

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки

«Затверджую»  
Зав. кафедри теоретичної та  
прикладної системотехніки  
д.т.н., проф. С. І. Шматков  
«\_\_» \_\_\_\_\_ 2024 р.

## Пояснювальна записка

до кваліфікаційної роботи  
бакалавра

на тему: «**Автоматизована система аналізу та прогнозування фінансових  
ринків на основі машинного навчання**»

Захищено на засіданні  
Атестаційної комісії № 44  
протокол № \_\_ від \_\_.06.2024 р.  
Оцінка \_\_\_\_ / \_\_\_\_\_  
Голова Атестаційної комісії  
\_\_\_\_\_ **СКОБ Ю. О.**  
(підпис) (прізвище та ініціали)

Виконала:  
студентка 4 курсу, групи КУ-41  
Галузь знань: 15 – Автоматизація та  
приладобудування  
Спеціальність: 151 – Автоматизація та  
комп'ютерно-інтегровані технології  
**КУЗИНСЬКА Катерина Дмитрівна**  
\_\_\_\_\_

Керівник: к.е.н, доцент кафедри ТПС  
**ЧУБ Ольга Ігорівна** \_\_\_\_\_

Рецензент: д.т.н., професор, професор  
кафедри теоретичної та прикладної  
інформатики  
**ФРОЛОВ В'ячеслав Вікторович**  
\_\_\_\_\_

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг роботи складає 70 сторінок, із яких 51 сторінка основної частини з 36 рисунками, 2 таблицями, 45 найменувань списку використаних джерел та додатками.

**Метою кваліфікаційної роботи** є розробка автоматизованої системи аналізу та прогнозування фінансових ринків на основі машинного навчання, яка дозволяє підвищити точність прогнозування цін на акції, ефективно аналізуючи великий обсяг біржової інформації та адаптуючись до нових ринкових трендів.

**Об'єктом дослідження** є процеси аналізу та прогнозування біржової інформації.

**Предметом дослідження** є методи машинного навчання для аналізу та прогнозування цін на акції.

Проблема, яка вирішується в кваліфікаційній роботі, полягає в підвищенні точності прогнозування цін на акції шляхом впровадження автоматизованої системи, що використовує методи машинного навчання та нейронні мережі.

**Область застосування** – розробка та впровадження автоматизованих систем прогнозування для фінансових ринків, що можуть бути використані аналітиками та інвесторами для покращення ефективності прийняття рішень.

**Ключові слова:** МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ПРОГНОЗУВАННЯ, ФІНАНСОВІ РИНКИ, ARIMA, LSTM, АНАЛІЗ ДАНИХ, АВТОМАТИЗОВАНА СИСТЕМА, СТАТИСТИЧНІ МОДЕЛІ, PYTHON.

## ABSTRACT

The explanatory note for the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices. The total volume of the work is 70 pages, of which 51 pages are the main part with 36 figures, 2 tables, 45 items in the list of references, and appendices.

**The purpose of the qualification work** is to develop an automated system for analyzing and forecasting financial markets based on machine learning, which allows improving the accuracy of stock price forecasts by effectively analyzing a large volume of stock exchange information and adapting to new market trends.

**The object of the research** is the processes of analyzing and forecasting stock exchange information.

**The subject of the research** is machine learning methods for analyzing and forecasting stock prices.

The problem addressed in the qualification work is to improve the accuracy of stock price forecasting through the implementation of an automated system that uses machine learning methods and neural networks.

**The application area** is the development and implementation of automated forecasting systems for financial markets, which can be used by analysts and investors to improve decision-making efficiency.

**Keywords:** MACHINE LEARNING, NEURAL NETWORKS, FORECASTING, FINANCIAL MARKETS, ARIMA, LSTM, DATA ANALYSIS, AUTOMATED SYSTEM, STATISTICAL MODELS, PYTHON.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Класифікація методів моделювання поведінки та прогнозування стану фінансових ринків.....	9
1.2 Системологічний огляд автоматизованих інформаційних систем фінансового прогнозування.....	11
1.3 Дослідження інструментарію прогнозування фінансових показників методами машинного навчання.....	14
1.4 Характеристика типів нейронних мереж.....	16
Висновки за розділом 1.....	20
РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	21
2.1 Формалізація вхідних даних.....	21
2.2 Побудова UML-діаграм предметної області.....	25
2.3 Обґрунтування вибору типу нейронної мережі як інструменту прогнозування.....	28
2.4 Розробка архітектури автоматизованої системи.....	31
Висновки за розділом 2.....	35
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ	36
3.1 Вибір середовища розробки автоматизованої системи.....	36
3.2 Методика навчання нейронної мереж.....	39
3.3 Експериментальне дослідження поведінки фінансових ринків в автоматизованій системі.....	48
Висновки за розділом 3.....	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТКИ.....	61

**ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ**

ARIMA (Autoregressive Integrated Moving Average)	модель авторегресійної інтегрованої ковзної середньої;
LSTM (Long Short-term Memory)	тип архітектура рекурентних нейронний мереж;
NCSS	офіційна назва статистичного програмного забезпечення американської корпорації NCSS LLC;
SAS (Statistical Analysis System)	офіційна назва статистичного програмного забезпечення американської корпорації SAS;
GRU (Gated Recurrent Unit)	керований рекурентний нейрон;
SOA (Service-oriented architecture)	сервісно-орієнтована архітектура.

## ВСТУП

Сучасний фінансовий ринок характеризується непередбачуваністю та складністю, що вимагає від учасників систематичного аналізу та прогнозування трендів для прийняття обґрунтованих рішень. Однак, завдяки швидкому розвитку технологій, особливо в галузі машинного навчання, з'являються нові можливості для створення ефективних інструментів аналізу та прогнозування фінансових ринків.

Робота присвячена розробці та впровадженню автоматизованої системи аналізу та прогнозування фінансових ринків, яка базується на методах машинного навчання. Ця система спрямована на вдосконалення процесу прийняття рішень у фінансовому секторі шляхом автоматизації аналізу великого обсягу даних та створення більш точних прогнозів з урахуванням різноманітних факторів, які впливають на ринок.

Застосування нейронних мереж для аналізу та прогнозування фінансових ринків має кілька ключових особливостей, які роблять цей підхід особливо ефективним. Нейронні мережі виявляють високу адаптивність до змін у фінансових ринкових умовах. Вони можуть автоматично адаптуватися до нових трендів, швидко реагуючи на зміни у великому обсязі даних.

Фінансові ринки є динамічними системами зі складними взаємозв'язками між різними факторами, такими як економічні показники, політичні події та психологічні особливості учасників ринку. Нейронні мережі здатні ефективно враховувати ці складні залежності, використовуючи глибоке навчання для виявлення прихованих закономірностей.

Загалом, застосування нейронних мереж для аналізу та прогнозування фінансових ринків відкриває нові можливості для розробки ефективних інструментів управління портфелем та прийняття обґрунтованих рішень у фінансовій сфері.

**Актуальність дослідження.** Актуальність дослідження полягає в тому, що фінансові ринки є складними та непередбачуваними системами, де

прийняття правильних інвестиційних рішень вимагає глибокого аналізу великої кількості даних та надійних прогнозів майбутніх трендів. Застосування методів машинного навчання в цій сфері дозволяє автоматизувати процес аналізу, виявляти складні закономірності та робити прогнози з високою точністю. Зокрема, розвиток нейронних мереж, глибоке навчання та інші методи машинного навчання відкривають нові можливості для створення ефективних інструментів аналізу та управління портфелем, що допомагає інвесторам та фінансовим установам приймати обґрунтовані рішення в умовах постійної зміни ринкових умов і ризиків. Така система може забезпечити конкурентну перевагу учасникам фінансових ринків, забезпечуючи швидке реагування на зміни, максимізацію прибутку та мінімізацію ризиків. Враховуючи динамічний характер фінансових ринків та зростаючу складність прийняття інвестиційних рішень, розробка автоматизованої системи аналізу та прогнозування фінансових ринків на основі машинного навчання має велике значення для практичного застосування в сучасній фінансовій сфері.

Розробкою зазначеної проблеми займається багато науковців, які є представниками різних галузей науки. Під час написання кваліфікаційної роботи автором були проаналізовані наукові результати Р. Брауна, Дж. Бокса, Г. Дженкінса, К. Верчелліса, Ю. Лукашина, А. Івахненка [26, 42] та інших вчених. Але було виявлено, що сьогодні ще існує потреба у дослідженні, яке б узагальнило наявний досвід та осучаснило методіку аналізу та прогнозування саме фінансових ринків.

Саме тому нами й була обрана наступна тема кваліфікаційної роботи: «Автоматизована система аналізу та прогнозування фінансових ринків на основі машинного навчання».

**Об'єкт дослідження** – процеси аналізу та прогнозування біржової інформації методами машинного навчання.

**Предмет** – методи машинного навчання для аналізу та прогнозування цін на акції на товарній біржі.

**Мета роботи** – підвищення прогностичної точності цін на акції на фінансових ринках на основі аналізу біржової інформації методами машинного навчання.

Відповідно до мети дослідження були визначені та виконані наступні наукові завдання:

1. Дослідження інструментарію прогнозування фінансових показників методами машинного навчання.
2. Класифікація нейронних мереж, аналіз принципів їх функціонування.
3. Обґрунтування вибору типу нейронної мережі як інструменту прогнозування.
4. Розробка архітектури автоматизованої системи аналізу та прогнозування фінансових ринків.
5. Опис методики навчання нейронної мережі.
6. Експериментальне дослідження поведінки фінансових ринків за допомогою нейронної мережі.
7. Підготовка практичних рекомендації щодо впровадження та використання системи аналізу та прогнозування фінансових ринків.

Для розв'язання поставлених завдань нами були використані такі **методи дослідження**: теоретико-критичний аналіз наукової літератури, методи математичного моделювання, методи класу Data mining, методи зіставлення, узагальнення та синтезування інформації.

## РОЗДІЛ 1

### ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1 Класифікація методів моделювання поведінки та прогнозування стану фінансових ринків

У сучасному світі економічні процеси швидко змінюються, що призводить до швидкого збільшення або зменшення значень змінних. Щобільше, як стаціонарні, так і нестаціонарні процеси можуть мати сезонну складову, яка посилює динаміку вивченого процесу. Існує дві загальні моделі для побудови прогнозів на основі часових рядів: мультиплікативна та адитивна. У мультиплікативній моделі компоненти виражені коефіцієнтами, а для оцінки явища використовується їх добуток. У випадку адитивної моделі – компоненти сумуються.

В статистиці нульова гіпотеза (часто позначається як  $H_0$ ) стверджує загальне припущення про відсутність зв'язку між двома вимірюваними явищами, тоді як альтернативна гіпотеза (позначається як  $H_1$ ) заперечує це. Існують тести для перевірки нульової гіпотези нестаціонарності часового ряду, такі як тест Дікі-Фулера, тест Квятковського-Філіпса-Шмідта-Шіна і Тест Філіпса-Перрона [24].

Фільтрація даних використовується для усунення зашумленості, зайвої інформації та великих коливань у прогнозованих явищах, коли стійкі закономірності відсутні або малопомітні. Рекурсивний метод Кальмана є ефективним і широко застосовується для розрахунку оптимальних оцінок залишків та побудови короткострокових прогнозів на основі обраних моделей. Процес фільтрації складається з двох кроків: спочатку відбувається екстраполяція, де прогнозується поточне значення на основі попереднього, а потім значення корегуються за допомогою середнього зваженого.

Для опису часових рядів використовуються різноманітні типи моделей, такі як інтелектуальний аналіз даних та алгоритми нечіткого виведення на

основі бази правил. Однак класичні регресивні моделі не завжди відображають складну поведінку реальних динамічних процесів. У таких випадках використовують ймовірнісні методи, такі як мережі Байеса. Для прогнозування може застосовуватися метод групового урахування аргументів, який покликаний підвищувати точність моделі через рекурсивний селективний відбір моделей [10].

Існують два основних підходи до прогнозування вартості акцій: фундаментальний аналіз і технічний аналіз. Крім того, існують різноманітні евристичні методи, за допомогою яких учасники ринку намагаються виявити патерни у динаміці ціни. Однак деякі з цих методів, такі як хвильовий підхід, не мають серйозного наукового або статистичного підтвердження.

Фундаментальний аналіз цінних паперів – це метод отримання інформації про справжню цінність акцій, незалежно від їхньої поточної ринкової ціни. За неоптимальності цін, трейдер може заробити, відкривши довгу або коротку позицію відповідно. Технічний аналіз, з іншого боку, базується на аналізі поточного балансу попиту та пропозиції акцій, а також на історичних даних для прогнозування трендів [18].

Біржова склянка – це засіб, який дозволяє отримати інформацію про те, хто саме зараз на ринку готовий купувати й продавати, а також за якою ціною (рис. 1.1).

ORDER	AMOUNT	TOTAL	PRICE	PRICE	TOTAL	AMOUNT	COUNT
1	0.1775	0.1775	8,670.8	8,672.1	0.0240	0.0240	1
3	0.6735	0.8510	8,670.7	8,672.4	0.2240	0.2000	1
2	1.520	2.371	8,670.2	8,672.6	0.3390	0.1150	1
1	0.5000	2.871	8,669.7	8,672.7	0.7671	0.4281	1
2	0.4556	3.327	8,669.6	8,673.0	0.8042	0.0371	1
1	3.000	6.327	8,669.0	8,674.1	1.320	0.5156	1
1	8.211	14.54	8,667.6	8,674.2	1.990	0.6700	2
1	3.000	17.54	8,667.4	8,674.4	12.66	10.67	2
3	2.314	19.85	8,667.3	8,675.9	12.67	0.0050	1
1	0.0050	19.86	8,666.7	8,676.0	14.77	2.100	1
1	1.392	21.25	8,665.5	8,676.5	14.98	0.2231	2
1	0.1047	21.35	8,665.7	8,676.6	15.13	0.1444	1
1	0.1524	21.51	8,665.0	8,676.7	17.23	2.100	1
1	0.0096	21.51	8,664.9	8,677.9	17.52	0.2881	1
1	0.0623	21.57	8,664.8	8,678.9	17.55	0.0231	1
5	3.364	24.93	8,664.7	8,679.0	17.74	0.1997	1
1	1.000	25.93	8,664.5	8,679.4	18.54	0.8000	1
1	0.1475	26.08	8,663.9	8,679.5	18.96	0.4200	1
1	0.0231	26.10	8,663.8	8,679.6	19.25	0.2880	1
1	0.5000	26.60	8,663.4	8,680.0	19.28	0.0300	1
2	1.496	28.09	8,663.0	8,680.3	19.38	0.072	1
2	0.6525	28.75	8,662.7	8,680.5	20.22	0.8400	2
1	0.3986	29.03	8,662.4	8,680.8	20.24	0.0200	1
1	0.1516	29.19	8,662.1	8,681.0	21.90	1.664	1

Рисунок 1.1 – Приклад біржової склянки

Цей інструмент корисний для виявлення коливань у попиті та пропозиції, що може мати важливе значення в короткостроковій перспективі, оскільки зростання попиту може призвести до підвищення ціни цінних паперів.

