

У переважній більшості випадків отримання результатів експериментальних досліджень є трудомісткою та тривалою задачею, яка часто вимагає як затрат праці дослідника, так і матеріальних затрат, часом доволі значних [32]. Ці труднощі найчастіше прямо пов'язані із необхідним обсягом даних експериментальних досліджень – при зростанні необхідного обсягу даних експериментальних досліджень затрати можуть зростати швидше, ніж лінійно. Крім того, бувають випадки, коли безпосередньо збільшити обсяг результатів експериментальних досліджень неможливо, наприклад, при прогнозуванні деяких подій поблизу початку їх реалізації – тоді значного обсягу результатів експериментальних досліджень просто не існує [32].

Як було показано у [30, 33, 34], на навчальних вибірках малого обсягу ШНМ погано навчаються. Поступове зниження обсягу навчальної вибірки спочатку веде до погіршення якості прогнозу (зростання похибки прогнозу), а потім до зривання процесу навчання ШНМ. При цьому зростає похибка як інтерполяції, так і екстраполяції даних за допомогою традиційних ШНМ.

Тому, як було вказано у Вступі, задача підвищення точності прогнозу за допомогою ШНМ при малих кількостях результатів експериментальних досліджень є актуальною задачею.

У [33] було запропоновано підвищити точність прогнозу за допомогою ШНМ шляхом спільної дії ансамблю двох типів ШНМ – апроксимуючої та прогнозуючої. При цьому на першому етапі апроксимуюча ШНМ навчається на малій кількості результатів експериментальних досліджень. На другому етапі навчена апроксимуюча ШНМ генерує збільшений обсяг даних для навчання прогнозуючої ШНМ, тобто штучно збільшує кількість результатів експериментальних досліджень. На третьому етапі прогнозуюча ШНМ

навчається на цьому штучно збільшеному обсязі даних. На четвертому етапі навчена прогнозуюча ШНМ проводить необхідний прогноз. Через спеціалізацію двох типів ШНМ такий метод, як це було показано у [33], дає змогу підвищити точність прогнозу.

Таким чином, основна ідея методу [33] полягає у штучному збільшенні навчальної вибірки для навчання прогнозуючої ШНМ за допомогою іншої ШНМ. Однак штучно збільшити навчальну вибірку для навчання прогнозуючої ШНМ можна також іншими методами. Зокрема, таке збільшення можна також отримати за рахунок апроксимації обмеженого ряду результатів експериментальних досліджень за допомогою інших методів.

Відповідно до [35, 36], крім використання для апроксимації ШНМ, відомі також інші методи – шляхом вирішення системи рівнянь високого порядку [37], за допомогою поліномів Лагранжа [38], за допомогою регресійного аналізу [35, 36, 39].

При цьому існує можливість, що у деякому конкретному випадку використання якогось конкретного методу забезпечить перевагу над іншими, тобто, у нашому випадку, меншу похибку прогнозу. У даній кваліфікаційній роботі магістра буде досліджено використання перелічених методів апроксимації для попереднього опрацювання обмеженого обсягу результатів експериментальних досліджень. Аналіз проведених досліджень дасть змогу обґрунтовано вирішити, який із перелічених методів можна рекомендувати для подальшого використання.

У даному підрозділі описано метод підвищити точність прогнозу при обмежених навчальних вибірках за допомогою ШНМ шляхом спільної дії ансамблю двох типів ШНМ – апроксимуючої та прогнозуючої.

3.2 Підготовка вхідних даних та вибір інструментальних засобів

Для проведення запропонованих у підрозділі 3.1 досліджень необхідно сформулювати відповідний (коректний) набір результатів експериментальних досліджень. Такий набір повинен відповідати наступним вимогам:

1) основний тренд результатів експериментальних досліджень повинен бути або виключно додатнім або виключно від’ємним;

2) обсяг результатів експериментальних досліджень повинен бути принципово обмеженим – експериментальне отримання проміжних або продовження ряду в обидві сторони повинно бути неможливим.

Як такі результати експериментальних досліджень було вивантажено з реєстру відкритих даних «Єдиної державної електронної бази з питань освіти» [40, 41] кількість осіб, зарахованих на навчання у 2016-2024 роках на спеціальність 122 «Комп’ютерні науки» Національного університету «Львівська політехніка». Ці дані були згруповані відповідно до різних категорій вступників (таблиця 3.1). Результати, отримані згідно таблиці 3.1, поміщено у таблицю 3.2, де просумовано по роках.

Реалізацію програмного коду на мові програмування Python здійснюємо в Google Colaboratory [42]. Це хмарне середовище розробки від Google, призначене для виконання коду та моделювання процесів машинного навчання. Для моделювання нейронної мережі, компілювання та навчання використаємо бібліотеку для машинного навчання TensorFlow та надбудову над нею – модуль Keras.

Для роботи у Google Colaboratory підключаємо бібліотеки, які необхідні для виконання завдання. Цей розділ імпортує всі необхідні бібліотеки Python для обробки даних, побудови та навчання нейронної мережі, оцінки результатів і візуалізації:

1) `numpy (np)` та `pandas (pd)` – для ефективної роботи з числовими масивами та табличними даними;

Таблиця 3.1 – Інформація про осіб, зарахованих на навчання

№	Назва поля	Код	Категорії вибору
1	Назва закладу освіти	97	Національний університет «Львівська політехніка»
2	Назва спеціальності	122	Комп'ютерні науки
3	Основа вступу	40	Повна загальна середня освіта
		520	Молодший спеціаліст
		620	Бакалавр
4	Форми навчання	–	Денна (бюджет)
		–	Денна (контракт)
		–	Заочна (бюджет)
		–	Заочна (контракт)
		–	Вечірня (бюджет)
		–	Вечірня (бюджет)

Таблиця 3.2 – Кількість зарахованих осіб на навчання у 2016-2024 роках

№	Рік вступу	Код спеціальн.	Денна (бюдж.)	Денна (конт.)	Заочна (бюдж.)	Заочна (конт.)	Сума
1	2016	122	261	148	5	44	458
2	2017	122	267	246	8	71	592
3	2018	122	252	337	7	74	670
4	2019	122	276	336	11	67	690
5	2020	122	327	256	11	27	621
6	2021	122	387	302	12	30	731
7	2022	122	405	252	17	23	697
8	2023	122	482	223	18	17	740
9	2024	122	427	91	17	9	544

- 2) matplotlib.pyplot (plt) – для побудови графіків;
- 3) MinMaxScaler (sklearn) – для нормалізації даних;
- 4) tensorflow.keras – основний фреймворк для створення нейронної мережі (Sequential, Dense, SGD, EarlyStopping);
- 5) sklearn.metrics – для розрахунку метрик якості (MSE, MAE, R^2);
- 6) graphviz – для візуалізації структури мережі.

У цьому випадку сервіс Colaboratory буде готовий для проведення досліджень.

У даному підрозділі поставлена задача прогнозу кількості осіб, які будуть зараховані на спеціальність 122 «Комп’ютерні науки» Національного університету «Львівська політехніка».

3.3 Прогноз кількості осіб, які будуть зараховані, за допомогою ШНМ з функцією активації гіперболічний тангенс

Відповідно до рекомендацій [42], для такого прогнозу рекомендована функцією активації гіперболічний тангенс. Функція активації гіперболічний тангенс (tanh) додає нейромережі «нелінійність» та перетворює вхідне число у діапазон від -1 до 1. У таблиці 3.3 подано найкращий результат, який має дуже високий показник коефіцієнта детермінації R^2 . На рисунку 3.1 показано результати прогнозування.

Таблиця 3.3 – Функція активації гіперболічний тангенс

Характеристики навчання	Прогноз на тренувальних даних	Прогноз вступників 2025-2027р
MSE : 408.09 RMSE: 20.20 MAE : 16.34 R^2 : 0.9126	2019: реальне=690, прогноз=687.1 2020: реальне=621, прогноз=653.5 2021: реальне=731, прогноз=730.7 2022: реальне=697, прогноз=677.8 2023: реальне=740, прогноз=711.9 2024: реальне=544, прогноз=559.0	Останнє вікно: [[697 1/1 <hr style="border: 1px solid green; width: 100%;"/> 2025: прогноз = 764 1/1 <hr style="border: 1px solid green; width: 100%;"/> 2026: прогноз = 809 1/1 <hr style="border: 1px solid green; width: 100%;"/> 2027: прогноз = 648



Рисунок 3.1 – Результат прогнозування

Як видно із рисунку 3.1, така ШНМ практично не має властивості узагальнення результатів експерименту.

У даному підрозділі показано, що функція активації гіперболічний тангенс не годиться для прогнозу при обмеженнях кількості даних.

3.4 Прогноз кількості осіб, які будуть зараховані, за допомогою ШНМ з функцією активації лінійних елементів виправлення

У [43] рекомендовано для даних умов використати функцію активації лінійних елементів виправлення. Результати подано у таблиці 3.4. Хоча коефіцієнт детермінації R^2 високий, судячи з рисунку 3.2, така ШНМ не має узагальнюючих властивостей, її не слід використовувати для прогнозування. У таблиці 3.5 подано результати аналогічного прогнозу ШНМ та збільшеною кількістю нейронів схованого шару, а на рисунку 3.3 – графік такого прогнозу. Хоч коефіцієнт детермінації R^2 зріс, однак із порівняння рисунків 3.2 і 3.3 видно, що така ШНМ перетворюється у асоціативну пам'ять. Це повністю відповідає висновкам, зробленим у [19].