Технічний аналіз використовує ліву частину графіка для прогнозування значень у правій частині. Деякі дослідники вважають цей підхід необґрунтованим через непередбачуваність ринку. Проте прихильники технічного аналізу стверджують, що він базується не на передбаченні графіка, а на прогнозуванні поведінки учасників ринку, які у подібних ситуаціях діють аналогічно.

Вчені продовжують проводити дослідження, щоб обґрунтувати, який метод прогнозування дає результати з меншою похибкою та вищою точністю: модель авторегресійної інтегрованої ковзної середньої (ARIMA) чи нейронна мережа з архітектурою Long Short-term Memory (LSTM). Вони використали історичні дані цін акцій кількох компаній та розділили їх на навчальний та тестовий набори даних у співвідношенні 70% до 30%. Результати показали, що модель на основі LSTM прогнозувала краще на 85% у середньому порівняно з ARIMA [24].

## **1.2 Системологічний огляд автоматизованих інформаційних систем фінансового прогнозування**

Існує значна кількість готових програмних рішень для обробки, аналізу та прогнозування часових рядів. Нижче оглянуто деякі програми, які доступні для використання на платформі Windows.

Однією з цих програм є система GMDH Shell – це професійне програмне забезпечення, що базується на нейронних мережах, призначене для розв’язання проблем прогнозування часових рядів [8, 19]. У цій системі доступні графічне відображення даних, побудова імітаційних моделей, аналітика, візуалізація даних та прогнозування (рис 1.2).

Програма підтримує формати файлів .csv та .xlsx. Після вибору файлу з даними вона пропонує обрати шаблон: кластеризація, класифікація, згладжування, регресійна модель, а також різні види прогнозів. Після цього програма розпочинає обчислення.



Рисунок 1.2 – Вікно системи GMDH Shell із результатами прогнозування

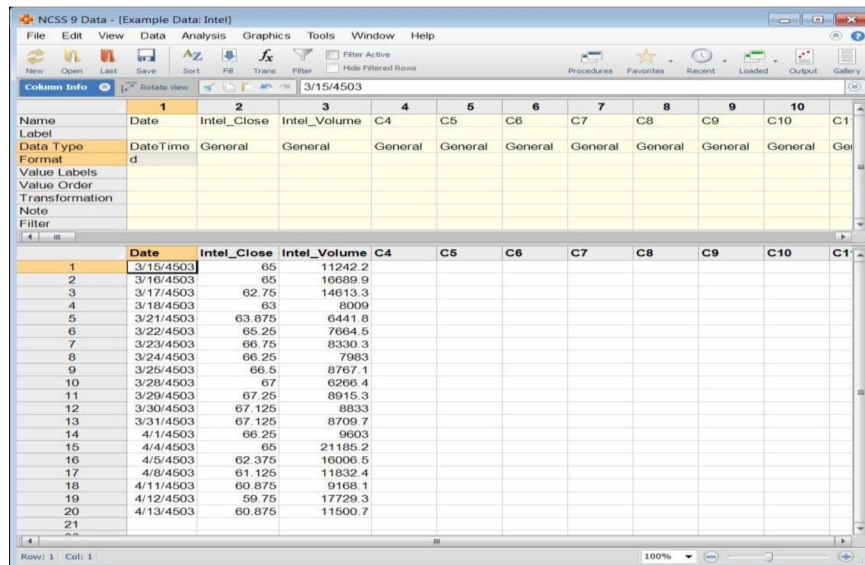
Поza самими значеннями, програма також розраховує інтервали довіри, коефіцієнти кореляції, відхилення та будує діаграми залишків (рис 1.3):



Рисунок 1.3 – Приклад побудови прогнозу в системі GMDH Shell

Система NCSS надає повну та просту у використанні колекцію сотень статистичних та графічних інструментів для аналізу та візуалізації даних [29]. Програмне забезпечення NCSS включає в себе набір інструментів для аналізу

часових рядів та прогнозування, такі як ARIMA, спектральний аналіз, декомпозиція, експоненційне згладжування та кореляції. Кожен інструмент для аналізу часових рядів та процедура прогнозування є простими у використанні та перевірені на точність. Підтримуються основні формати зберігання даних: .txt, .csv, .xlsx, а також власний формат .NCSS (рис. 1.4):



The screenshot shows the NCSS 9 Data software interface. The main window displays a data table with the following columns: Date, Intel\_Close, Intel\_Volume, C4, C5, C6, C7, C8, C9, C10, and C11. The data is organized into rows, with the first row being the header. The data rows show dates from 3/15/4503 to 4/13/4503, along with corresponding Intel\_Close and Intel\_Volume values. The C4 through C10 columns are currently empty.

	1	2	3	4	5	6	7	8	9	10	11
Name	Date	Intel_Close	Intel_Volume	C4	C5	C6	C7	C8	C9	C10	C11
Label											
Data Type	Date Time	General	General	General	General	General	General	General	General	General	General
Format	d										
Value Labels											
Value Order											
Transformation											
Note											
Filter											
1	3/15/4503	65	11242.2								
2	3/16/4503	65	16689.9								
3	3/17/4503	62.75	14613.3								
4	3/18/4503	63	8009								
5	3/21/4503	63.875	6441.8								
6	3/22/4503	65.25	7664.5								
7	3/23/4503	66.75	8330.3								
8	3/24/4503	66.25	7983								
9	3/25/4503	66.5	8767.1								
10	3/28/4503	67	6266.4								
11	3/29/4503	67.25	8915.3								
12	3/30/4503	67.125	8833								
13	3/31/4503	67.125	8709.7								
14	4/1/4503	66.25	9603								
15	4/4/4503	65	21185.2								
16	4/5/4503	62.375	16006.5								
17	4/8/4503	61.125	11832.4								
18	4/11/4503	60.875	9168.1								
19	4/12/4503	59.75	17729.3								
20	4/13/4503	60.875	11500.7								
21											

Рисунок 1.4 – Користувацький інтерфейс системи NCSS

Система Statistical Analysis System (SAS) включає набір програмних пакетів, серед яких варто звернути увагу на SAS Enterprise Miner. SAS Enterprise Miner володіє потужним інструментарієм для обробки та пошуку шаблонів у текстових даних, набором сучасних статистичних методів та зручним інтерфейсом для візуалізації даних.

Дані можуть бути завантажені з файлів різних типів або введені вручну. Під час обробки даних вони перевіряються на помилки, виправляються, перетворюються в зручний формат, перевіряються на правильність і зберігаються для подальшої обробки. У процесі аналізу будуються прогнозні моделі, застосовуються статистичні методи. Під візуалізацією розуміється створення графіків та діаграм для відображення результатів аналізу [39].

SAS Enterprise Miner підтримує формати .txt, .csv, .dat, а також інтеграцію з базами даних Oracle та Teradata. За допомогою графічного

інтерфейсу (рис 1.5) користувачі можуть пройти всі етапи процесу аналізу даних: вибір джерел даних, їх дослідження та модифікація, моделювання та оцінка якості моделей, а також подальше використання цих моделей [25].



Рисунок 1.5 – Користувацький інтерфейс програми SAS

### 1.3 Дослідження інструментарію прогнозування фінансових показників методами машинного навчання

Для сучасних трейдерів, інвесторів, інвестиційних фондів та всіх, хто прагне зберегти або збільшити свої фінансові активи, рекомендується використовувати методи дескриптивного аналізу, інтелектуального аналізу даних, машинного навчання та обробки статистичної інформації, зокрема роботу з великими обсягами даних, для забезпечення конкурентних переваг. Більшість досліджень в галузі методів машинного навчання можна розділити на дві групи: індуктивне та дедуктивне навчання. Індуктивне навчання, яке базується на виявленні емпіричних закономірностей у даних, містить методи екстраполяції, класичний регресійний аналіз та використання нейронних мереж. Дедуктивне навчання натомість передбачає формалізацію експертних знань та їхнє перенесення на комп'ютер для створення бази знань з досліджуваної проблематики. Обидва напрямки машинного навчання мають свої властивості, які залежать від вибірки даних та характеристик програмного забезпечення.

Для розробки прогнозів можна використовувати нейронну мережу, яка втілює методологію алгоритмів LSTM з довготривалою та короткотривалою пам'яттю, а також алгоритми машинного навчання, які виконані за допомогою хмарних технологій в середовищі Google Colab та мови програмування Python з використанням бібліотек, таких як numpy, pandas, matplotlib, datetime, keras, sklearn, tensorflow. Numpy забезпечує необхідний математичний апарат для роботи з матрицями та векторами, pandas – пакет для роботи з даними у формі таблиць, matplotlib – для візуалізації даних, datetime – для роботи з датою та часом, а keras, sklearn та tensorflow – бібліотеки для розробки та застосування моделей машинного навчання [32].

Composer – це комплексна платформа, яка дозволяє користувачам створювати торгові алгоритми, використовуючи штучний інтелект без необхідності програмування, тестувати їх та автоматично виконувати угоди. Платформа також пропонує готові до використання стратегії, нульові комісії та зручний інтерфейс з умовними твердженнями (if...else) щодо торгових стратегій та параметрами фільтрації для точного контролю над торговими рішеннями [41].

EquBot від QuantumStreet AI – це інноваційна платформа, що базується на штучному інтелекті. Вона використовує передові алгоритми машинного навчання, графи знань і процесор природної мови IBM Watson для надання послуг портфолію як платформи (PaaS) для фахівців у галузі інвестицій по всьому світу. Завдяки високопродуктивним засобам обробки даних EquBot швидко аналізує масові обсяги новин, пости у соціальних мережах і фінансові звіти у режимі реального часу, щоб забезпечити точні прогнози. Інтеграція графів знань і обробки природної мови допомагає збирати цінні дані з різноманітних джерел, тоді як повноцінний аналіз у реальному часі дозволяє інвесторам приймати обґрунтовані рішення [25].

Компанія Kavout створила інноваційну інвестиційну програму, основою якої є функція, відома як K-score. Цей показник відзначається своєю унікальністю, оскільки він формує свою оцінку, об'єднуючи понад 200

показників і факторів, що використовуються в оцінці капіталу. Застосовуючи технологію розпізнавання патернів штучного інтелекту для щоденного виявлення тенденцій, пропозиції з акцій формуються за допомогою глибокого аналізу, який визначає тенденції або виявляє будь-які аномалії, що робить Kavout привабливим варіантом для інвесторів, які бажають приймати обґрунтовані рішення з мінімальними витратами [31].

Усі досліджені рішення мають значні відмінності та переваги, але у них всіх є спільний недолік: необхідність звертатися до служби підтримки та замовлення аналізу акцій, який виконується штучним інтелектом. Головним недоліком цих рішень є їх висока ціна. Проте, інформаційна система, яка пропонується, буде повністю відкритою для користувачів та матиме простий інтерфейс. Він не буде відривати користувача від основної функціональності – прогнозування цін акцій за допомогою різних методів машинного навчання та проведення відповідного аналізу.

#### **1.4 Характеристика типів нейронних мереж**

Нейронна мережа – це сукупність нейронів та їх взаємозв'язків. Основним складовим елементом нейронної мережі є формальний нейрон, який виконує операцію нелінійного перетворення суми добутків вхідних сигналів на вагові коефіцієнти [24].

Використання штучних нейронних мереж є ефективним при дослідженні різноманітних систем [40]. Загальний випадок описує штучну нейронну мережу як адаптивну нелінійну динамічну систему. Завдяки цьому можна розв'язувати складні обчислювальні або математичні задачі, такі як апроксимація функцій, класифікація, кластеризація, прогнозування та інші, використовуючи рівноважні стани системи.

Можна виділити основні аспекти, за якими класифікуються нейронні мережі. Серед них – типи нейронів, структура моделі мережі, методи навчання та конкретні завдання, які вони вирішують. До завдань відносяться обробка

даних, класифікація, кластерний аналіз, прогнозування, апроксимація, розпізнавання текстової тональності, стиснення даних за допомогою нейромережі, асоціативну пам'ять тощо.

Також можна класифікувати нейронні мережі залежно від методів навчання. Одна з таких класифікацій – мережі, що навчаються з учителем і мережі, що навчаються без учителя. Перші отримують дані про внутрішні взаємодії мережі, використовуючи визначений набір параметрів як вибіркового вихід. Мережа намагається знайти залежність між початковою інформацією та результатами. На відміну від цього, нейронні мережі, що навчаються без учителя, використовують навчання за допомогою конкуренції, де вузли змагаються між собою за найкраще відповідність вхідному набору. Переможець коригує структуру мережі [11]. Такий тип мереж відомий як адаптивні нейронні мережі. Вони відрізняються чисельним методом обробки даних, що дозволяє їм розподіляти обчислювальні завдання на окремі потоки з високим рівнем паралелізму.