Таблиця 3.4 – Функція активації лінійних елементів виправлення

Характеристики навчання	Прогноз на тренувальних даних	Прогноз вступників 2025-2027р
MSE : 496.10 RMSE: 22.27 MAE : 15.71 R ² : 0.8938	2019: реальне=690, прогноз=690.9 2020: реальне=621, прогноз=621.6 2021: реальне=731, прогноз=747.9 2022: реальне=697, прогноз=693.7 2023: реальне=740, прогноз=698.8 2024: реальне=544, прогноз=575.2	Останнє вікно: [[697 1/1 2025: прогноз = 814 1/1 2026: прогноз = 814 1/1 2027: прогноз = 653

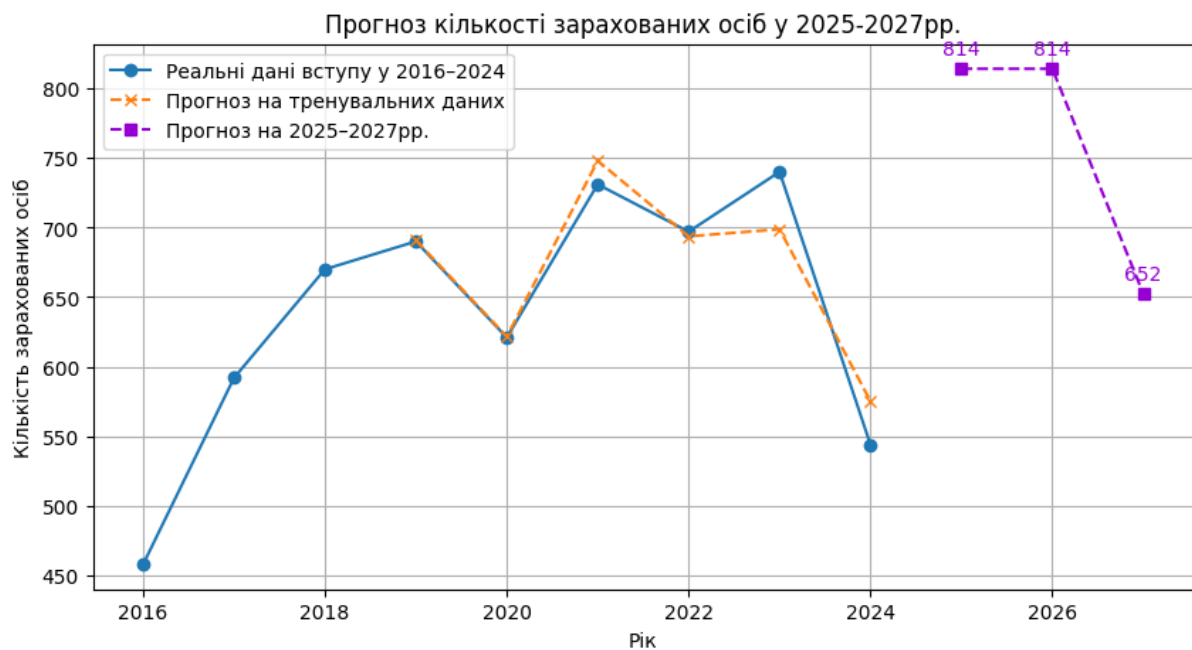


Рисунок 3.2 – Результат прогнозування MLP (3->3->1)

Таблиця 3.5 – Функція активації лінійних елементів виправлення

Характеристики навчання	Прогноз на тренувальних даних	Прогноз вступників 2025-2027р
MSE : 19.96 RMSE: 4.47 MAE : 3.67 R ² : 0.9957	2019: реальне=690, прогноз=691.1 2020: реальне=621, прогноз=627.5 2021: реальне=731, прогноз=733.3 2022: реальне=697, прогноз=695.1 2023: реальне=740, прогноз=732.1 2024: реальне=544, прогноз=546.2	Останнє вікно: [[697 1/1 2025: прогноз = 814 1/1 2026: прогноз = 814 1/1 2027: прогноз = 773



Рисунок 3.3 – Результат прогнозування MLP (3->5->1)

У даному підрозділі показано, що функція активації лінійних елементів виправлення практично не годиться для прогнозу при таких обмеженнях кількості даних і для такого розкиду даних.

3.5 Прогноз кількості осіб, які будуть зараховані, при штучному збільшенні кількості результатів експериментальних досліджень

У [33, 44] запропоновано при малій кількості результатів експериментальних досліджень штучно збільшити їх кількість шляхом апроксимації за допомогою ШНМ. Але хороші можливості апроксимації має метод регресійного аналізу [35, 36]. Тому дослідимо можливості прогнозу за допомогою ШНМ кількості осіб, які будуть зараховані, при попередній апроксимації результатів експериментальних досліджень шляхом регресійного аналізу за допомогою вказаних у таблиці 3.6 моделей, які візуально представлені на рисунку 3.4.

Критерії вибору моделі:

1) R^2 (коефіцієнт детермінації) – частка варіації залежної змінної, що пояснюється моделлю; значення $> 0,7$ вважається хорошим.

2) Скоригований R^2 – враховує кількість параметрів моделі, запобігає перенаванчання; від’ємні значення свідчать про неадекватність моделі.

3) RMSE (Root Mean Square Error) – середньоквадратична похибка в одиницях вимірювання; менше значення – краще.

4) MAPE (Mean Absolute Percentage Error) – середня абсолютна відсоткова похибка; $MAPE < 10\%$ вважається прийнятним.

5) AIC (Akaike Information Criterion) – інформаційний критерій, що враховує точність і складність моделі; менше значення – краще.

Таблиця 3.6 – Рівняння моделей

№	Назва моделі	Рівняння
1	Лінійна модель	$y = 564,5 + 14,7x$
2	Степенева модель	$y = 518,6 * x^{0,14}$
3	Експоненціальна модель	$y = 557,5 * e^{0,025x}$
4	Квадратична (поліноміальна) модель	$y = -11,0x^2 + 124,4x + 363,5$
5	Логарифмічна	$y = 523,5 + 80,6 * \ln(x)$
6	Кубічна модель	$y = -0,46x^3 - 4,0x^2 + 95x + 394$

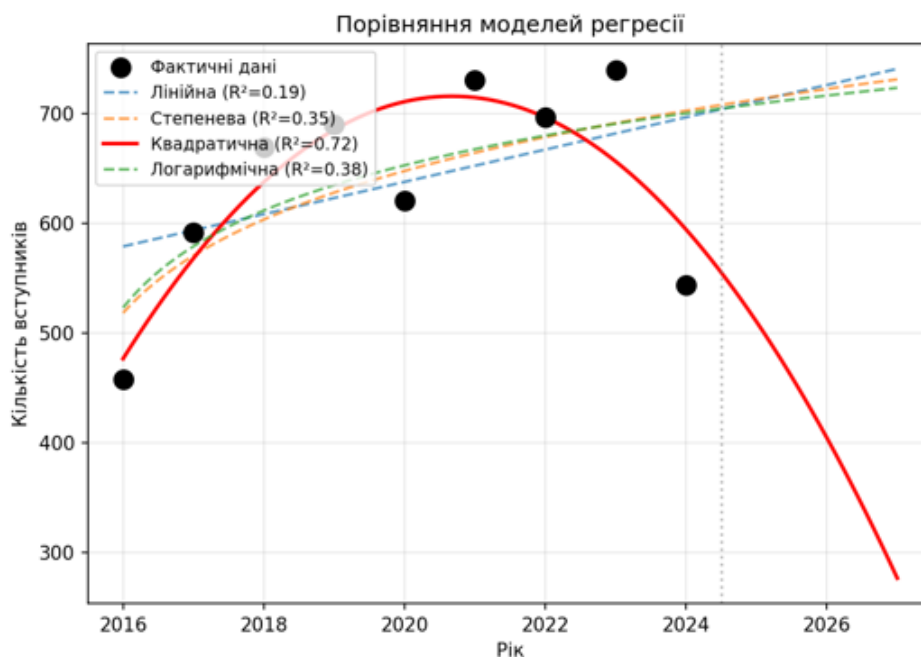


Рисунок 3.4 – Візуалізація моделей таблиці 3.6

Квадратична модель продемонструвала найкращі показники якості: $R^2 = 0,716$, $MARE = 5,65\%$, $AIC = 75,30$. Експоненціальна та лінійна моделі виявилися неадекватними для даного часового ряду. Візуально апроксимація за допомогою квадратичної моделі подано на рисунку 3.5, а порівняння за вказаними вище критеріями подано на рисунках 3.6 і 3.7. З рисунка 3.5 видно, що квадратична модель має узагальнюючі властивості, тому вона рекомендується для практичного застосування.

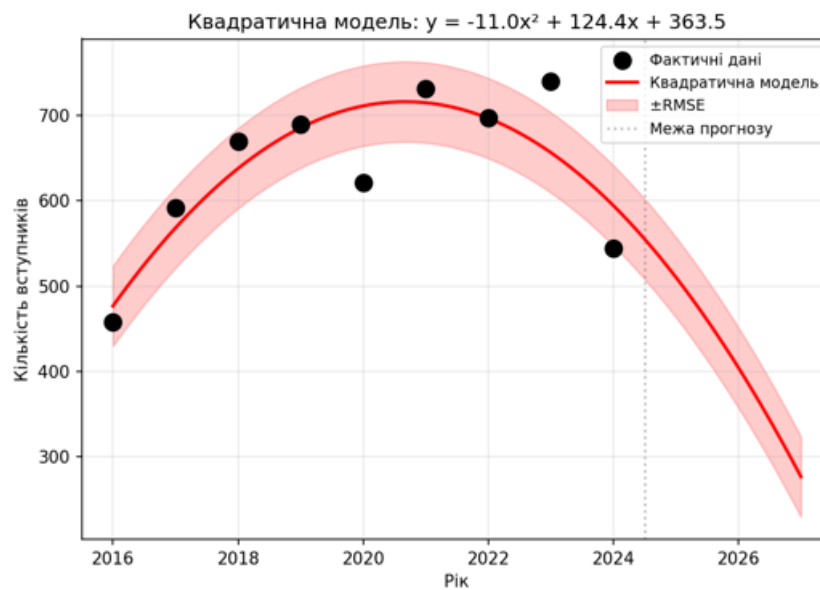


Рисунок 3.5 – Апроксимація за допомогою квадратичної моделі

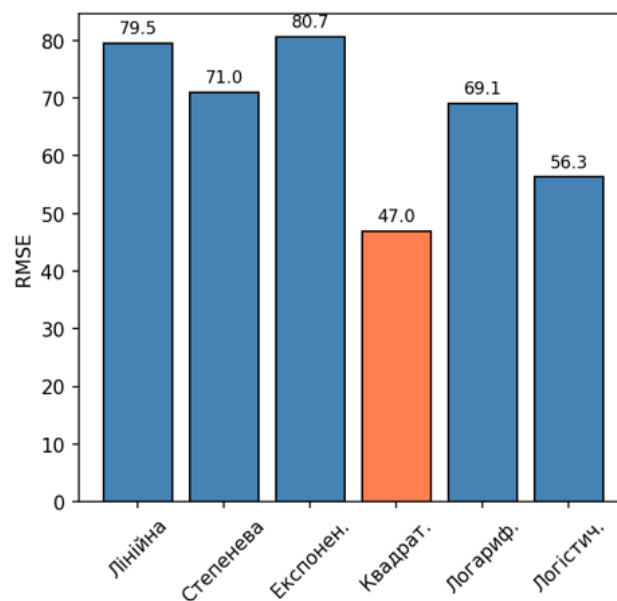


Рисунок 3.6 – Середньоквадратична похибка моделей

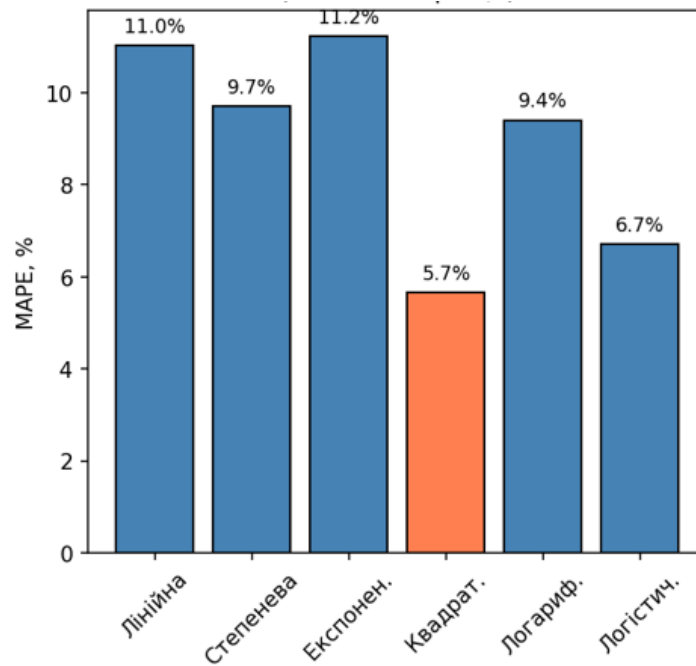


Рисунок 3.7 – Середня абсолютна похибка моделей

Результати прогнозування на збільшеній за рахунок квадратичної апроксимації у 10 разів навчальній вибірці подано на рисунку 3.8.

Як видно з рисунку 3.8, прогноз за результатами апроксимації результатів експериментальних досліджень методом регресійного аналізу найкраще узагальнює дані експериментальних досліджень, при цьому отримано такі характеристики мережі LSTM:

1) на тренувальному наборі даних:

$MSE=207,6404$; $RMSE=14,4097$; $MAE=12,8912$; $R^2=0,9302$;

2) на валідаційному наборі даних:

$MSE=282,4312$; $RMSE=16,8057$; $MAE=14,4384$; $R^2=0,5203$.

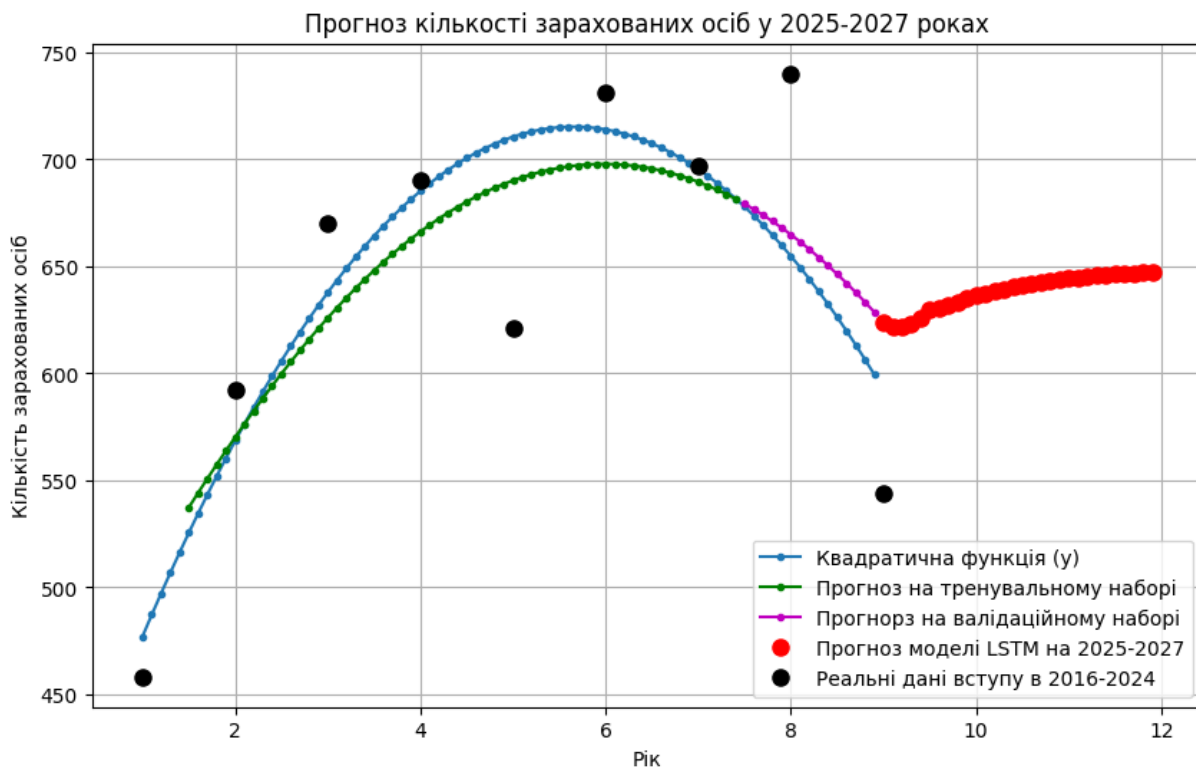


Рисунок 3.8 – Результат прогнозування для квадратичної апроксимації

Отримані результати прогнозу подані у таблиці 3.7.

Таблиця 3.7 – Результати прогнозу

Рік	Прогноз (квадратична)	Прогноз (кубічна)
2025	636	638
2026	644	648
2027	647	651

Результати прогнозування в аналогічних умовах при використанні кубічної апроксимації подано на рисунку 3.9.

При цьому отримано такі характеристики мережі LSTM:

1) на тренувальному наборі даних:

MSE=206,3482; RMSE=14,3648; MAE=12,9058; $R^2=0,9381$;

2) на валідаційному наборі даних:

MSE=376,9902; RMSE=19,4162; MAE=16,4144; $R^2=0,5173$.

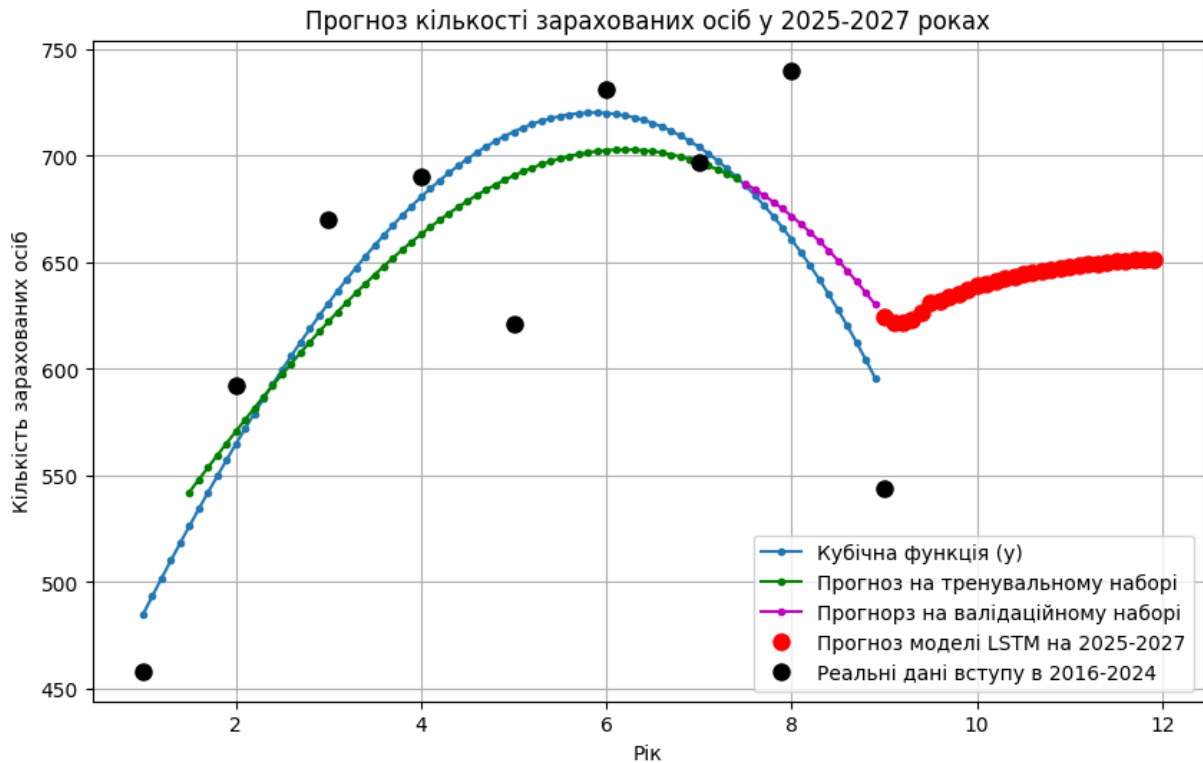


Рисунок 3.9 – Результат прогнозування для кубічної апроксимації

Отримані результати прогнозу практично співпадають (див. таблицю 3.7) з незначною похибкою у 2–4 студенти. Це показує, що метод штучного збільшення кількості результатів експериментальних досліджень за допомогою регресійного аналізу дає високу стійкість прогнозу.