Рекурентні нейронні мережі являють собою форму штучних нейронних мереж, що використовуються переважно для класифікації та розпізнавання. Вони спроможні розпізнавати послідовність даних та прогнозувати майбутні сценарії, користуючись шаблонами. Зазвичай вони застосовуються в глибокому навчанні та при моделюванні активності людського мозку. Рекурентні мережі відрізняються від інших штучних нейронних мереж тим, що використовують зворотний зв'язок для обробки послідовних даних, тобто вони спираються на минулу інформацію для прийняття рішень у поточний момент. Це можна порівняти з пам'яттю, яка допомагає мережі зберігати попередні дані.

Рекурентні нейронні мережі широко застосовуються в різних областях, таких як побудова мовних моделей, де вони допомагають передбачати наступні літери у слові або наступні слова у реченні, базуючись на попередніх даних.

У рекурентних нейронних мережах інформація може пересуватися як вперед, так і назад, завдяки зворотному зв'язку. Це дозволяє їм обробляти послідовні та часові дані. Наприклад, у нейронній мережі зі скороченим зворотнім поширенням часу кількість часових кроків у вхідній послідовності обмежується, що корисно для моделей, які працюють із послідовностями, коли вхідна послідовність довша, ніж вихідна.

Пам'ять у рекурентних нейронних мережах, яка була розглянута раніше, старається зберегти тривалі зв'язки. Це пов'язано з проблемою зникаючого градієнта, яка може виникати, коли ранні частини вхідної послідовності мають велике значення для контексту. Ця проблема добре відома у контексті зворотного поширення та градієнтного спуску, які важко застосувати до рекурентних нейронних мереж, а також до мереж прямого поширення через їхню глибоку структуру. У перших випадках проблеми виникають через глибкість мережі, а у других – через ширину рекурентних мереж [22].

Рекурентні нейронні мережі використовуються для обробки даних в часовому порядку, аналізуючи їхні зміни та прогнозуючи майбутні. Це робиться завдяки їхній структурі, що дозволяє моделювати тимчасові ряди, що мають загальну статистичну обробку.

Необхідність у модифікаціях нейронних мереж була обумовлена математичною складністю сучасних задач. Компоненти нейронних мереж стали використовуватися для створення різноманітних модифікацій систем моделювання, або, іншими словами, модифікацій нейронних мереж. З правильним використанням новітніх розробок, які спрощують комунікацію та обмін сигналами в мережі, а також полегшують або ефективно формують шари навчання, відтворення, рекурентні та згорткові шари, а також застосування модифікацій, що підсилюють ці функції нейронних мереж, можна отримати ефективний та швидкий інструмент для розв'язання потрібних завдань [14].

Модифікація нейронних мереж, яка усуває проблеми, що характерні для звичайних нейронних мереж, полягає в наданні можливості нейронам

запам'ятовувати останню отриману інформацію, не втрачаючи при цьому можливості зберігання інформації, що була опрацьована багато ітерацій тому, незалежно від якості чи важливості цієї інформації. Ця модифікація, відома як LSTM, використовує складну систему нейронів, побудовану досить складно.

Основна концепція цих мереж полягає в передачі пам'яті між шарами, досягаючи цього завдяки включенню спеціального показника, відомого як «стан клітини», до складу кожного нейрона. Цей показник пройде через всю мережу, зберігаючи інформацію та передаючи її на наступні ітерації.

Особливими елементами нейрона є фільтри, які визначають, як опрацьовувати інформацію, що проходить через них. Кожен нейрон має власний «стан», який відіграє роль пам'яті в мережі, а також основні типи фільтрів: вхідний, вихідний і забуваючий. Метою цих фільтрів є захист інформації та визначення, яка частина інформації буде передана далі і в якому обсязі [23].

Існує ще одна модифікація рекурентних нейронних мереж, спрямована на розв'язання завдань прогнозування та передбачення, відома як керований рекурентний нейрон (GRU). GRU можна розглядати як варіацію LSTM, оскільки обидва методи використовують подібні модифікації та нейрони і, в деяких випадках, дають подібні результати. GRU використовує ворота оновлення та скидання для розв'язання проблеми зникаючого градієнта, дозволяючи зберігати інформацію на тривалий час і не видаляти її з пам'яті.

Прогнозування цін на акції є надзвичайно складною задачею, особливо з урахуванням високої волатильності та впливу різноманітних факторів (політичних, економічних, соціальних тощо). Використання глибокого навчання, зокрема рекурентних нейронних мереж, для прогнозування ринку акцій стає все більш поширеним явищем в літературі. LSTM і GRU є одними з найпоширеніших типів рекурентних нейронних мереж, оскільки вони добре пристосовані для роботи з послідовними даними.

## **Висновки за розділом 1**

Перший розділ кваліфікаційної роботи присвячений огляду та класифікації статистичних моделей, які використовуються для прогнозування фінансових ринків, а також опису характеристик нейронних мереж, які застосовуються в аналізі фінансових показників.

Системологічний огляд автоматизованих інформаційних систем фінансового прогнозування включав вивчення різних підходів до розробки та впровадження таких систем, їх структуру та можливості. Висвітлення цього аспекту дозволило зрозуміти, як сучасні технології допомагають у підвищенні точності та швидкості фінансового прогнозування.

В цілому, проведена науково-дослідна робота дозволила систематизувати різноманітні методи та підходи до прогнозування фінансових ринків, розкрити їх переваги та недоліки, а також виокремити напрямки подальших досліджень у цій області.

## РОЗДІЛ 2.

### МОДЕЛЮВАННЯ СТРУКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ

#### 2.1 Формалізація вхідних даних

Створення сучасного програмного рішення для аналізу фондових ринків є важливим завданням у сучасному контексті. Готове програмне забезпечення спростить процес збору хронологічних даних, необхідних для відстеження динаміки акцій, та надасть користувачеві зручний графічний інтерфейс для візуалізації цих змін.

Програмне забезпечення, яке розробляється, призначене для трейдерів та аналітиків, осіб, які активно працюють або цікавляться участю на фондовому ринку. Головна задача програмного забезпечення – автоматизувати прогнозування цін на акції.

Серед функціональних вимог до інформаційної системи для прогнозування фінансових ринків на базі алгоритмів машинного навчання виділяються [19]:

- здатність системи виявляти складні закономірності для прогнозування акцій на ринку;
- наявність актуальних даних, які не були порушені нетиповими подіями у світі, які не можна було передбачити.
- адаптованість до сезонних змін на ринку.

До нефункціональних вимог входять:

- забезпечення безпеки та конфіденційності;
- надійність.
- відновлення після збоїв.
- висока продуктивність.
- збереження великих обсягів даних.
- ефективне керування помилками.
- використання правильних правил перевірки.

- відповідність стандартам.

Вхідними даними для інформаційної системи є інформація, що вводить користувачем. Це включає тикер акції, щоб отримати прогнозування, і метод машинного навчання, який використовується для прогнозування ціни на акції.

Особливу увагу приділено розробці окремого модуля для прогнозування курсів валют, який надає точні показники у реальному часі. Використання статистично-імовірнісного прогнозування дозволяє отримати більш точні результати без значного збільшення обчислювального навантаження [5].

Наша подальша робота зосереджена на розв'язанні завдання за допомогою аналізу фундаментальних показників. Ми скористалися спеціальним датасетом, що містить велику кількість значень фінансових звітів компаній за останні 25 років. Кожний запис у цьому датасеті включає значення, які потрібно прогнозувати.

Значення, які необхідно прогнозувати:

- $E_{t+1}$  Earnings (ib – spi) in year  $t + 1$  – прибуток за рік;
- $EPS_{t+1}$  Earnings (ib – spi) in year  $t + 1$  scaled by shares outstanding (csho) – прибуток за рік, поділений на кількість акцій.

Вхідні параметри для регресійних моделей:

- $E_t$  Earnings (ib – spi) in year  $t$ ;
- $A_t$  Total assets (at);
- $D_t$  Dividend payment (dvc);
- $DD_t$  Dummy variable indicating dividend payers;
- $NegE_t$  Dummy variable indicating negative earnings;
- $AC_t$  підсумовані негрошові активи, включаючи короткострокові борги й податки;
- $EPS_t$  – Earnings per share when earnings are positive, and zero otherwise;
- $AC_t$  – Accruals per share when accruals are negative, and zero otherwise;
- $AG_t$  – Percentage change in total assets;
- $NDD_t$  – Dummy variable indicating zero dividend per share;

- $DIV_t$  – Dividend per share (dvpsx\_f);
- $BTM_t$  – Закредитованність до ринку;
- $Price_t$  Stock price as of three months after;
- $BVE_t$  Book value of equity (ceq);
- $TACC_t$  Total accruals;

Вхідні параметри моделей машинного навчання:

- Sale – sales (продаж);
- COGS – cost of goods sold (собівартість проданих товарів);
- XSGA – selling, general and admin. exp. (загальні витрати);
- XAD – Advertising expense (витрати на маркетинг);
- XRD – Research and development (витрати на дослідження);
- $DP_t$  – Depreciation and amortization (dp);
- $XINT_t$  – Interest and related expense (xint);
- $NOPIO_t$  non-operating income (expense) – other (nopio);
- TXT – Income taxes (податок на прибуток);
- $XIDO_t$  Extraordinary items and discontinued operations (xido);
- E – Earnings (прибуток);
- Common dividend (dvc);
- $CHE_t$  – короткострокова інвестиція (che);
- $INVT_t$  – сумарна інвентаризація (invt);
- $RECT_t$  – сумарний прибуток (rect);
- ACS – (вартість усього майна);
- $PPENT_t$  – вартість нерухомості й фізичних об'єктів;
- $IVAO_t$  – інвестиції;
- $INTAN_t$  – нематеріальні активи (intan);
- $AT_t$  – загальні активи (at);
- $AP_t$  – балансовий рахунок (ap);
- $DLC_t$  – борги в поточних зобов'язаннях (dlc);
- $TXP_t$  – податки на прибуток (txp);
- $LCT_t$  – усі пасиви (lct);

- $DLTT_t$  – довгострокові борги (dltt);
- $LT_t$  – усі борги (lt);
- $SEQ_t$  – звичайний капітал (seq) усього 28 параметрів для 1000 компаній за 25 років.

Датасет був розширений інформацією про ціни акцій на закінчення торгового дня з частотою один раз на тиждень. Це означає, що кожен запис включав додаткові дані про ціну на момент звіту та ціну через три місяці. На основі цих даних проводилось навчання всіх моделей [24].

Ключовими метриками успішності моделі будуть показники точності класифікації за мультикласовою шкалою. Ми обмежили наше завдання до прогнозування, чи зросте вартість активу протягом наступних трьох місяців, впаде чи залишиться незмінною.

Довіра до результатів розрахунків залежить від меншої точності прогнозування. Користувачу легше повірити, що ціни акцій зростатимуть через три місяці, ніж точно визначити їх конкретні значення в цей момент. Важливою характеристикою моделі є її здатність обґрунтувати свої прогнози, що збільшить ймовірність і покращить споживчі якості всієї системи [31].

Моделювання систем рекурентних нейронних мереж базувалося на використанні даних, які найкраще підходили для навчання таких мереж і відповідали основній меті – прогнозуванню часових рядів. У ролі таких часових рядів були взяті датасети цін акцій двох міжнародних компаній, які містили інформацію про ціни акцій на момент закриття торгівлі в кінці кожного дня.

Отже, було розглянуто застосування рекурентних нейронних мереж для прогнозування цін акцій на фондовому ринку. Це важлива задача, яка потребує аналізу часових рядів, таких як дані про ціни акцій компаній. Для побудови моделей навчання були використані відповідні набори даних, і головною метою було забезпечення точних прогнозів для вирішення завдань у фінансовій сфері.

## 2.2 Побудова UML-діаграм предметної області

У програмному забезпеченні присутня діаграма класів, яка відноситься до статичних структурних діаграм (рис 2.1). Вона служить для відображення класів системи разом з їх атрибутами, методами та зв'язками між об'єктами. Ці діаграми корисні: вони відображають структуру класів у системі; вони є основою для інших структурних діаграм, що використовуються в UML; вони корисні для розробників та інших учасників команди. Також, бізнес-аналітики можуть використовувати їх для моделювання систем з погляду бізнесу [18].

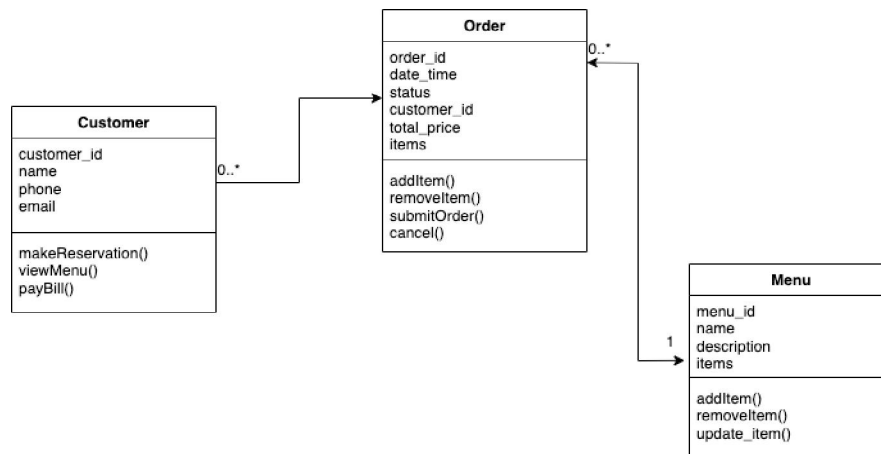


Рисунок 2.1 – Діаграма класів системи

В програмі запропоновано 4 класи.

Основний клас, який має назву Application, є точкою входу в систему. Він має два методи: `init`, що створює інтерфейс програми, та `predict`, який виконує логіку завантаження моделей. Далі йдуть три класи, які реалізують алгоритми машинного навчання: `ARIMAPredictor`, `LSTMPredictor` та `RandomForestPredictor`. Кожен з них завантажує, тренує, тестує та робить передбачення за допомогою відповідних моделей [21].