У даному підрозділі показано, що при штучному збільшенні кількості результатів експериментальних досліджень ШНМ отримує узагальнюючі властивості та забезпечує найкращий прогноз.

Висновки до розділу 3

1. При обмежених навчальних вибірках та великому розкиді даних експериментальних досліджень прогнозування часових рядів даних відбувається з великою похибкою, тому вимагає подальших досліджень.

2. Для проведення таких досліджень поставлена задача прогнозу кількості осіб, які будуть зараховані на спеціальність 122 «Комп'ютерні науки» Національного університету «Львівська політехніка».

3. За результатами досліджень показано, що результати прогнозу за допомогою мережі MLP з рекомендованими функціями активації гіперболічний тангенс та активації лінійних елементів виправлення практично не годиться для прогнозу при таких обмеженнях кількості даних і для такого розкиду даних.

4. За результатами досліджень показано, що при штучному збільшенні кількості результатів експериментальних досліджень за допомогою регресійного аналізу мережі LSTM отримує узагальнюючі властивості та забезпечує найкращий прогноз.

5. За результатами досліджень показано, що прогноз за п. 4 є стійким і майже не залежить від порядку апроксимуючого полінома.

ВИСНОВКИ

1. Проведено огляд структури типової штучної нейронної мережі, функцій її складових, основних видів нейронних мереж та основні сфери їх застосування.

2. Розглянуто основні архітектури нейронних мереж – мережі прямого поширення та рекурентні мережі. Проаналізовані їх види та проведено класифікацію НМ.

3. Проведений аналіз моделей MLP та LSTM для прогнозування часових рядів за допомогою метрик оцінювання якості показав, що саме ці моделі найкраще підходять для прогнозування часових рядів нейромережевими методами.

4. Проведений аналіз результатів прогнозування часових рядів результатів експериментальних досліджень за допомогою ШНМ показав, що при обмежених навчальних вибірках та великому розкиді даних прогнозування відбувається з великою похибкою, тому вимагає подальших досліджень.

5. Для проведення таких досліджень поставлена задача прогнозу кількості осіб, які будуть зараховані на спеціальність 122 «Комп'ютерні науки» Національного університету «Львівська політехніка».

6. За результатами досліджень показано, що результати прогнозу за допомогою мережі MLP з рекомендованими функціями активації гіперболічний тангенс та активації лінійних елементів виправлення практично не годиться для прогнозу при таких обмеженнях кількості даних і для такого розкиду даних.

7. За результатами досліджень показано, що при штучному збільшенні кількості результатів експериментальних досліджень за допомогою регресійного аналізу мережі LSTM отримує узагальнюючі властивості та забезпечує найкращий прогноз.

8. За результатами досліджень показано, що такий прогноз є стійким і майже не залежить від порядку апроксимуючого полінома.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Румянцев Г. Біла книга з регулювання ШІ в Україні: бачення Мінцифри. Версія для консультацій. URL: <https://surl.li/sssodw>.
2. Українці та штучний інтелект: як ми використовуємо ШІ у житті й покупках / Kantar Panel Ukraine. 2025. URL: <https://www.kantar.com/ua/inspiration/consumers/shopping-with-ai>.
3. Регулювання ШІ в Україні: що вже зроблено. 2024. URL: <https://unba.org.ua/news/9568-regulyuvannya-shi-v-ukraini-sho-vzhe-zrobleno.html>.
4. Ситник Л. Г. Основи штучного інтелекту : конспект лекцій / ред. О. Б. В'юненко. Суми, 2025. 74 с.
5. Методи та системи штучного інтелекту : конспект лекцій для здобувачів другого (магістерського) рівня / І. М. Соловська, О. П. Русу, В. В. Педяш. Одеса : Міжнар. гуманіт. ун-т, 2024. 142 с.
6. Пулеко І. Штучний інтелект в задачах кібербезпеки. URL: <https://learn.ztu.edu.ua/course/view.php?id=4117#section-0>.
7. Нейронні мережі. Ч. 1 : Вступ до нейронних мереж : навч. посіб. / уклад. І. В. Федорін ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2023. 225 с.
8. Розум. Вікіпедія. 2023. URL: <https://surl.li/mozxwo>.
9. Мислення. Вікіпедія. 2023. URL: <https://surl.li/lodwsj>.
10. Згуровський М. З. Вступ до штучного інтелекту: тенденції, виклики, перспективи : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2025. 107 с.
11. Харченко В. О. Основи машинного навчання : навч. посіб. Суми : СумДУ, 2023. 264 с.
12. Згуровський М. З. Вступ до штучного інтелекту: тенденції, виклики, перспективи : навч. посіб. для здобувачів ступеня бакалавра, магістра за спец. 122 (F3) Комп'ютерні науки та 124 Системний аналіз (F4 Системний аналіз та наука про дані) / М. З. Згуровський ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2025. 107 с.

13. Alippi C., Ferrero A., Piuri V. Artificial Intelligence for Instruments & Applications. IEEE Instrumentation and Measurement Magazine. 1998. June. P. 9–17.

14. Новотарський М. А., Нестеренко Б. Б. Штучні нейронні мережі: обчислення. Праці Інституту математики НАН України. Т. 50. Київ : Ін-т математики НАН України, 2004. 408 с.

15. Кононюк А. Ю. Нейронні мережі і генетичні алгоритми. Київ : Корнійчук, 2008. 446 с.

16. Штучний інтелект. Вступний курс : навч. посіб. / О. С. Звенігородський та ін. Київ : ДУТ, 2022. 193 с.

17. Батіна О. А. Технології створення освітніх комп'ютерних ігор та проектування доповненої реальності : конспект лекцій : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 236 с.

18. Крамаренко Я. Чим нейромережа відрізняється від штучного інтелекту? 2023. URL: <https://www.gncrypto.news/ua/news/neural-network-vs-ai-whats-the-difference>.

19. Головка В. А. Нейроінтелект: теорія та застосування. Кн. 1 : Організація та навчання нейронних мереж із прямими та зворотними зв'язками. Брест : БПІ, 1999. 264 с.

20. Головка В. А. Нейроінтелект: теорія та застосування. Кн. 2 : Самоорганізація, відмовостійкість та застосування нейронних мереж. Брест : БПІ, 1999. 228 с.

21. Штучні нейронні мережі. URL: <https://learn.ztu.edu.ua/mod/resource/view.php?id=17833>.

22. Кожем'яко В. П., Дмитрук В. В., Белік Н. В. Наука і технічна творчість в навчальному процесі (від абітурієнта до аспіранта). Ч. 2 : Основи сучасної бази знань людства: навч. посіб. Вінниця : ВНТУ, 2008. 215 с.

23. Терейковський І. А., Бушуєв Д. А., Терейковська Л. О. Штучні нейронні мережі: базові положення: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2022. 123 с.

24. Гавриленко С. Ю. Машинне навчання : конспект лекцій / С. Ю. Гавриленко. Харків : НТУ «ХПІ», 2024. 232 с.
25. Федорін І. В. Методи та технології обчислювального інтелекту : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2022. 314 с.
26. Субботін С. О. Нейронні мережі : теорія та практика : навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с.
27. The Neural Network Zoo / The Asimov Institute. 2019. URL: <https://www.asimovinstitute.org/neural-network-zoo>.
28. Прогнозування з використанням нейромережових моделей. 2023. URL: <https://surl.li/magpks>.
29. Застосування технологій штучного інтелекту у фізиці високих енергій : навч. посіб. / Р. В. Єрмоленко та ін. Київ, 2024. 152 с.
30. Kroese B. An Introduction to Neural Networks. Amsterdam : University of Amsterdam, 1996. 120 p.
31. Кубик Л. Я. Оцінка ефективності архітектури LSTM-моделі у прогнозуванні динаміки температури повітря. Modern Perspectives on Global Scientific Solutions : proc. of the 5th Int. Sci. and Practical Conf., Oct. 13-15, 2025. Bergen : European Open Science Space, 2025. P. 82–85. URL: <https://www.eoss-conf.com/en/archive/modern-perspectives-on-global-scientific-solutions-13-10-25>. DOI: 10.70286/EOSS-13.10.2025.
32. Кочан О. В. Методи і засоби підвищення точності вимірювання температури термоелектричними перетворювачами з неоднорідними термопарами : дис. ... д-ра техн. наук. Львів, 2020.
33. Турченко В. Нейромережові методи і засоби підвищення ефективності дистрибутивних мереж збору та обробки сенсорних даних : дис. ... канд. техн. наук. Львів : Нац. ун-т «Львівська політехніка», 2001.
34. Васильків Н. Підвищення точності вимірювання температури термопарами в процесі експлуатації : дис. ... канд. техн. наук. Львів, 2010.
35. Економетрика : навч. посіб. / В. О. Єрмоленко та ін. Тернопіль : Підручники і посібники, 2023.

36. Методичні рекомендації для виконання комплексних практичних індивідуальних завдань з курсу «Економетрика» / К. М. Березька та ін. Тернопіль : ТНЕУ, 2017.

37. Moin P. Fundamentals of engineering numerical analysis. Cambridge University Press, 2012.

38. Методи та алгоритми комп'ютерних обчислень. Теорія і практика : підручник / Р. Н. Кветний та ін., 2023.