Діаграма послідовності – це форма діаграми взаємодії, яка демонструє, як група об'єктів працює разом та в якому порядку. На рис 2.2 наведена діаграма послідовності системи, яка моделюється.

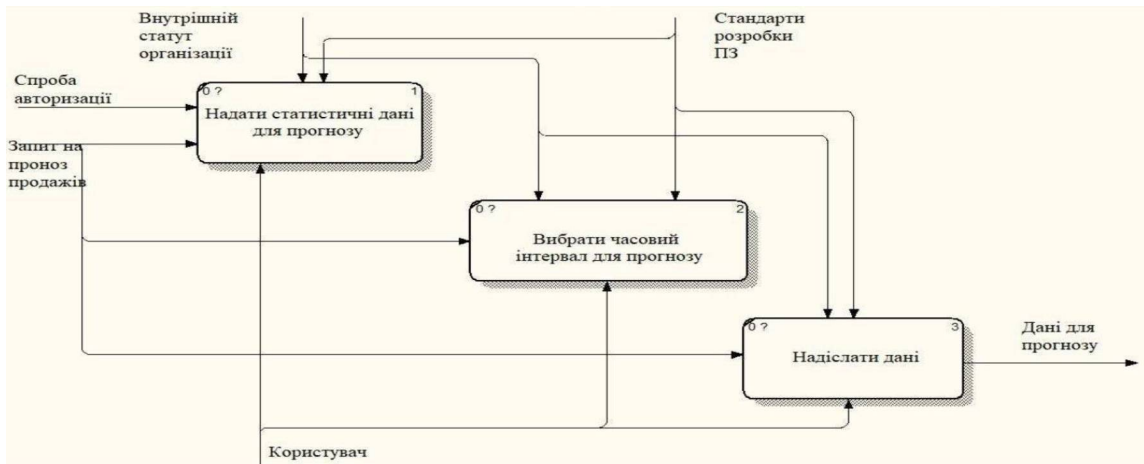


Рисунок 2.2 – Діаграма послідовності системи

У Таблиці 2.1 наведено основні функції класів, які має система, що розробляється.

Діаграма діяльності – це різновид блок-схеми, яка використовується уніфікованою мовою моделювання для відображення послідовності дій у системі або процесі. Вона дозволяє описати різні динамічні аспекти системи та візуалізувати складні процеси, часто застосовуючись для моделювання бізнес-процесів або етапів роботи з діаграмами варіантів використання в організаціях [10]. На рис. 2.3 наведена діаграма діяльності системи.

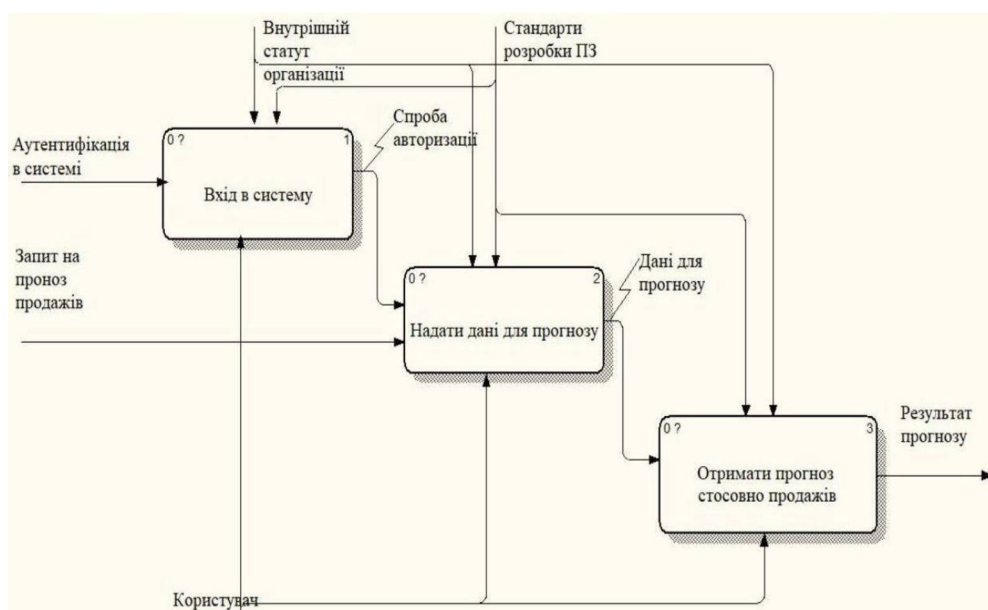


Рисунок 2.3 – Діаграма діяльності системи

### Специфікація функції розробленої системи

Назва	Опис
Application	Клас, що повністю відповідає UI частині продукту
RandomForestPredictor	Клас, який відповідає за завантаження, підготовку та тренування даних, які потім будуть використовуватися в базовому методі машинного навчання Random Forest Regression
LSTMPredictor	Клас, який відповідає за завантаження, підготовку та тренування даних, які потім будуть використовуватися в рекурентній нейронній мережі LSTM
ARIMAPredictor	Клас, який відповідає за завантаження, підготовку та тренування даних, які потім будуть використовуватися в одному з методів прогнозування часових рядів ARIMA

Представлена на рис. 2.3 діаграма діяльності описує взаємодію користувача з системою, де користувач, після запуску системи, обирає, що спочатку ввести – тікер чи алгоритм машинного навчання. Після цього система завантажує дані, створює модель, тренує, тестує та створює графік передбачуваної та реальної ціни акцій. Після виконання цих дій користувач отримує зручний інструмент для аналізу фондового ринку.

Діаграма станів – це інструмент, що містить у себе стани, переходи, події та дії. Вона використовується для візуалізації динаміки системи, особливо важлива для моделювання поведінки інтерфейсів, класів або взаємодії [16]. Ця діаграма відображає упорядковану поведінку об'єкта відповідно до подій.

Ключові стани, які відображені на діаграмі (рис 2.4):

- початковий стан, коли систему тільки що запущено;
- очікування користувача (система очікує введення користувачем тікера та вибору алгоритму машинного навчання);
- після вибору тікера та алгоритму завантажуються ціни;
- якщо ціни успішно завантажено, то створюється модель, вибрана раніше користувачем;
- тренування моделі;
- тестування моделі.

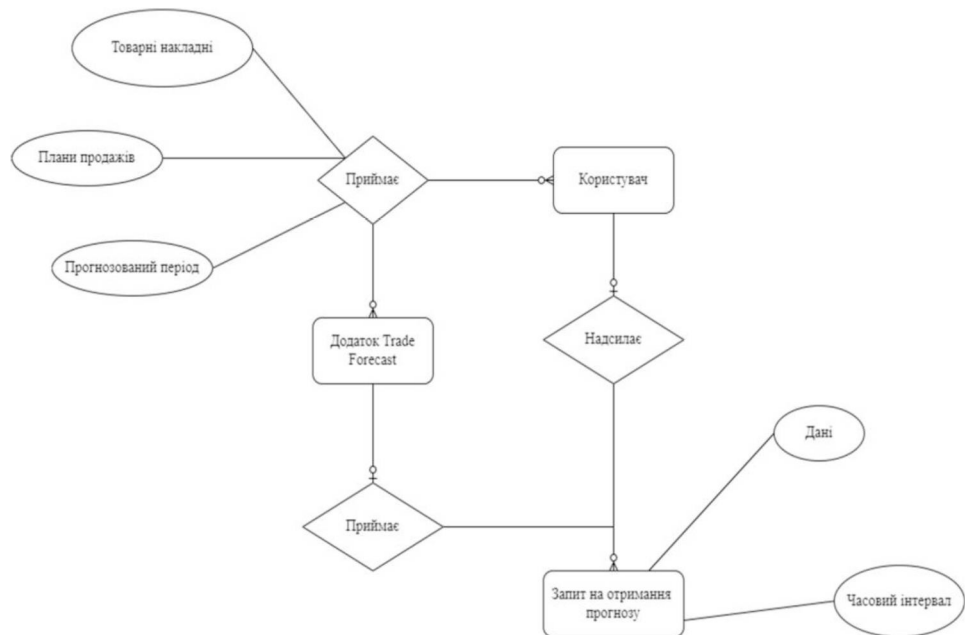


Рисунок 2.4 – Діаграма станів системи

### 2.3 Обґрунтування вибору типу нейронної мережі як інструменту прогнозування

Основною метою концепції архітектури рекурентних нейронних мереж є забезпечення того, щоб кожен етап ділив одну й ту ж прогресуючу внутрішню структуру, і мережа могла отримувати інформацію не лише про поточний момент часу, але й про попередній [20]. Для досягнення цієї мети

необхідно, наприклад, встановлювати спільні зв'язки між прихованими шарами, передавати сигнали з попереднього кроку часу до прихованих шарів поточного кроку часу [34].

Вибуховий градієнт відзначається тим, що кожна вага стає занадто великою, що унеможлиблює навчання мережі. Градієнти такого типу насичуються наприкінці ітерацій, і вони надто потужні. Проблему вибухових градієнтів можна вирішити, поділяючи їх на усічені та розсічені. Однак, нейронна мережа, описана в цьому контексті, має важливий недолік: вона не може отримувати оцінку подій, що сталися давно [15].

Нейронні мережі з довгою короткостроковою пам'яттю виникають як відповідь на проблеми, які виникають у звичайних рекурентних нейронних мережах, про які йшлося у попередньому розділі. Ці модифіковані мережі призначені для розв'язання проблем, пов'язаних з тривалими залежностями в даних. У них є здатність зберігати та обробляти інформацію протягом тривалого періоду. Зазвичай рекурентні нейронні мережі мають ланцюжкову структуру, що складається з модулів мережі, що повторюються. Звичайна рекурентна нейронна мережа складається з таких модулів, які мають досить просту будову, наприклад, модуль може складатися лише з одного шару з функцією активації гіперболічного тангенсу, як це було описано раніше. Нижче наведено приклад такої мережі (рис. 2.5):

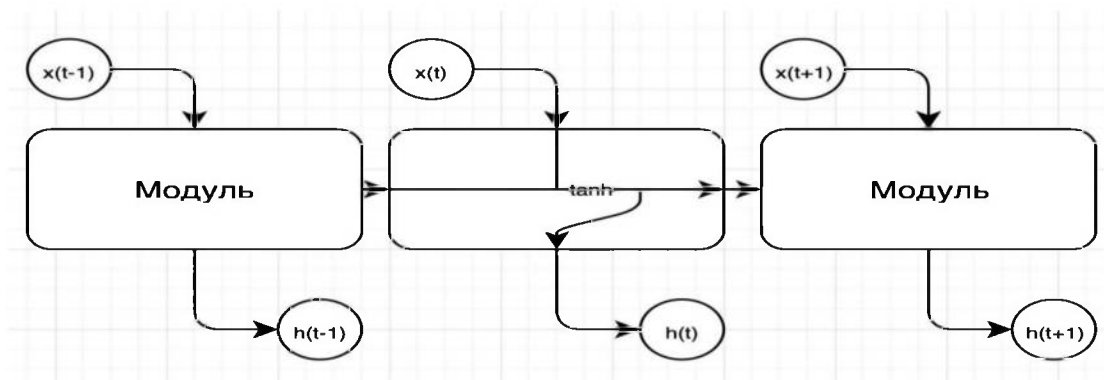


Рисунок 2.5 – Рекурентна нейронна мережа з функцією активації  $\tanh(x)$

При детальному аналізі модуля звичайної рекурентної нейронної мережі очевидно, що він не має складної структури – модуль складається лише з одногошарового перцептрона з вказаною функцією активації. У нейронних мережах з довгою короткостроковою пам'яттю використовується значно складніший модуль, який містить кілька функцій активації та багато різноманітних математичних перетворень, як показано на рис 2.6.

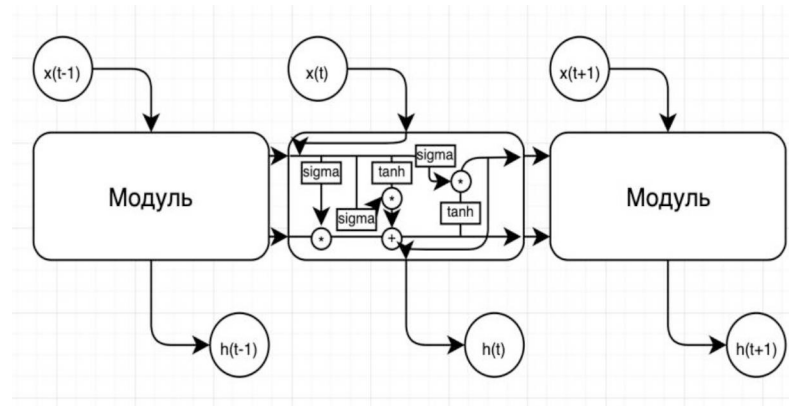


Рисунок 2.6 – Модуль мережі LSTM

Керований рекурентний нейрон також отримав популярність у різних завданнях, таких як прогнозування та передбачення. Цей вид рекурентних нейронних мереж відрізняється від мереж з довгою короткостроковою пам'яттю, і тому заслуговує окремого розгляду як особливий тип рекурентних нейронних мереж [18].

GRU можна розглядати як спрощену версію мереж з довгою короткостроковою пам'яттю. Хоча він містить багато схожих концепцій, він має менше параметрів і, отже, швидше навчається при однаковому розмірі прихованого шару. Швидкість навчання є важливим фактором для будь-якої нейронної мережі, оскільки вона відображає швидкість роботи системи та обчислювальні потужності, необхідні для розв'язання завдань прогнозування та передбачення. Недавні дослідження також показали, що керований рекурентний нейрон та мережі з довгою короткостроковою пам'яттю мають

приблизно однакову точність, а в деяких випадках GRU є навіть точнішим. Таким чином, немає чіткого правила, яке вказувало б, який саме тип обирати. Це схоже на вибір найкращої функції активації для звичайних нейронних мереж: потрібно експериментувати з різними варіантами та визначити, який з них працює краще для конкретних даних.