39. Крилик Л. В., Богач І. В., Прокопова М. О. Обчислювальна математика. Інтерполяція та апроксимація табличних даних. Вінниця : ВНТУ, 2013.

40. Єдина державна електронна база з питань освіти (ЄДЕБО). URL: <https://registry.edbo.gov.ua/opendata/entrant>.

41. Кубик Л. Я. Порівняння ефективності архітектур LSTM та MLP для прогнозування коротких часових рядів. Scientific Exploration: Bridging Theory and Practice : proc. of the 5th Int. Sci. and Practical Conf., Oct. 20-22, 2025. Berlin : European Open Science Space, 2025. P. 79–83. URL: <https://www.eoss-conf.com/en/archive/scientific-exploration-bridging-theory-and-practice-20-10-25>. DOI: 10.70286/EOSS-20.10.2025.

42. Google Colaboratory Instruments. URL: <https://colab.research.google.com>.

43. Обчислювальний інтелект : опорний конспект лекцій для студентів другого ступеня вищої освіти «магістр» / уклад.: В. С. Коваль, Х. В. Лип'яніна-Гончаренко. Тернопіль : ЗУНУ, 2021. 200 с.

44. Савицький Ю. В. Методи та засоби генерації штучних нейронних мереж для вирішення задач прогнозування: Автореф. дис. ... канд. техн. наук: 05.13.15 / Білоруський державний університет інформатики та радіоелектроніки. – Мінськ, 2000. – 22 с.

45. Комар М.П., Саченко А.О., Васильків Н.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2021. 32 с.

Додаток А
Копії публікацій



Proceedings of the 5th International Scientific
and Practical Conference
**"Modern Perspectives on Global Scientific
Solutions"**

October 13-15, 2025
Bergen, Norway

Collection of Scientific Papers

Bergen, 2025



UDC 01.1

Collection of Scientific Papers with the Proceedings of the 5th International Scientific and Practical Conference «Modern Perspectives on Global Scientific Solutions» (October 13-15, 2025, Bergen, Norway). European Open Science Space, 2025. 304 p.

ISBN 979-8-89704-959-2 (series)
 DOI 10.70286/EOSS-13.10.2025



The conference is included in the Academic Research Index ReserchBib International catalog of scientific conferences.



The conference is registered in the database of scientific and technical events of UkrISTEI to be held on the territory of Ukraine (Certificate №556 dated 16.06.2025).



The materials of the conference are publicly available under the terms of the CC BY-NC 4.0 International license.

The materials of the collection are presented in the author's edition and printed in the original language. The authors of the published materials bear full responsibility for the authenticity of the given facts, proper names, geographical names, quotations, economic and statistical data, industry terminology, and other information.

ISBN 979-8-89704-959-2 (series)



© Participants of the conference, 2025
 © Collection of scientific papers, 2025
 © European Open Science Space, 2025

ОЦІНКА ЕФЕКТИВНОСТІ АРХІТЕКТУРИ LSTM- МОДЕЛІ У ПРОГНОЗУВАННІ ДИНАМІКИ ТЕМПЕРАТУРИ ПОВІТРЯ

Кубик Лілія Ярославівна

здобувачка вищої освіти магістерського рівня
0000-0002-3652-6835

Кафедра інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, м. Тернопіль, Україна

Прогнозування часових рядів є ключовим завданням у багатьох галузях: у медичній діагностиці, передбаченні попиту, при ухваленні рішень на фінансових ринках, метеорології тощо. Прогностичних моделей існує безліч, і всі вони виконують завдання передбачення часових рядів – знаходження майбутніх значень залежно від часу [1, с. 24]. Для метеорологічних даних, які характеризуються яскраво вираженою сезонністю та різкими перепадами температури, традиційні статистичні методи часто виявляються недостатньо точними. Сучасні підходи пропонують використання технологій глибокого навчання, зокрема рекурентних нейронних мереж – Recurrent neural network (RNN) та їхньої удосконаленої архітектури мережі довгострокової короткочасної пам'яті – Long short-term memory (LSTM). Це спеціальні архітектури нейронних мереж, які здатні обробляти послідовні дані, для яких важливий хронологічний порядок.

LSTM-мережі здатні зберігати контекст на довгих послідовностях даних завдяки механізмам вентилів (гейтів) [2, с. 97], що робить їх ідеальними для виявлення сезонних коливань та перепадів температури. У статті представлено результати прикладного дослідження з розробки та навчання нейронної мережі типу LSTM для прогнозування середньодобової температури повітря. Розроблена модель використовувала реальні архівні метеорологічні дані за 2024 рік для міста Тернопіль.

Кобилін О. та співавтори описують принцип роботи мережі: LSTM зменшує або збільшує кількість інформації про стан комірки, залежно від потреб. Для цього використовуються структури, що ретельно налаштовуються, які називаються гейтами [2, с. 99]. Гейт – це «брама», яка пропускає або не пропускає інформацію. Гейти складаються з сигмовидного шару нейронної мережі та операції поточкового множення. На виході сигмовидного шару видаються числа від нуля до одиниці, визначаючи скільки відсотків кожної одиниці інформації пропустити далі. Значення «0» означає «не пропустити нічого», значення «1» – «пропустити все» [2, с. 99].

Вихідні дані для дослідження були отримані з платформи Historical Weather API [3] для міста Тернопіль, вибірка, яких охоплювала період з 01.01.2024р. по 31.12.2024р, основним об'єктом аналізу була середньодобова температура повітря.

Згідно з алгоритмом побудови прогнозу, запропонованим Харченком В. О. [1, с. 24], дані пройшли певні етапи попередньої обробки:

1. Агрегацію – погодинні дані агреговано до середньодобових значень (рис.1).



Рис. 1. Динаміка середньодобової температури у місті Тернопіль за 2024р.

2. Нормалізацію – для підвищення стабільності та швидкості навчання нейронної мережі застосовано масштабування за мінімумом і максимумом всіх значень температури у діапазоні від 0 до 1.

3. Формування послідовності «вікна» – використано підхід «багато-до-одного», для прогнозування температури на день формувалася вхідна послідовність із 7 попередніх днів. Загалом отримано 8760 годинних записів (24×365), після агрегації до середньодобової температури – 365 добових записів. Через розмір вікна 7, загальна кількість становила: $365 - 7 + 1 = 359$ послідовностей.

4. Поділ вибірки – масив послідовностей було розділено на навчальну, яка включає перших 80% днів року (з січня по вересень) та тестову, яка включає останніх 20% днів року (з жовтня по грудень) вибірки. З загальної кількості послідовностей визначимо кількість тестових прикладів за формулою $359 \times 0,2 \approx 71,8$ (округлюємо до 71) та тренувальних: $359 - 71 = 288$. Отже, модель здійснювала навчання на 288 тренувальних послідовностях, кожна з яких містить середньодобові значення температури за 7 днів, і перевірялася на 71 тестовій послідовності.

Схема архітектури розробленої LSTM-моделі (рис. 2) представляє послідовність шарів, а саме – вхідного, LSTM, прихованого і вихідного Dense, які обробляють дані для прогнозування:

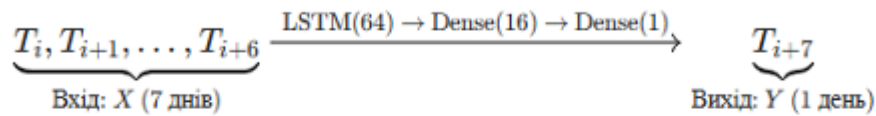


Рис. 2. Спрощена схема розробленої моделі

Шар LSTM виконує основну роботу, витягуючи часові ознаки з 7-ми денної послідовності та повертаючи фінальний вектор стану розміром 64, шари Dense слугують для подальшої нелінійної обробки та проєкції результату в фінальне єдине значення прогнозу.

Навчання моделі показало швидку збіжність, на 60-й епісі мінімальне значення валідаційних втрат становило 0,0036 у масштабованому діапазоні від 0 до 1, що є ознакою високої точності моделі. Спостерігалось відсутність перенавчання, оскільки крива валідаційних втрат не зростала, що підтверджує здатність моделі до узагальнення.

На рис. 3 зображено реальні та прогнозовані значення середньодобової температури для тестової частини даних. Синім кольори виведено реальні дані, а оранжевим – прогноз моделі LSTM.

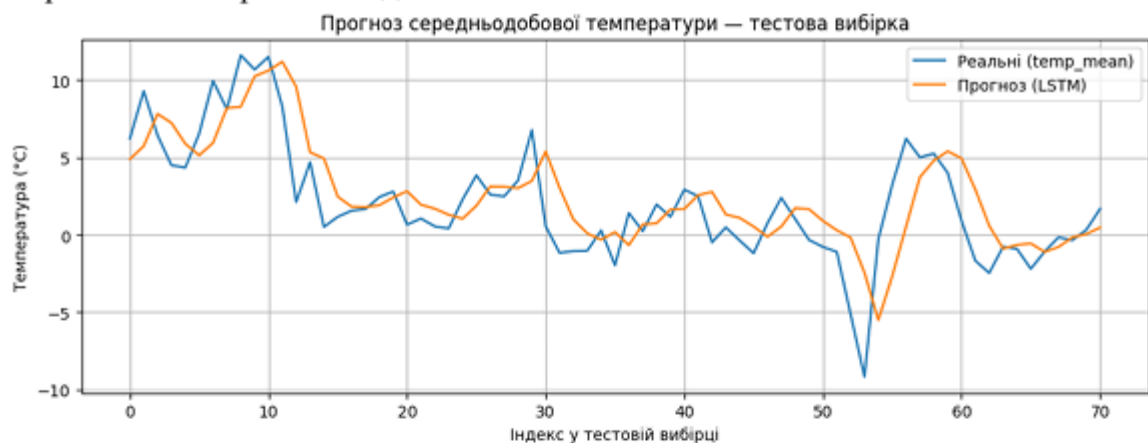


Рис. 3. Порівняння реальних і прогнозованих температур

Дослідження підтвердило високу ефективність LSTM-мереж для прогнозування середньодобової температури повітря у часових рядах. Розроблена модель, сформована на архівних даних 2024 року для м. Тернопіль, продемонструвала точність і надійність її використання, що дозволило ефективно врахувати довгострокові сезонні залежності та короткострокові перепади температур, які є невід'ємною частиною метеорологічних даних. Обраний підхід є ефективним для обробки часових рядів, оскільки LSTM-модель демонструє високу точність, досягаючи середньої абсолютної похибки на тестовій вибірці 0,76°C, що підтверджує її практичну придатність для короткострокового метеорологічного прогнозування.



Список використаних джерел

1. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.
2. Кобилін О., Вечірська І., Кравченко О. Порівняння нейронних мереж типу RNN та LSTM. Information Technology: Computer Science, Software Engineering and Cyber Security. 2024. Вип. 3. С. 97–107. DOI: 10.32782/IT/2024-3-10. Historical Weather API. Open-Meteo. URL: <https://open-meteo.com/en/docs/historical-weather-api> (дата звернення: 05.10.2025).

ОЦІНКА РИЗИКІВ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ПІДПРИЄМСТВА ТА ЇХ ВПЛИВ НА ЕФЕКТИВНІСТЬ УПРАВЛІННЯ

Моргунова Тетяна Іванівна

к.т.н., доцент

Кафедра кримінального аналізу та інформаційних технологій
Одеський державний університет внутрішніх справ, Україна

Цифрова трансформація підприємства – це комплексний процес, що охоплює всі сфери діяльності й передбачає не лише впровадження технологій, а й зміну підходів до управління, бізнес-процесів і корпоративної культури. Її мета – створення доданої цінності завдяки інформаційно-комунікаційним технологіям, аналітиці даних, автоматизації та гнучким моделям управління. Цифровізація є стратегічним чинником, який змінює саму логіку функціонування підприємства.

Цифрова трансформація не полягає в одномоментних змінах, а вимагає стратегічного плану для послідовного й систематичного слідування за тенденціями технологічного поступу [1].

Водночас вона породжує нові ризики, пов'язані з технологічною базою бізнесу та швидкістю змін зовнішнього середовища. На відміну від традиційних економічних, цифрові ризики охоплюють технічні, організаційні, кадрові, інформаційні й правові аспекти та впливають як на операційний, так і на стратегічний рівень управління.

До технічних ризиків належать збої IT-систем, кібератаки та застарілі технології. Фінансові ризики пов'язані з перевищенням витрат і низькою віддачею інвестицій. Організаційні виникають через опір змінам і відсутність узгодженості між підрозділами, кадрові – через нестачу цифрових фахівців і низьку мотивацію персоналу. Інформаційні та правові ризики пов'язані з витоком даних і порушенням норм безпеки, що може призвести до фінансових втрат і репутаційних проблем.



Proceedings of the 5th International Scientific
and Practical Conference
**"Scientific Exploration: Bridging Theory
and Practice"**
October 20-22, 2025
Berlin, Germany

Collection of Scientific Papers

Berlin, 2025



UDC 01.1

Collection of Scientific Papers with the Proceedings of the 5th International Scientific and Practical Conference «Scientific Exploration: Bridging Theory and Practice» (October 20-22, 2025. Berlin, Germany). European Open Science Space, 2025. 297 p.

ISBN 979-8-89704-960-8 (series)

DOI 10.70286/EOSS-20.10.2025



The conference is included in the Academic Research Index ReserchBib International catalog of scientific conferences.



The conference is registered in the database of scientific and technical events of UkrISTEI to be held on the territory of Ukraine (Certificate №558 dated 16.06.2025).



The materials of the conference are publicly available under the terms of the CC BY-NC 4.0 International license.

The materials of the collection are presented in the author's edition and printed in the original language. The authors of the published materials bear full responsibility for the authenticity of the given facts, proper names, geographical names, quotations, economic and statistical data, industry terminology, and other information.

ISBN 979-8-89704-960-8 (series)



© Participants of the conference, 2025
 © Collection of scientific papers, 2025
 © European Open Science Space, 2025

ПОРІВНЯННЯ ЕФЕКТИВНОСТІ АРХІТЕКТУР LSTM ТА MLP ДЛЯ ПРОГНОЗУВАННЯ КОРОТКИХ ЧАСОВИХ РЯДІВ

Кубик Лілія Ярославівна

здобувачка вищої освіти магістерського рівня

<https://orcid.org/0000-0002-3652-6835>

Кафедра інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, м. Тернопіль, Україна

Прогнозування часових рядів та складних системних процесів є однією з найактуальніших проблем сучасної науки та прикладних досліджень. У цьому контексті, найперспективнішим методом прогнозування визнано використання нейронних мереж. Особливо важливим є для таких сфер, як охорона здоров'я (діагностика захворювань за зображеннями (мрт, рентген), прогнозування ефективності лікування, розробка ліків); метеорологія (деталізоване прогнозування погоди та кліматичних змін); інженерія (прогнозування терміну служби обладнання, оптимізація виробничих процесів, моніторинг структурної цілісності); вивчення ринку (прогнозування попиту, аналіз споживчої поведінки та динаміки трендів); фінанси (прогнозування динаміки цін на акції та валюту, оцінка кредитних ризиків, виявлення шахрайства) тощо [1]. Точність передбачень має вирішальне значення для прийняття стратегічних рішень, мінімізації ризиків та оптимізації ресурсів.

Нейронні мережі здобули широке визнання завдяки низці суттєвих переваг над традиційними статистичними та регресійними алгоритмами [1, 2]:

- здатність до нелінійного моделювання, через вміння ефективно фіксувати складні нелінійні взаємозв'язки у даних та здатність узагальнювати будь-яку неперервну і нелінійну функцію, забезпечуючи високу точність, недосяжну для простих статистичних моделей;

- автоматичне вилучення ознак, особливо моделі глибокого навчання, мають унікальну властивість автоматично визначати та витягувати відповідні ознаки без потреби у ручній, трудомісткій інженерії ознак, що значно спрощує етап попередньої підготовки даних та підвищує загальну ефективність моделі;

- стійкість до «зашумлених» даних, завдяки своїй паралельній структурі та здатності до узагальнення, нейронні мережі навчаються ідентифікувати приховані патерни та ігнорувати окремі аномалії (викиди), а не просто запам'ятовувати їх, що робить прогнози більш надійними в умовах реальних даних;

- масштабованість, нейронні мережі здатні використовувати великі масиви інформації для підвищення точності моделювання, що є ключовою перевагою над багатьма класичними алгоритмами, які мають значні обмеження при роботі з великими даними.



Незважаючи на ці переваги, для вирішення задач прогнозування немає універсальної архітектури нейронної мережі. Вибір моделі нейронної мережі є індивідуальним, який залежить від природи процесу та типу даних.

В задачі прогнозування вступу кількість здобувачів освіти на бакалаврат у 2025, 2026 та 2027 роках на спеціальність F3 «Комп'ютерні науки» Національного університету «Львівська політехніка» був вирішальним саме етап вибору моделі нейронної мережі. Розглянемо етапи вирішення даної задачі, які були розроблені на основі запропонованого алгоритму Харченком В. О. [1, с. 24]:

1. Завантаження даних: імпортуємо архівні значення кількості зарахованих студентів у 2016–2024 роках (табл. 1).

2. Масштабування: виконуємо нормалізацію даних в діапазоні від 0 до 1.

3. Формування послідовностей: для прогнозу одного року використовуються значення попередніх трьох років, тобто для прогнозу на 2019 рік використовуються дані за 2016, 2017, 2018 роки.

4. Побудова моделі: обираємо рекурентну нейромережу – LSTM, використовуємо два приховані шари LSTM по 50 нейронів і 2 для запобігання перенавчанню.

5. Навчання: модель навчається на послідовностях, застосовуємо оптимізатор і функцію втрат.

6. Прогнозування: після навчання мережа робить послідовні прогнози на 2025–2027 роки, використовуючи останні 3 відомі значення як вхід.

7. Денормалізація та округлення: прогнози повертаються до реальної шкали (кількість студентів), округлюються до цілих.

8. Візуалізація результатів: будуємо графіки фактичних даних (синім кольором), навченого прогнозу (зеленим) та прогнозу моделі LSTM на 2025–2027 роки (червоним) (рис. 1).

9. Аналіз похибки обраної моделі та визначення адекватності обраної моделі.

Прогноз здійснювався на основі відкритих даних, а саме на кількості зарахованих осіб на навчання для здобуття ступеня бакалавра (основа вступу – повна загальна середня освіта) у 2016–2024 роках, з реєстру Єдиної державної електронної бази з питань освіти [4] (табл. 1).

Таблиця 1 – Кількість зарахованих осіб на бакалаврат у 2016–2024 роках

№ з/п	Рік	Кількість зарахованих осіб
1	2016	402
2	2017	472
3	2018	537
4	2019	586
5	2020	547
6	2021	675
7	2022	644
8	2023	684
9	2024	514

*дані сформовано [4]

Для створення прогнозу спершу було обрано модель глибокого навчання з довгою короткочасною пам'яттю – LSTM, яка ефективно себе проявила у задачі прогнозування погоди [3], але виявилася не ефективною для передбачення кількості майбутніх студентів на спеціальності F3 «Комп'ютерні науки».



Рисунок 1. Візуалізація результатів прогнозу моделі LSTM

Прогнозовані результати моделі LSTM щодо кількості зарахованих осіб на спеціальність F3 «Комп'ютерні науки» наведені в табл. 2.