## 2.4 Розробка архітектури автоматизованої системи

Для системи, що розробляється, підходить сервісно-орієнтована архітектура (SOA). SOA – це метод розробки програмного забезпечення, який використовує програмні компоненти, які називаються службами, для створення бізнес-додатків. Кожна така служба має свої бізнес-можливості і може взаємодіяти з іншими службами незалежно від платформи або мови. Розробники використовують SOA для ефективного використання служб у різних системах або для поєднання декількох незалежних служб для вирішення складних завдань.

Переваги SOA включають наступне [23]:

- Швидкий вихід на ринок: Розробники можуть економити час і гроші, повторно використовуючи служби в різних бізнес-процесах і швидше збирати програми за допомогою SOA, ніж писати код і інтегрувати його з нуля.
- Ефективне обслуговування: Створення, оновлення та налагодження невеликих служб у SOA легше, ніж великих блоків коду у монолітних програмах. Зміна однієї служби у SOA не впливає на загальну функціональність бізнес-процесу.
- Більша адаптивність: SOA краще адаптується до технологічного прогресу, дозволяючи ефективно та економічно модернізувати програми.

Більшість нейронних мереж мають типову структуру, де нейрони розміщені у шарах і передача інформації між шарами відбувається лише в прямому напрямку.

Навчання виконується з використанням раніше створеного набору даних. Вихідні дані моделі включають:

- компанія недооцінена;
- компанія переоцінена;
- компанія правильно оцінена.

Модель приймає вхідні приклади, які складаються з 28 параметрів. Навчання відбувалося для моделей випадкового лісу і градієнтного бустингу. Крім того, були виконані приклади навчання моделі на основі повнозв'язної нейронної мережі. Результати навчання представлені в Таблиці 2.2.

*Таблиця 2.2.*

### **Метрики кращих моделей у своїх топологіях**

Модель	Random forest	Gradient boosting	Dense NN1	Dense NN2
Accuracy	0.6815	0.6712	0.6577	0.6134
Recall	0.6433	0.6311	0.5933	0.6314
Precision	0.621	0.6811	0.5814	0.6211

Було виконано ряд експериментів з вибором оптимальних коефіцієнтів та макропараметрів навчання [6]. Параметри навчання для випадкового лісу:

- `n_estimators=500;`
- `criterion='mse';`
- `oob_score= True;`
- `n_jobs=-1;`
- `random_state=10;`
- `cv=5;`
- `n_jobs=-1;`
- `scoring='neg_mean_squared_error';`

Параметри навчання градієнтного бустинга:

- `n_estimators=500;`

- learning\_rate=0.1;
- loss='huber';
- alpha=0.7;
- subsample=0.9;
- random\_state=10;
- cv=5;
- scoring='neg\_mean\_squared\_error';

Топологія Dense NN1, Optimizer = ADAM:

- 28 relu;
- Dropout 0,4;
- 256 relu;
- Dropout 0,4;
- 256 relu;
- 3 softmax;

Топологія Dense NN2, Optimizer = ADAM:

- 28 relu;
- 128 relu;
- Dropout 0,4;
- 512 relu;
- Dropout 0,4;
- 512 relu;
- 128 relu;
- 3 softmax.

Користувачі вважають пояснення методів розв'язку проблеми важливішими, ніж самі розв'язки. Ця проблема особливо актуальна, коли система самостійно приймає рішення, замінюючи експерта [22].

Для розуміння принципів прогнозування був проведений аналіз значущості та ранжування параметрів (рис. 2.7).

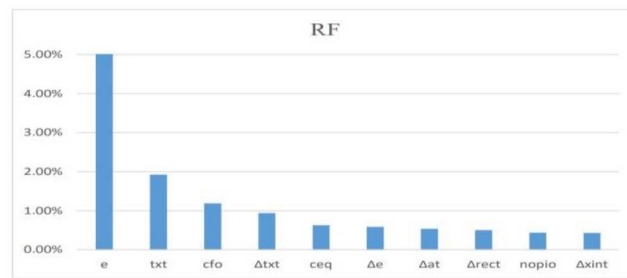


Рисунок 2.7 – Ступінь впливу параметра на результат

Найважливішим фактором є величина прибутку компанії, яка має тісний зв'язок з розміром податкового навантаження. Основна модель базується на реальних світових даних. Проведемо інтерпретацію окремих тестів, використовуючи бібліотеку `shap` [19].

Вектори Шеплі ґрунтуються на концепції проведення серії ігор з невеликою зміною вхідних параметрів. За розміром зміни виходу моделі можна зрозуміти, який параметр є найбільш важливим, так що навіть невелика зміна в цьому параметрі серйозно впливає на результат роботи моделі (рис. 2.8).



Рисунок 2.8 – Вектор впливу параметрів

Прогнозування з використанням бібліотеки `shap` відокремлює найважливіші та впливові параметри та показує, як саме кожен параметр вплинув на результат. Наприклад, у цьому випадку компанія мала річний прибуток нижче середнього для своєї галузі, тому ми спостерігаємо велике переважання синього кольору (що ілюструє негативний вплив параметра).

## Висновки за розділом 2

В другому розділі кваліфікаційної роботи було проведено моделювання структури автоматизованої системи, що включало формалізацію вхідних даних, побудову UML-діаграм предметної області, обґрунтування вибору типу нейронної мережі для прогнозування та розробку архітектури системи.

Аналіз формалізованих вхідних даних дозволив уявити структуру і потреби системи з погляду її функціонування. Побудова UML-діаграм предметної області допомогла візуалізувати взаємозв'язки між різними компонентами системи та виявити можливості для оптимізації її роботи.

Обґрунтування вибору типу нейронної мережі як інструменту прогнозування було проведено на основі аналізу характеристик нейронних мереж та їхньої придатності для розв'язання конкретних завдань у фінансовій сфері.

Результати аналізу вхідних даних, побудови UML-діаграм, обґрунтування вибору нейронної мережі та розробки архітектури допомагають у зрозумінні потреб користувачів та спрощують процес реалізації системи.

Моделювання структури системи відіграє ключову роль у забезпеченні її ефективності, масштабованості та надійності. Основний внесок цього розділу полягає в тому, що він надає чітке розуміння функціональних вимог до системи та сприяє розв'язанню технічних проблем у процесі розробки.

## РОЗДІЛ 3.

### ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

#### 3.1 Вибір середовища розробки автоматизованої системи

Сьогодні існує значна кількість різних програмних засобів і рішень, які дозволяють прогнозувати часові ряди. Більшість з цих засобів є комерційними, що означає, що їх розповсюдження пов'язане з певними витратами – чи то оплата за повний функціонал, чи то використання обмеженої безплатної версії. Це обмежує можливості використання таких програм для особистих цілей безплатно та ускладнює доступ до сучасних методів прогнозування і аналізу часових рядів. Після розгляду особливостей побудови математичних моделей і прогнозування часових рядів, пропонуємо розглянути основні технології, які будемо використовувати для розв'язання завдань кваліфікаційної роботи.

Google Colab, або Google Colaboratory, є безплатним сервісом, що дозволяє запускати та розробляти код на мові Python прямо у веббраузері. Цей сервіс спрощує обмін проєктами та спільну роботу над ними в реальному часі з іншими користувачами. Заснований на відомому фреймворку Jupyter Notebook, він стає зручним інструментом для роботи з даними, машинного навчання, глибокого навчання та інших обчислювальних завдань.

Google Colab надає доступ до обчислювальних ресурсів Google, таких як графічні процесори (GPU) та тензорні процесори (TPU), які використовуються для різних обчислювальних завдань. Розглянемо принципи роботи процесорів CPU, GPU та TPU, а також можливості, які надає Google Colab [26].

Google Colab, або Google Colaboratory, надає можливість створення інтерактивного середовища для експериментів, аналізу даних та навчання моделей. Нижче перераховано функції, які ви можете використовувати у Google Colab:

- тренувати нейронні мережі для класифікації зображень і відео з використанням бібліотек Tensorflow або Pytorch;

- створювати та тренувати моделі для обробки тексту з використанням бібліотек NLTK або Spacy;
- досліджувати та візуалізувати зібрані дані з використанням бібліотек Python, таких як Pandas, Matplotlib та Seaborn;
- навчати моделі для визначення мови та виявлення тональності;
- використовувати бібліотеки Gym для навчання моделей з підкріпленням на середовищах (ігрові симулятори);
- застосовувати алгоритми кластеризації або зменшення розмірності для аналізу даних без міток;
- аналізувати великі набори даних, включаючи дані з хмари або BigQuery;
- обробляти потокові дані з використанням Apache Spark.

CPU є загальноприйнятим типом процесора, який використовується для виконання загальних обчислювальних завдань. У Google Colab CPU використовується для виконання звичайних завдань, таких як обробка даних та виконання коду Python.

GPU – це графічний процесор, доступний у Google Colab від NVIDIA, такий як Tesla K80, Tesla T4 і Tesla P100. GPU використовується винятково для роботи з графікою та виконання задач паралельно, що відрізняє його від CPU.

TPU – тензорний процесор, розроблений Google, призначений для тренування нейронних мереж. TPU має набагато вищу продуктивність шляхом паралельного виконання обчислень, особливо при обробці великих обсягів даних.

Для реалізації проєкту була обрана мова програмування Python, яка відзначається великою кількістю доступних моделей і бібліотек, які можуть бути використані для моделювання та аналізу різноманітних процесів. Завдяки вбудованим бібліотекам, користувач може розширити свої можливості у моделюванні, оскільки ці бібліотеки оптимізовані для виконання різноманітних завдань та представлення отриманих результатів [23].

Розглянемо деякі конкретні інструменти, які використовувалися під час моделювання. Наприклад, бібліотека Scikit-Learn, яка спеціалізується на алгоритмах машинного навчання. Вона призначена для розв'язання різних завдань, таких як класифікація та регресія для навчання з учителем, або кластеризація та зниження розмірності для навчання без учителя. Ця бібліотека також допомагає виявляти аномалії. Ми використовували її для створення та оцінки метрик MSE та RMSE. Для математичних обчислень та інтерпретації результатів використовувалась бібліотека NumPy. Ця бібліотека дозволяє працювати з векторами, матрицями та числовими масивами, а також надає засоби для побудови графіків та гістограм.

Для ефективної обробки наборів даних було використано Pandas – високорівневу бібліотеку, що базується на NumPy. Pandas дозволяє швидко і ефективно працювати з вхідними даними, зчитувати інформацію з файлів та перетворювати її у різноманітні типи даних, такі як словники, масиви або часові ряди. З урахуванням того, що вхідна інформація найчастіше подається у вигляді матриць, Pandas є ефективним рішенням для її зчитування [23].

Для обчислень та машинного навчання була обрана бібліотека Keras, яка використовується для цих цілей разом з різними програмними пакетами. Для обчислень у цьому випадку використовувалась бібліотека TensorFlow, розроблена Google з метою створення ефективних моделей для моделювання та навчання нейронних мереж. Ці бібліотеки значно полегшили використання різних типів нейронних мереж для різних дослідників.

Також було використано бібліотеку statsmodels, яка містить різноманітні методи, класи та функції для оцінки різноманітних статистичних моделей, а також для інших цілей, пов'язаних з моделюванням, таких як вивчення статистичних даних та проведення статистичних тестів. Ця бібліотека була використана для перевірки статистичних результатів моделювання та проведення альтернативного моделювання.

Наведення результатів моделювання у графічному вигляді було забезпечено бібліотекою Matplotlib.

### 3.2 Методика навчання нейронної мережі

Для створення моделі нейронної мережі використовується функція тренування TRAINLM.

TRAINLM є функцією тренування мережі, яка модифікує значення ваг і зсувів згідно з методом оптимізації Левенберга-Маркара.

Вхідні параметри:

- NET – безпосередньо нейронна мережа;
- Pd – вектори вхідних затримок;
- Tl – вектори еталонів шару;
- Ai – початкові умови вхідних затримок;
- Q – розмір пакета;
- TS – часові кроки;
- VV – порожня матриця або структура контрольних векторів;
- TV – порожня матриця або структура тестових векторів;

Вихідні параметри:

- NET – навчена мережа;
- TR – запис, що включає параметри тренування;
- TR.epoch – кількість епох тренування;
- TR.perf – характеристика тренування;
- TR.vperf – характеристика контрольної перевірки;
- TR.tperf – характеристика тестування;
- TR.mu – значення  $\mu$ .

Навчання відбувається відповідно до параметрів функції TRAINLM.

Наведемо ці параметри разом з їх значеннями за замовчуванням:

- net.trainParam.epochs – (100) – Максимальна кількість епох тренування;
- net.trainParam.show – (25) – Кількість епох між графіками;
- net.trainParam.goal – (0) – Умова зупинки за відхиленням від еталону;
- net.trainParam.time – (inf) – Максимальний час тренування в секундах;

- `net.trainParam.min_grad` – (1e-10) – Мінімальний градієнт;
- `net.trainParam.mu` – (0.001) – Початкове значення  $\mu$ ;
- `net.trainParam.mu_dec` – (0.1) – Коефіцієнт зменшення  $\mu$ ;
- `net.trainParam.mu_inc` – (10) – Коефіцієнт зростання  $\mu$ ;
- `net.trainParam.mu_max` – (1e10) – Максимальне значення  $\mu$ ;
- `net.trainParam.mem_reduc` – 1 – Коефіцієнт використання пам'яті;
- `net.trainParam.max_fail` м (5) – Максимальна кількість помилок на контрольному масиві.

Контрольні вектори використовуються для припинення тренування передчасно, якщо показник ефективності мережі на контрольному масиві векторів перестане покращуватися або залишатиметься на одному рівні протягом `MAX_FAIL` послідовних епох.

Функція `TRAINLM (CODE)` повертає корисну інформацію для кожного з рядків `CODE`:

- 'pnames' – Імена параметрів тренування;
- 'pdefaults' – Значення параметрів тренування за замовчуванням.

Стандартну мережу, що використовує `TRAINLM`, можна створити за допомогою функцій `NEWFF`, `NEWCF` або `NEWELM`. Вікно конфігурації параметрів нейромережі показано на рис. 3.1.

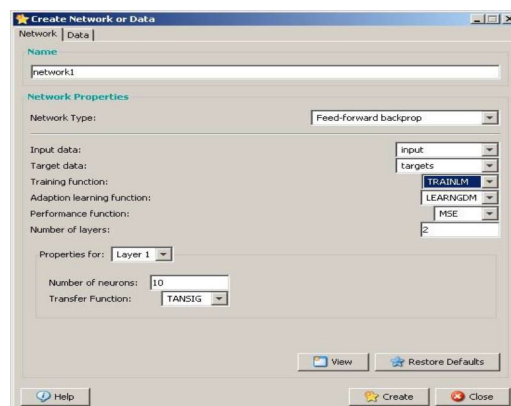


Рисунок 3.1 – Вікно конфігурації параметрів нейромережі

Після встановлення всіх необхідних параметрів у цьому вікні можна використати кнопку «View», щоб візуалізувати структуру розробленої нейронної мережі. Результат представлено на рис. 3.2.

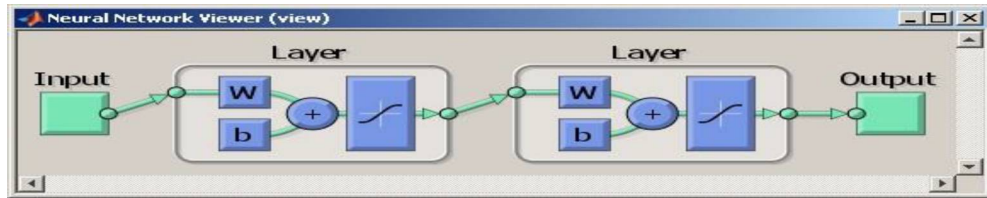


Рисунок 3.2 – Схематичний вигляд структури розробленої нейромережі

Щоб створити мережу для користувача, яку необхідно тренувати за допомогою TRAINLM, було виконано наступне:

1. Встановлено NET.trainFcn як 'trainlm'. Це призведе до встановлення за замовчуванням параметрів NET.trainParam для TRAINLM.

2. Встановлені потрібні значення для властивостей NET.trainParam.

Виклик TRAIN з готовою мережею забезпечить тренування мережі за допомогою TRAINLM. TRAINLM може тренувати будь-яку мережу, якщо її вагові, входні і активаційні функції мають похідні.

Параметр MEM\_REDUС визначає кількість пам'яті, необхідну для обчислення якобіана  $JX$ , а також швидкість обчислень. При MEM\_REDUС = 1 швидкість тренування максимальна, але вимагається найбільший обсяг пам'яті. Зі збільшенням MEM\_REDUС швидкість тренування зменшується, але зменшується і обсяг використовуваної пам'яті.

Тренування припиняється, якщо виконується одна з наступних умов:

1. Досягнуто максимального значення кількості епох «EPOCHS».
2. Перевищено максимальний час тренування «TIME».
3. Ефективність досягає значення «GOAL».
4. Градієнт ефективності зменшується нижче «MINGRAD».
5. Значення MU досягає MU\_MAX.

6. Контрольне значення функціонування збільшується більше, ніж «MAX\_FAIL» разів поспіль після останнього зменшення (при використанні контрольного масиву).

При створенні нейронної мережі потрібно задати відповідні параметри її роботи у вкладках «Train», «Simulate», «Adapt». Вікно для задання вхідних даних для тренування розробленої нейромережі наведено на рис. 3.3.

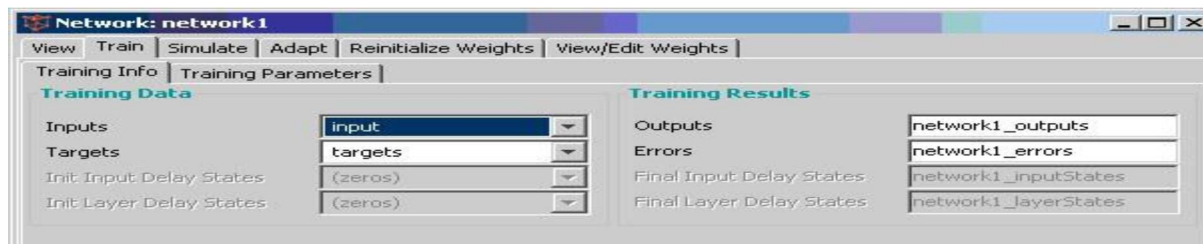


Рисунок 3.3 – Вікно вхідних даних для тренування розробленої нейромережі

Вікно завдання параметрів тренування розробленої нейромережі наведено на рис. 3.4.

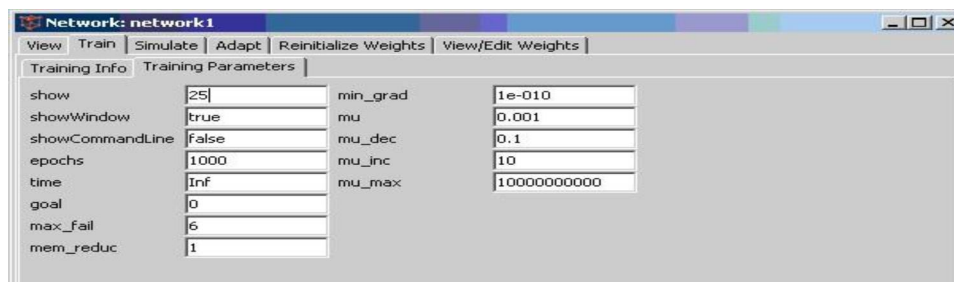


Рисунок 3.4 – Вікно завдання параметрів тренування розробленої нейромережі

Вікно для встановлення параметрів симуляції роботи розробленої нейромережі показано на рис. 3.5.

Вікно для встановлення параметрів адаптації розробленої нейромережі наведено на рис. 3.6. У цих вікнах потрібно вказати вхідні та цільові вектори даних для роботи нейромережі.



Рисунок 3.5. – Вікно завдання параметрів симуляції роботи нейромережі

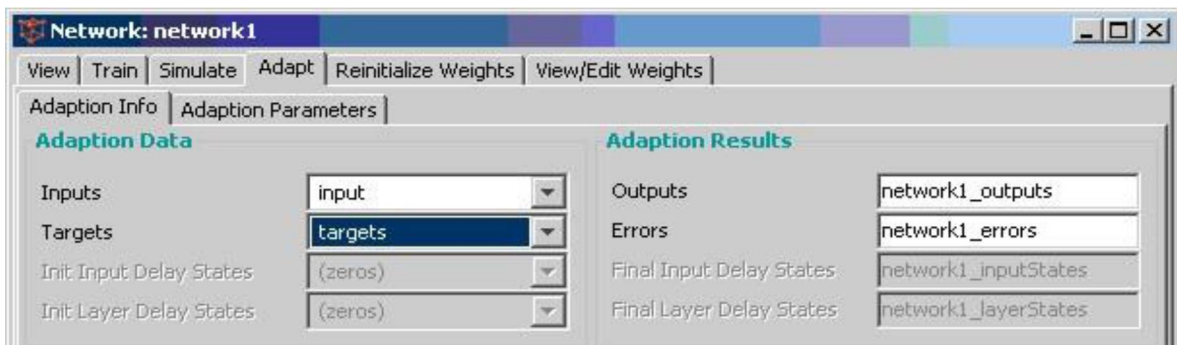


Рисунок 3.6 – Вікно завдання параметрів адаптації розробленої нейромережі

Для моніторингу процесу навчання нейронної мережі використовується інструмент nntaintool. Це вікно можна відкрити, щоб візуально спостерігати за процесом навчання, починаючи з його початку й до завершення.

Вигляд форми візуалізації алгоритму та процесу навчання розробленої нейромережі наведено на рис. 3.7.

Процес навчання можна візуалізувати за допомогою графіка, на якому відображена залежність оцінки функціонування від номера циклу навчання. Графік, який демонструє залежність середньоквадратичної помилки навченої нейромережі від кількості пройдених епох, наведено на рис. 3.8. На цьому графіку тренувальний процес позначено синім кольором, валідаційний –

зеленим, а тестовий – червоним. Точка перетину пунктирних ліній позначає місце найкращого функціонування нейромережі. Після цієї точки мережа починає функціонувати з більшими похибками, що свідчить про початок процесу перенавчання.

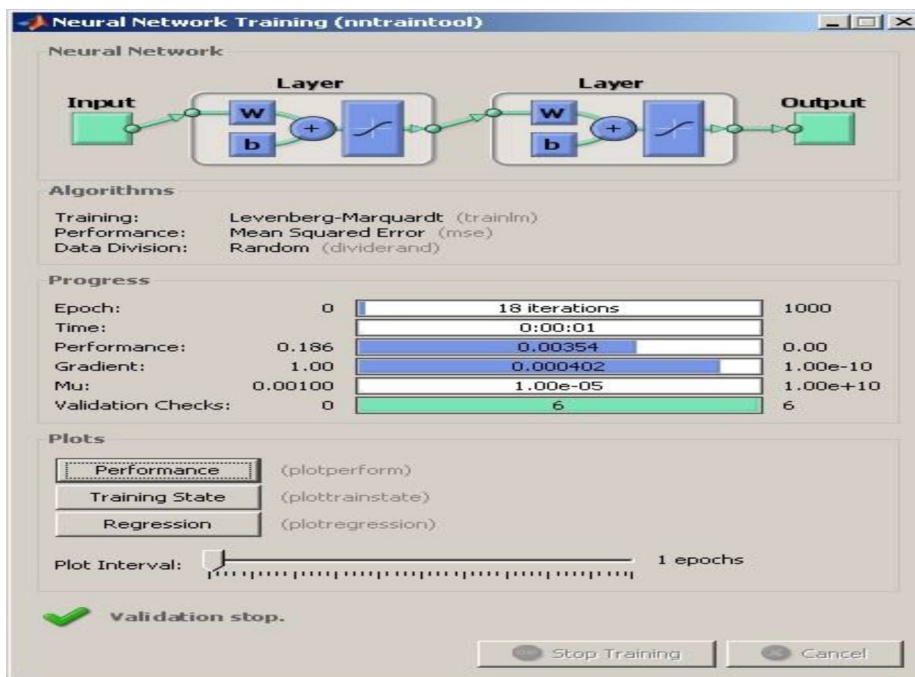


Рисунок 3.7 – Вигляд форми візуалізації алгоритму та процесу навчання

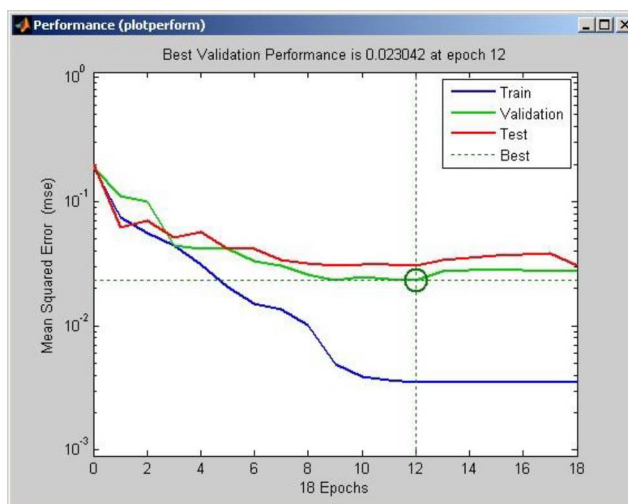


Рисунок 3.8 – Графік залежності середньоквадратичної помилки навченої нейромережі від кількості пройдених епох

Графіки відображення стану тренування нейромережі наведено на рис.3.9.

Перш ніж використовувати нейронну мережу, важливо оцінити достовірність результатів, які вона надає, на тестовому наборі вхідних векторів. Цей тестовий набір має відрізнятися від того, що використовувався для навчання мережі. У нашому випадку, для отримання тестового набору, можна використати програму `mas1`.

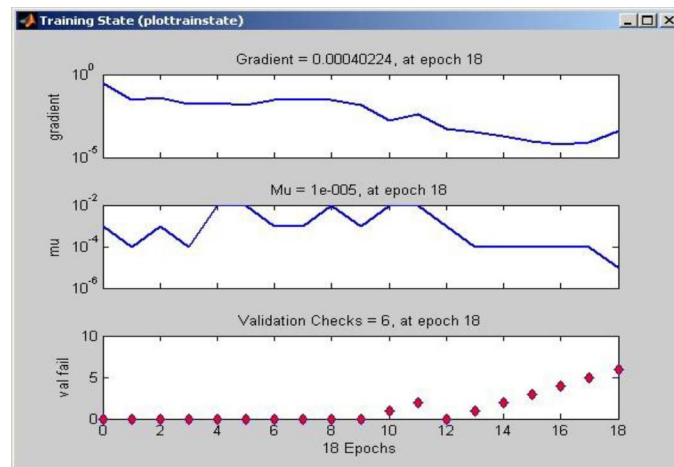


Рисунок 3.9 – Графіки відображення стану тренування нейромережі

Для оцінки достовірності результатів роботи мережі можна порівняти еталонні значення зі значеннями, які мережа видавала на виході, коли на вхід подавалися вхідні вектори тестового набору. У середовищі MATLAB це можна зробити за допомогою функції `postreg` [24].

На рис. 3.10 показано графік, який відображає результати навчання нейромережі на вхідних даних. Еталонні значення векторів відображені пунктирною лінією, окремі значення даних позначені маркерами типу «коло», а кольорова лінія відображає кореляцію по всім досліджуваним значенням моделі.