Таблиця 2 – Результат прогнозу моделі LSTM

№ з/п	Рік	Прогноз кількості зарахованих осіб
1	2025	607
2	2026	608
3	2027	616

*дані сформовано автором

Для прогнозу кількості майбутніх вступників на спеціальність F3 використовувалося лише одне значення на рік з 2016 по 2024 – це всього 9 точок (табл. 1), що утворює надзвичайно короткий часовий ряд для моделі LSTM, який є малоінформативним. Також негативно вплинуло те, що зв'язок між роками не є часовий, а причинно-соціальний, тобто залежить від зміни правил вступу чи війни в країні. Результати оцінювання якості моделі LSTM свідчать про низьку точність прогнозування та обмежену здатність до узагальнення даних. Модель лише частково відтворює тенденції у вибірці, що вказує на недостатню ефективність її навчання. Для підвищення якості прогнозів доцільно збільшити навчальний матеріал використавши допоміжні змінні (наприклад, врахувати кількість студентів, які навчаються на даній спеціальності поквартально).



Наступним кроком дослідження було повернення до етапу 4 та заміна моделі LSTM на класичну архітектуру – MLP, а саме багатошаровий перцептрон, який демонструє кращі показники на коротких рядах.



Рисунок 2. Візуалізація результатів прогнозу моделі MLP

Прогнозовані результати моделі MLP щодо кількості зарахованих осіб на спеціальність F3 «Комп'ютерні науки» наведені в табл. 3.

Таблиця 3 – Результат прогнозу моделі MLP

№ з/п	Рік	Прогноз кількості зарахованих осіб
1	2025	591
2	2026	511
3	2027	644

*дані сформовано автором

Отримані результати оцінювання якості моделі MLP свідчать про достатньо високу узгодженість прогнозованих і фактичних значень. Значення показників похибки підтверджують прийнятний рівень точності моделі та її здатність адекватно відображати основні закономірності у вихідних даних. Модель пояснює близько 73% варіації даних, що є досить хорошим результатом для емпіричних або соціально-економічних показників. Це дає підстави вважати побудовану нейронну мережу придатною для подальшого використання у задачах прогнозування.

Модель LSTM традиційно застосовується для аналізу послідовних даних і має здатність вловлювати часові залежності між спостереженнями. Проте для коротких рядів, зокрема таких, що містять лише дев'ять річних спостережень,



LSTM стикається з низкою проблем. По-перше, обсяг даних є недостатнім для ефективного формування внутрішньої пам'яті моделі. По-друге, у даному випадку спостережуваний процес (кількість вступників на певну спеціальність) не є типовим часовим рядом у класичному розумінні, оскільки залежить не лише від попередніх значень, а й від зовнішніх соціально-економічних чинників (зміни правил вступу, демографічна ситуація, воєнний стан, державне замовлення тощо). Результати моделі засвідчили невисоку якість прогнозування.

На відміну від LSTM, модель MLP показала стабільніші результати на тій самій вибірці. На етапі побудови багат шаровий перцептрон розглядав кожне спостереження як незалежну точку, що дозволило уникнути проблеми недостатності часової інформації. Завдяки своїй простішій архітектурі MLP менш схильна до перенавчання на малих наборах даних і здатна узагальнювати основні закономірності навіть за обмеженої кількості прикладів.

Узагальнюючи результати дослідження, можна зробити висновок, що для коротких історичних вибірок (до десяти спостережень) та даних, у яких часові залежності не є домінуючими, MLP виявляється ефективнішою та стабільнішою моделлю, ніж LSTM. Хоча LSTM має значні переваги у випадках довгих часових рядів, її застосування у задачах із короткими і соціально залежними даними є недоцільним. Отже, для прогнозування кількості вступників на спеціальність F3 «Комп'ютерні науки» у 2025–2027 роках оптимальним рішенням є використання моделі MLP з подальшою адаптацією її структури та параметрів під специфіку освітніх даних.

Список використаних джерел

1. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.
2. Бурлесв, О., Василенко, О., & Іваненко, Р. (2021). ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ В ЕКОНОМІЦІ. *Економіка та суспільство*, (31). <https://doi.org/10.32782/2524-0072/2021-31-27>
3. Кубик Л.Я. Оцінка ефективності архітектури LSTM-моделі у прогнозуванні динаміки температури повітря. Collection of Scientific Papers with the Proceedings of the 5th International Scientific and Practical Conference «Modern Perspectives on Global Scientific Solutions» (October 13-15, 2025, Bergen, Norway). European Open Science Space. 2025. № 57. С. 82–85. URL: <https://www.eoss-conf.com/en/archive/modern-perspectives-on-global-scientific-solutions-13-10-25/>. DOI: 10.70286/EOSS-13.10.2025.001
4. Реєстр суб'єктів освітньої діяльності. – Режим доступу: URL: <https://registry.edbo.gov.ua/opendata/entrant/> (дата звернення: 15.10.2025р.).

Додаток Б

Формування розширеного набору даних

1) Квадратична функція	2) Кубічна функція
<pre>import matplotlib.pyplot as plt import numpy as np a=-11.0 b=124.4 c=363.5 x_list=[] y_list=[] for x in np.arange(1,9,0.1): y = a*x*x + b*x + c x_list.append(x) y_list.append(y) print(f"x={x:.2f} y={y:.2f}") # Створення даних x_array = np.array(x_list) y_array = np.array(y_list) plt.plot(x_array, y_array) plt.xlabel("X") plt.ylabel("Y") plt.title("y = a*x*x + b*x + c") plt.show()</pre>	<pre>import matplotlib.pyplot as plt import numpy as np a=-0.46465 b=-3.99675 c=95.016 d=394.14 x_list=[] y_list=[] for x in np.arange(1,9,0.1): y = a*x*x*x + b*x*x + c*x + d x_list.append(x) y_list.append(y) print(f"x={x:.2f} y={y:.2f}") # Створення даних x_array = np.array(x_list) y_array = np.array(y_list) plt.plot(x_array, y_array) plt.xlabel("X") plt.ylabel("Y") plt.title("y = a*x*x*x + b*x*x + c*x + d") plt.show()</pre>

3) Квадратична функція	4) Кубічна функція
<pre>x=1.00 y=476.90 x=1.10 y=487.03 x=1.20 y=496.94 x=1.30 y=506.63 x=1.40 y=516.10 x=1.50 y=525.35 x=1.60 y=534.38 x=1.70 y=543.19 x=1.80 y=551.78 x=1.90 y=560.15 x=2.00 y=568.30 x=2.10 y=576.23 x=2.20 y=583.94 x=2.30 y=591.43 x=2.40 y=598.70 x=2.50 y=605.75 x=2.60 y=612.58 x=2.70 y=619.19 x=2.80 y=625.58 x=2.90 y=631.75</pre>	<pre>x=1.00 y=484.69 x=1.10 y=493.20 x=1.20 y=501.60 x=1.30 y=509.89 x=1.40 y=518.05 x=1.50 y=526.10 x=1.60 y=534.03 x=1.70 y=541.83 x=1.80 y=549.51 x=1.90 y=557.06 x=2.00 y=564.47 x=2.10 y=571.74 x=2.20 y=578.88 x=2.30 y=585.88 x=2.40 y=592.73 x=2.50 y=599.44 x=2.60 y=606.00 x=2.70 y=612.40 x=2.80 y=618.65 x=2.90 y=624.74</pre>

x=3.00	y=637.70	x=3.00	y=630.67
x=3.10	y=643.43	x=3.10	y=636.44
x=3.20	y=648.94	x=3.20	y=642.04
x=3.30	y=654.23	x=3.30	y=647.47
x=3.40	y=659.30	x=3.40	y=652.73
x=3.50	y=664.15	x=3.50	y=657.81
x=3.60	y=668.78	x=3.60	y=662.72
x=3.70	y=673.19	x=3.70	y=667.45
x=3.80	y=677.38	x=3.80	y=671.99
x=3.90	y=681.35	x=3.90	y=676.35
x=4.00	y=685.10	x=4.00	y=680.52
x=4.10	y=688.63	x=4.10	y=684.50
x=4.20	y=691.94	x=4.20	y=688.28
x=4.30	y=695.03	x=4.30	y=691.87
x=4.40	y=697.90	x=4.40	y=695.25
x=4.50	y=700.55	x=4.50	y=698.44
x=4.60	y=702.98	x=4.60	y=701.42
x=4.70	y=705.19	x=4.70	y=704.19
x=4.80	y=707.18	x=4.80	y=706.75
x=4.90	y=708.95	x=4.90	y=709.09
x=5.00	y=710.50	x=5.00	y=711.22
x=5.10	y=711.83	x=5.10	y=713.13
x=5.20	y=712.94	x=5.20	y=714.82
x=5.30	y=713.83	x=5.30	y=716.28
x=5.40	y=714.50	x=5.40	y=717.52
x=5.50	y=714.95	x=5.50	y=718.52
x=5.60	y=715.18	x=5.60	y=719.29
x=5.70	y=715.19	x=5.70	y=719.83
x=5.80	y=714.98	x=5.80	y=720.12
x=5.90	y=714.55	x=5.90	y=720.18
x=6.00	y=713.90	x=6.00	y=719.99
x=6.10	y=713.03	x=6.10	y=719.55
x=6.20	y=711.94	x=6.20	y=718.87
x=6.30	y=710.63	x=6.30	y=717.93
x=6.40	y=709.10	x=6.40	y=716.73
x=6.50	y=707.35	x=6.50	y=715.28
x=6.60	y=705.38	x=6.60	y=713.56
x=6.70	y=703.19	x=6.70	y=711.58
x=6.80	y=700.78	x=6.80	y=709.34
x=6.90	y=698.15	x=6.90	y=706.82
x=7.00	y=695.30	x=7.00	y=704.04
x=7.10	y=692.23	x=7.10	y=700.97
x=7.20	y=688.94	x=7.20	y=697.63
x=7.30	y=685.43	x=7.30	y=694.01
x=7.40	y=681.70	x=7.40	y=690.11
x=7.50	y=677.75	x=7.50	y=685.92
x=7.60	y=673.58	x=7.60	y=681.44
x=7.70	y=669.19	x=7.70	y=676.67
x=7.80	y=664.58	x=7.80	y=671.60
x=7.90	y=659.75	x=7.90	y=666.24
x=8.00	y=654.70	x=8.00	y=660.58
x=8.10	y=649.43	x=8.10	y=654.61
x=8.20	y=643.94	x=8.20	y=648.34
x=8.30	y=638.23	x=8.30	y=641.76
x=8.40	y=632.30	x=8.40	y=634.86
x=8.50	y=626.15	x=8.50	y=627.66
x=8.60	y=619.78	x=8.60	y=620.13
x=8.70	y=613.19	x=8.70	y=612.29
x=8.80	y=606.38	x=8.80	y=604.13
x=8.90	y=599.35	x=8.90	y=595.64