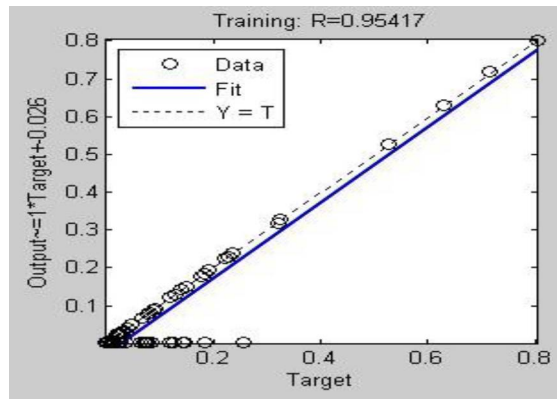


Рисунок 3.10 – Графік відображення результатів навчання нейромережі на вхідній вибірці даних

Графік відображення результатів тренування нейромережі наведено на рис. 3.11.

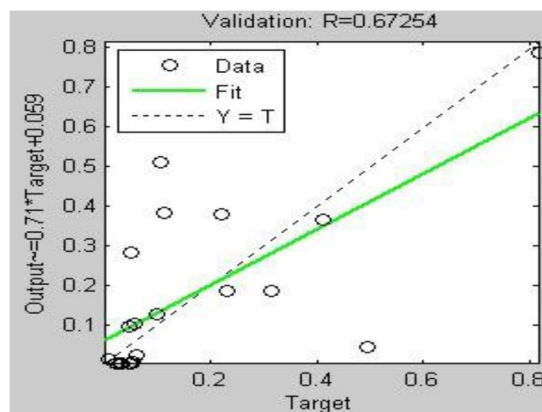


Рисунок 3.11 – Графік відображення результатів валідації нейромережі

Еталонні значення векторів подано у вигляді пунктирної лінії, окремі значення даних позначені маркерами «коло», а кольорова лінія відображає кореляцію по всім вивченим значенням моделі.

Графік, що показує результати тестування нейромережі на перевіркому наборі даних, представлено на рис. 3.12.

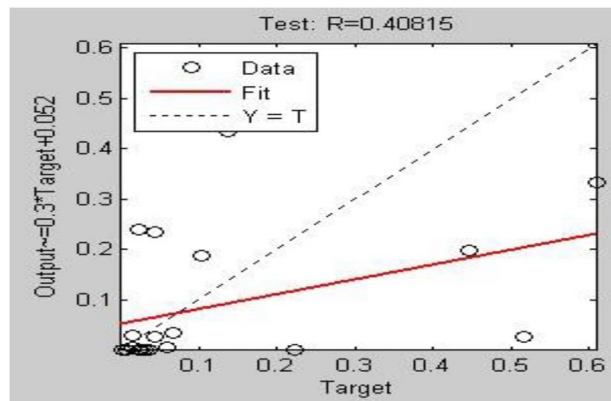


Рисунок 3.12 – Графік відображення результатів тестування нейромережі на перевіркової вибірці даних

Графік, що відображає загальний розподіл даних нейромережі, наведено на рис. 3.13.

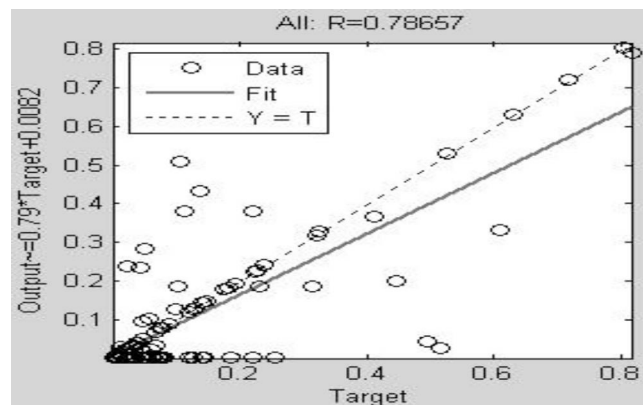


Рисунок 3.13 – Графік відображення загального розподілу даних нейромережі

Перед використанням мережі важливо дослідити її надійність, зокрема шляхом тестування на відокремленому наборі даних. Для оптимізації роботи мережі використовуються різні методи, такі як тренування за допомогою TRAINLM та встановлення параметрів навчання. В цілому, цей процес передбачає кропітку роботу з даними, підготовку мережі та оцінку її ефективності для розв'язання конкретної задачі.

### 3.3 Експериментальне дослідження поведінки фінансових ринків в автоматизованій системі

В роботі була створена автоматизована система – застосунок «Shiny».

Розглянемо його структуру:

1. Опис UI елементів у частині `ui <- fluidPage()`, де `fluidPage` є функцією, що дозволяє автоматично адаптуватися до розмірів вікна браузера користувача.

2. Основна програмна частина описується у функції `server <function(input, output, session){}`.

3. Виклик основної функції `shinyApp(ui = ui, server = server)`.

У `fluidPage` були додані функції `sidebarLayout()` та `mainPanel()` для бокової та основних панелей інтерфейсу. У цьому інтерфейсі використовуються такі функції:

- `fileInput` – для завантаження файлу в систему;
- `selectInput` – для вибору із списку;
- `actionButton` – для створення кнопки;
- `textOutput` – для відображення тексту;
- `plotOutput` – для відображення графіку;
- `verbatimTextOutput` – для відображення тексту у консольному стилі.

Деякі елементи були розроблені за допомогою проєкту Twitter Bootstrap. Кожен елемент має свій ідентифікатор, що полегшує звернення до них у серверній функції. Також доступна підтримка деяких HTML5 елементів, наприклад, `<p>`, `<h1>`–`<h5>`. Ці інструменти дозволяють створювати інтерактивні інтерфейси, подібні до того, як показано на рис. 3.14.

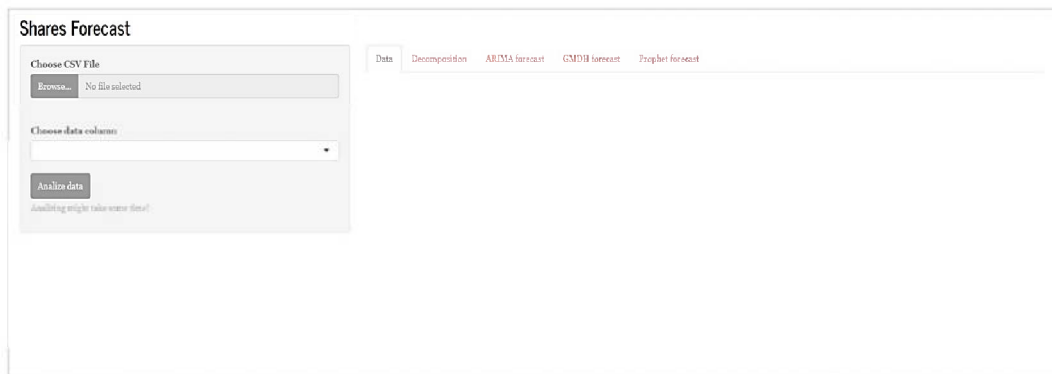


Рисунок 3.14 – Головна сторінка інформаційної системи

На вкладці декомпозиції показується графік часового ряду. Після завантаження файлу, дані необхідно перетворити у формат `tsibble` за допомогою відповідного пакету (`as_tsibble(dataset, key=NULL, index=Date)`). Формат `tsibble` передбачає, що кожне спостереження має обов'язковий індекс (змінна з впорядкованим розташуванням від минулого до теперішнього моменту) та ключ (набір змінних, що визначають одиниці в часі). У разі регулярного розподілу спостережень кожному одиницю часового інтервалу слід вимірювати.

Для декомпозиції ряду потрібно перетворити його у формат часових рядів за допомогою функції `ts()` (`timeSeries <- ts(data=dataset, [...], frequency = 30)`). Розбиття часового ряду на сезонні, тенденційні та нерегулярні компоненти відбувається за допомогою функції `stl()` (`stlData <- stl(timeSeries[,1],s.window="periodic")`), результат якого на рис. 3.15.

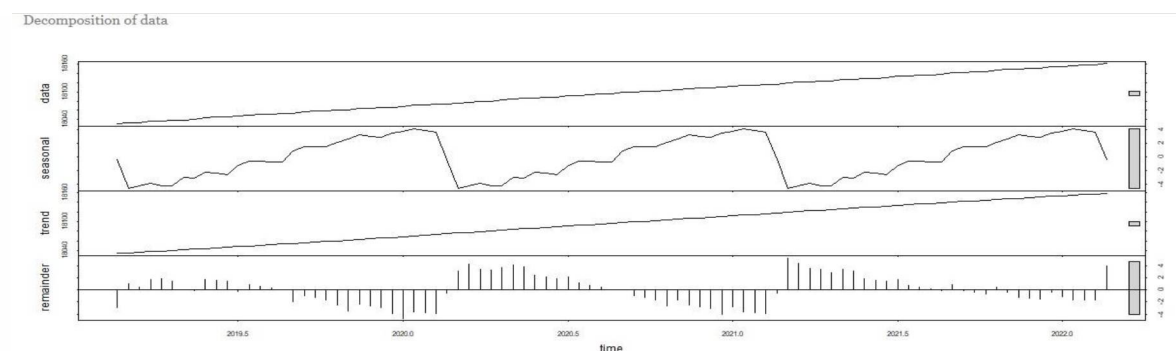


Рисунок 3.15 – Графік декомпозиції ряду

На трьох нижніх графіках показані складові часового ряду: тренд (trend), сезонна компонента (seasonal) та залишкова частина (remainder).

Під час проведення дослідження були використані датасети, що були вже показані раніше. Як статистичну модель ми використовували ARIMA. Для демонстрації роботи модифікацій рекурентних нейронних мереж ми обрали дві модифікації – LSTM та GRU.

На рис. 3.16 наведена візуалізація реальних даних для першого датасету.

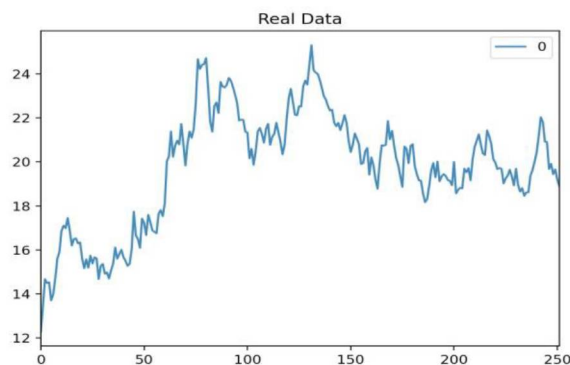


Рисунок 3.16 – Візуалізація реальних даних для першого датасету

Для аналізу даних ми використали модель ARIMA (3,2,1) та виконали пристосування коефіцієнтів моделі до реальних значень за допомогою методу `fit()`. Результати моделювання, які ми отримали за допомогою математичної моделі, будуть представлені через візуалізацію їх похибок на рис. 3.17.

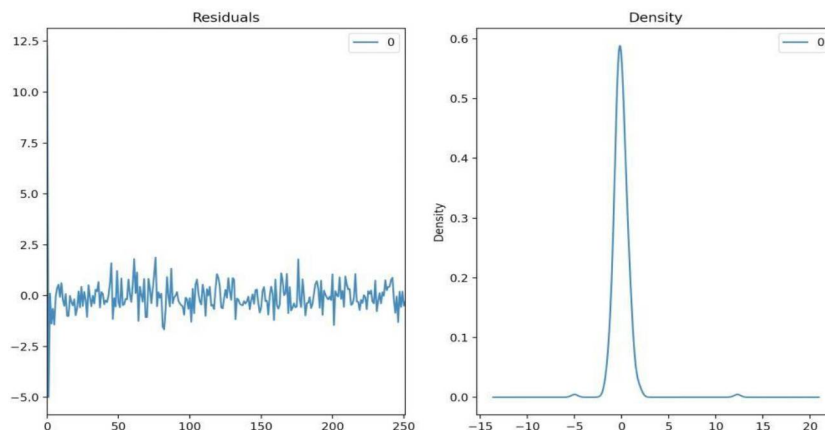


Рисунок 3.17 – Похибки математичної моделі

Процес навчання тривав 100 епох. Для цієї моделі отримані наступні значення метрик для прогнозованих даних:  $RMSE = 0.60556$ ,  $MAE = 0.44781$ ,  $DW = 1.33420$ ,  $MAPE = 2.26591$ . Коефіцієнт  $DW$  вказує на те, що у залишків моделі спостерігається автокореляція.

Результати прогнозування наведені на рис. 3.18. Тут синім кольором представлені реальні дані, червоним – результати прогнозування тренувальних даних, а зеленим – результати прогнозування тестового набору даних.

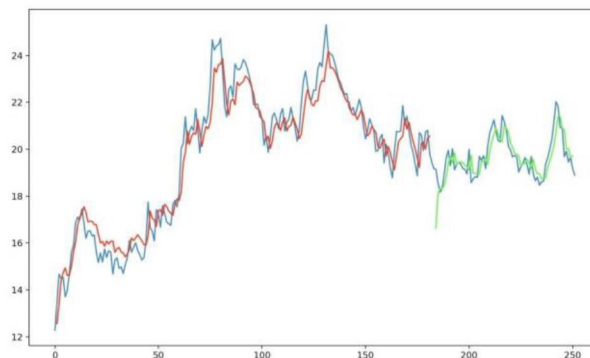


Рисунок 3.18 – Результати прогнозування за допомогою рекурентної нейронної мережі з довгою короткостроковою пам'яттю

Збільшити кількість епох навчання у рекурентній нейронній мережі з довгою короткостроковою пам'яттю до 200. При цьому всі інші параметри нейронної мережі, такі як активаційні функції, кількість шарів, функцію витрат і метод пошуку, залишимо незмінними. Показники помилок моделі відображені на рис. 3.19.

У цьому випадку ми отримали наступні значення метрик:  $RMSE = 0.52503$ ,  $MAE = 0.40758$ ,  $DW = 2.12501$ ,  $MAPE = 2.06534$ . Важливо відзначити, що значення  $DW$  стало ближчим до 2, що може свідчити про те, що при продовженому тренуванні моделі автокореляція у залишках зменшується. Також помітно, що інші статистичні показники для цієї моделі також істотно покращилися.