Додаток В

Програмна реалізація прогнозу за допомогою штучної нейронної мережі

```
# Модель LSTM: апроксимація поліноміальною функцією - датасет -> часовий ряд
-> тренування/валідація -> LSTM -> прогноз + метрики
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.metrics import r2_score
from math import sqrt
import random
import os

# Фіксуємо seed для генератора випадкових чисел, щоб результати були
однаковими при кожному запуску
seed = 42
np.random.seed(seed)
tf.random.set_seed(seed)
random.seed(seed)
os.environ['PYTHONHASHSEED'] = str(seed)

# 1. Створення штучного набору даних
# Генерація даних на основі кубічної (поліноміальної) функції
a = -0.46465
b = -3.99675
c = 95.016
d = 394.14
x_list = []
y_list = []

for x in np.arange(1, 9, 0.1):
    y = a*x*x*x + b*x*x + c*x + d
    x_list.append(x)
    y_list.append(y)

# Створення DataFrame
df = pd.DataFrame({"x": np.round(x_list,2), "y": np.array(y_list)})

# Виведення набору даних
print("ЗГЕНЕРОВАНИЙ НАБІР ДАНИХ (перші 10 рядків)")
print(df.head(10))
print("... всього рядків:", len(df))
```

```

# 2. Підготовка часового ряду для LSTM
# Ми будемо використовувати лише значення у як одновимірний часовий ряд
series = df["y"].values.reshape(-1,1)

# Масштабування до діапазону [0,1]
# LSTM дуже чутливі до масштабу даних, тому це обов'язковий крок
scaler = MinMaxScaler()
series_scaled = scaler.fit_transform(series)

# Функція для створення контрольованих (supervised) зразків за допомогою
ковзного вікна
def create_sequences(data, window):
    X, y = [], []
    for i in range(window, len(data)):
        X.append(data[i-window:i, 0])
        y.append(data[i, 0])
    X = np.array(X)
    y = np.array(y)
    return X, y

# Параметри
window = 5          # кількість попередніх часових кроків, що
                    # використовуються для передбачення наступного
n_future = 3       # скільки майбутніх кроків прогнозувати
test_ratio = 0.2   # частка даних для тестування (валідації)

X, y_ts = create_sequences(series_scaled, window)
# Зміна форми для подачі в LSTM: (зразки, часові кроки, ознаки)
X = X.reshape((X.shape[0], X.shape[1], 1))

# Розділення на тренувальну та валідаційну вибірки
split_idx = int(len(X) * (1 - test_ratio))
X_train, X_val = X[:split_idx], X[split_idx:]
y_train, y_val = y_ts[:split_idx], y_ts[split_idx:]

print(f"\nВсього зразків: {len(X)}, Тренувальні: {len(X_train)},
Валідаційні: {len(X_val)}")

# 3. Побудова моделі LSTM
model = Sequential([
    # Шар LSTM з 64 нейронами, функція активації tanh (стандарт для LSTM)
    LSTM(64, activation='tanh', input_shape=(window,1)),
    # Вихідний шар з 1 нейроном (прогнозуємо 1 значення)
    Dense(1)
])
model.compile(optimizer='adam', loss='mse', metrics=['mae'])
model.summary()

```

```

# 4. Навчання
from tensorflow.keras.callbacks import EarlyStopping
# Зупинка навчання, якщо помилка на валідації не зменшується протягом 20
епох
es = EarlyStopping(monitor='val_loss', patience=20,
restore_best_weights=True, verbose=1)

history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=500,
    batch_size=8,
    callbacks=[es],
    verbose=1
)

# Необов'язково: побудова графіка історії навчання
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='втрати на тренуванні (train loss)')
plt.plot(history.history['val_loss'], label='втрати на валідації (val
loss)')
plt.legend()
plt.title('Історія навчання')
plt.xlabel('епоха')
plt.ylabel('mse (середньоквадратична помилка)')
plt.grid()
plt.show()

# 5. Оцінка на тренувальних та валідаційних даних (зворотне перетворення до
початкового масштабу)
def inv_transform(arr_scaled):
    return scaler.inverse_transform(arr_scaled.reshape(-1,1)).flatten()

train_pred_scaled = model.predict(X_train)
val_pred_scaled = model.predict(X_val)

train_pred = inv_transform(train_pred_scaled)
val_pred = inv_transform(val_pred_scaled)

y_train_orig = inv_transform(y_train)
y_val_orig = inv_transform(y_val)

# Функція оцінки
def metrics(true, pred):
    mse = mean_squared_error(true, pred)
    rmse = sqrt(mse)
    mae = mean_absolute_error(true, pred)
    r2 = r2_score(true, pred) # Розрахунок R^2
    return mse, rmse, mae, r2

```

```

# Розрахунок метрик для тренувальної та валідаційної вибірок
train_mse, train_rmse, train_mae, train_r2 = metrics(y_train_orig,
train_pred)
val_mse, val_rmse, val_mae, val_r2 = metrics(y_val_orig, val_pred)

print("\nМЕТРИКИ")
print(f"Train: MSE={train_mse:.4f}, RMSE={train_rmse:.4f},
MAE={train_mae:.4f}, R2={train_r2:.4f}")
print(f" Val : MSE={val_mse:.4f}, RMSE={val_rmse:.4f}, MAE={val_mae:.4f},
R2={val_r2:.4f}")

# 6. Показ зразків train/val (початковий масштаб)
# Побудова DataFrame для відображення: для кожного зразка показуємо вхідне
вікно та ціль
def build_samples_df(X_arr, y_arr, prefix=''):
    rows = []
    for i in range(len(X_arr)):
        window_vals_scaled = X_arr[i].reshape(-1,1)
        window_vals = inv_transform(window_vals_scaled.flatten())

        # --- передбачення ---
        pred_scaled = model.predict(X_arr[i].reshape(1, window, 1))[0][0]
        pred_original = inv_transform(np.array([pred_scaled]))[0]

        # --- ціль (зворотне перетворення) ---
        target_original = inv_transform(np.array([y_arr[i]]))[0]

        rows.append({
            f'{prefix}вхідне_вікно': np.round(window_vals,2).tolist(),
            f'{prefix}ціль': round(target_original,2),
            f'{prefix}прогноз': round(pred_original,2)
        })
    return pd.DataFrame(rows)

print("\nЗразки рядків навчання (оригінальний масштаб)")
display_train_df = build_samples_df(X_train, y_train, prefix='train_')
print(display_train_df.head(6).to_string(index=False))

print("\nЗразки рядків валідації (оригінальний масштаб)")
display_val_df = build_samples_df(X_val, y_val, prefix='val_')
print(display_val_df.head(6).to_string(index=False))

# 7. Прогноз наступних n_future кроків
# Ми будемо ітеративно передбачати наступні n_future кроків, використовуючи
останнє доступне вікно з ряду
last_window = series_scaled[-window:].flatten().copy()
future_preds_scaled = []
current_window = last_window.copy()

```

```

for i in range(n_future):
    # Прогноз одного кроку вперед
    pred_scaled = model.predict(current_window.reshape(1,window,1))[0,0]
    future_preds_scaled.append(pred_scaled)
    # Зсув вікна: видаляємо найстаріший елемент, додаємо новий прогноз
    current_window = np.concatenate([current_window[1:], [pred_scaled]])

future_preds = inv_transform(np.array(future_preds_scaled))

# Підготовка позицій x для майбутніх точок (останній x у df був 8.0, крок 0.1)
last_x = df['x'].iloc[-1]
future_x = [round(last_x + (i+1)*0.1, 2) for i in range(n_future)]

forecast_df = pd.DataFrame({
    "x": future_x,
    "forecast_y": np.round(future_preds,2)
})

print("\nПРОГНОЗ (наступні кроки)")
print(forecast_df.to_string(index=False))

# 8. Графік оригінального ряду, передбачення train/val, прогноз
plt.figure(figsize=(10,6))
# оригінал
plt.plot(df['x'], df['y'], label='Кубічна функція (y)', marker='o',
markersize=3)

# вирівнювання передбачень до їх цільових індексів
# обчислення індексів для передбачень: цілі починаються з індексу = window
indices = np.arange(window, window + len(y_ts))

# індекси прогнозів навчання
train_idx = indices[:len(y_train)]
val_idx = indices[len(y_train):]

plt.plot(df['x'].iloc[train_idx],
inv_transform(train_pred_scaled).flatten(), 'g.-', label='Прогноз на
тренувальному наборі')
plt.plot(df['x'].iloc[val_idx], inv_transform(val_pred_scaled).flatten(),
'm.-', label='Прогноз на валідаційному наборі')

# точки прогнозу
plt.plot(forecast_df['x'], forecast_df['forecast_y'], 'ro', markersize=8,
label='Прогноз моделі LSTM на 2025-2027')

# РЕАЛЬНІ ДАНІ ПРО ЗАРАХУВАННЯ
real_y = [458, 592, 670, 690, 621, 731, 697, 740, 544]
real_x = list(range(1, 10)) # 1..9

```

```
plt.plot(real_x, real_y, 'ko', markersize=8, label='Реальні дані вступу в  
2016-2024')  
  
plt.xlabel('Рік')  
plt.ylabel('Кількість зарахованих осіб')  
plt.title('Прогноз кількості зарахованих осіб у 2025-2027 роках')  
plt.legend()  
plt.grid()  
plt.show()
```