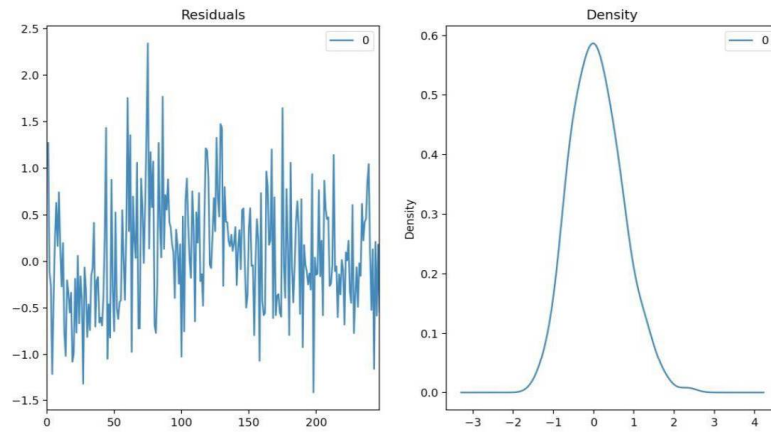


Рисунок 3.19 – Похибки моделювання за допомогою рекурентної нейронної мережі з довгою короткостроковою пам'яттю при кількості епох 200

Результати прогнозування представлені на рис. 3.20.

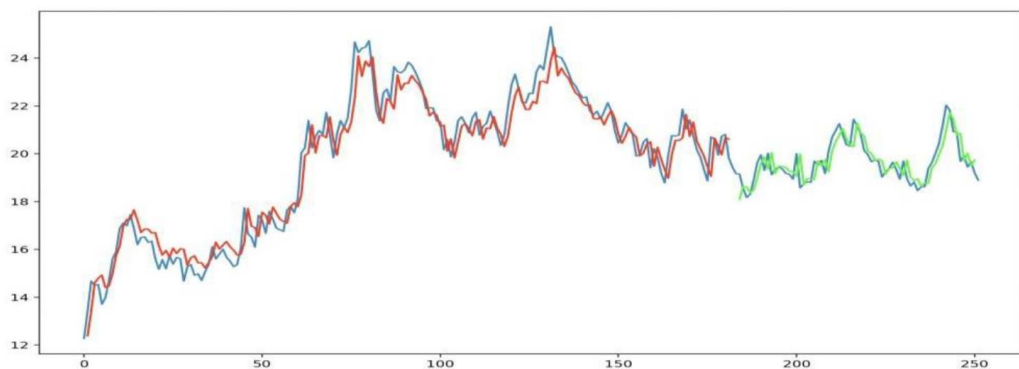


Рисунок 3.20 – Результати прогнозування за допомогою рекурентної нейронної мережі з довгою короткостроковою пам'яттю при кількості епох 200

Компанія «Coca-Cola» є публічною компанією. Її акції торгуються на Нью-Йоркській фондовій біржі під тикером «КО». Тікер «КО» має схожість з індексом S&P 500, адже є лідером в індустрії безалкогольних напоїв, маючи при цьому малу волатильність, що зумовлено наближенням до індексу.

На рис. 3.21. наведено передбачення ціни акцій за допомогою методу випадкового лісу. Графік передбачень ціни акцій виявився більш волатильним, ніж реальний, проте кінцева ціна на середину січня збігається в одній точці.

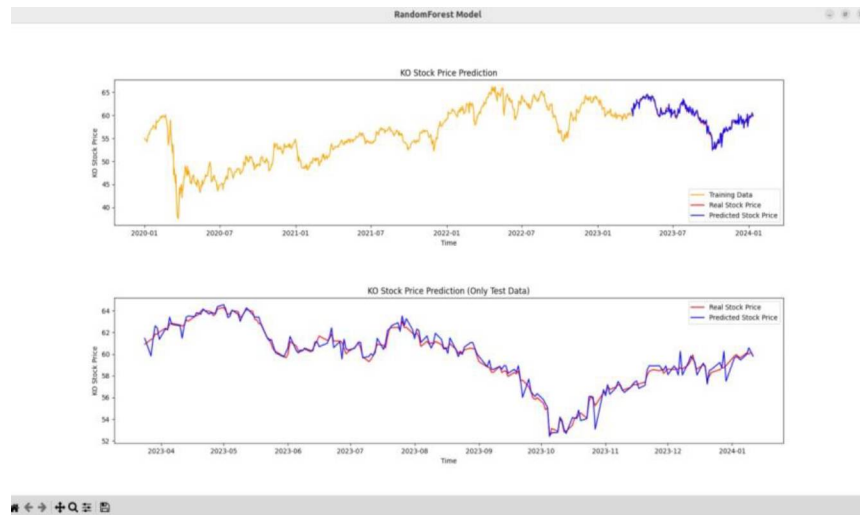


Рисунок 3.21 – Алгоритм Random Forest Regression для тікера «КО»

На рис. 3.22 зображено передбачення ціни акцій за допомогою методу LSTM. Порівняно з методом Random Forest Regression, LSTM має менш волатильний графік, який більше наближений до реальних цін. Ще одна важлива різниця полягає в тому, що кінцева ціна на середину січня збігається в одній точці.

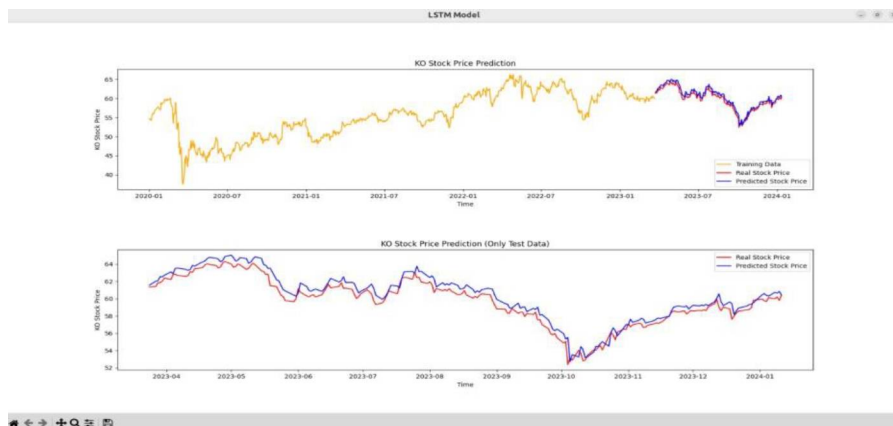


Рисунок 3.22 – Алгоритм LSTM для тікера «КО»

На рис. 3.23 відображено передбачення ціни акцій за допомогою методу ARIMA. Цей метод, здається, найбільш адаптований для такої задачі, оскільки передбачена цінова динаміка максимально наближена до реальної, з багатьма точками зіткнення, попри те, що фактична ціна акцій на останню дату впала, порівняно з передбачуваною.

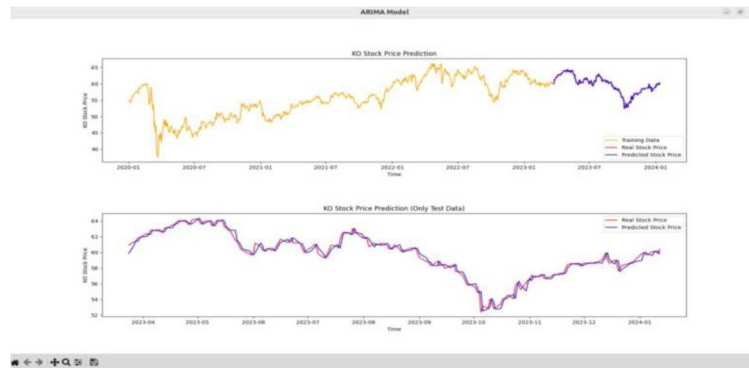


Рисунок 3.23 – Алгоритм ARIMA для тікера «КО»

Можна зазначити, що модель групового урахування аргументів виявилася найточнішою за всіма показниками, у той час, як модель Prophet показала кращі результати порівняно з ARIMA, але гірше за модель групового урахування аргументів. Важливо відзначити, що точність моделі Prophet може змінюватися залежно від кількості точок злому, і швидкість її обчислення досить низька – у цьому випадку розрахунок зайняв близько двох хвилин.

Модель ARIMA показала найгірші результати, що можна пояснити неправильним підбором значень порядків  $p$  (автокореляція),  $d$  (інтегроване),  $q$  (коливання) для моделі. Функція `auto.arima()` автоматично визначає ці порядки, але також дозволяє користувачу вручну змінювати їх. Це свідчить про можливість розширення функціонала інформаційної системи шляхом дозволу користувачеві налаштовувати ці параметри.

### **Висновки за розділом 3**

Було проведено програмну реалізацію автоматизованої системи, включно з вибором середовища розробки, методики навчання нейронної мережі та експериментальне дослідження поведінки фінансових ринків в системі.

Вибір середовища розробки є важливим кроком у процесі реалізації системи. Аналіз хмарної технології Google Colab та використання Python-бібліотек для роботи з нейронними мережами показали їхню ефективність та зручність у використанні. Також, вивчення принципів роботи механізмів автоматичного зчитування даних в API дозволило налагодити автоматизований процес обміну даними між системою та зовнішніми джерелами.

Методика навчання нейронної мережі була розроблена з урахуванням специфіки фінансових даних та потреб системи. Це дозволило ефективно використовувати навчену модель для прогнозування стану фінансових ринків.

Експериментальне дослідження поведінки фінансових ринків в автоматизованій системі дозволило оцінити точність та надійність розробленої системи. Аналіз результатів експериментів допоміг зрозуміти переваги та обмеження системи та виявити можливості для її подальшого вдосконалення.

## ВИСНОВКИ

У кваліфікаційній роботі було розроблено та впроваджено автоматизовану систему аналізу та прогнозування фінансових ринків на основі методів машинного навчання. В процесі дослідження було проведено огляд теоретичних аспектів аналізу фінансових ринків та методів машинного навчання, зібрано та оброблено великі обсяги даних, розроблено та навчено моделі машинного навчання, а також розроблено програмне забезпечення для автоматизованої системи.

Також було проведено аналіз та класифікація статистичних моделей, які використовуються для моделювання поведінки та прогнозування стану фінансових ринків. Визначені основні характеристики та переваги цих моделей у контексті фінансового аналізу.

Проведено огляд та класифікація різних типів нейронних мереж, які використовуються для прогнозування фінансових ринків. Визначені основні характеристики кожного типу мережі та їхнє застосування в фінансовому аналізі.

У результаті моделювання структури автоматизованої системи було розроблено чітку та логічну архітектуру, яка забезпечує ефективну роботу системи з аналізу та прогнозування фінансових ринків. Обрано відповідний тип нейронної мережі, який відповідає поставленим цілям та завданням дослідження.

Було показано, що обрані середовище розробки, Python-бібліотеки та методи роботи з даними дозволяють ефективно реалізувати автоматизовану систему аналізу та прогнозування фінансових ринків.

Експериментальні результати підтвердили ефективність системи та її можливість успішного використання у практичній діяльності для прийняття обґрунтованих фінансових рішень.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрусенко Ю.О. Аналіз основних моделей прогнозування часових рядів. *Збірник наукових праць Харківського національного університету Повітряних Сил*. 2020. Вип. № 3 (65). С. 91 – 96.
2. Берзлев О.Ю. Сучасний стан інформаційних систем прогнозування часових рядів. *Управління розвитком складних систем*. 2013. Вип. № 13. С. 78 – 82.
3. Берзлев О.Ю. Метод прогнозування знаків приростів часових рядів. *Eastern-European Journal of Enterprise Technologies*. 2013. Вип. № 2 (62). С. 8 – 11.
4. Берзлев О.Ю. Методи ідентифікації моментів зміни тенденцій часового ряду для вироблення стратегій прийняття рішень на фінансовому ринку. *Системи обробки інформації*. 2013. Вип. 9 (116). С. 194 – 199.
5. Берзлев О.Ю. Методика передпрогнозного фрактального аналізу часових рядів. *Управління розвитком складних систем*. 2013. Вип. №16. С. 76 – 81.
6. Берзлев О.Ю. Інформаційна система для прогнозування і прийняття рішень у фінансовій сфері. *Управління розвитком складних систем*. 2014. Вип. 18. С. 106 – 111.
7. Бідюк П.І., Романенко В.Д., Тимошук О.Л. Аналіз часових рядів : навч. посіб. Київ : Політехніка, 2010. 317 с.
8. Бідюк П.І., Гожий О.П., Коршевнюк Л.О. Комп'ютерні системи підтримки прийняття рішень : навч. посіб. Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2012. 380 с.
9. Булка Б.А. Особливості та імплікації застосування машинного навчання для створення музичного матеріалу. *Architecture and art*. 2020. Вип. № 10. С. 24 – 27.

10. Джерело фінансової інформації *Investopedia*.  
URL: <https://www.investopedia.com/terms/c/closingprice.asp> (дата звернення: 05.05.2024).

11. Дяченко П.В., Максимов А.Є. Побудова моделі прогнозу вартості акцій на основі використання МГВА. *Інформаційні технології в освіті, науці і техніці* : тези V Міжнар. наук.-практ. конф., м. Черкаси, 21-23 травня 2020 р. / ЧДТУ. Черкаси, 2020. С. 58 – 60.

12. Карпенко А.Н., Якимчук В.С., Носовець О.К. Інтелектуальний аналіз даних : метод. вказівк. Київ : НТУУ «КПІ ім. І. Сікорського», 2017. 51 с.

13. Катренко А.В. Системний аналіз об'єктів та процесів комп'ютеризації : навч. посіб. Львів : «Новий світ-2010», 2011. 424 с.

14. Качаловський А.С. Методи і моделі прогнозування лінійних та нелінійних нестационарних процесів. *International Scientific Journal*. 2016. Вип. № 5. С. 52 – 60.

15. Качаловський А.С. Моделювання і прогнозування нелінійних нестационарних процесів на основі Байєсівських структурних часових рядів . *International Scientific Journal*. 2016. Вип. № 6. С. 33 – 36.

16. Кучанський О.Ю., Білощицький А.О. Прогнозування часових рядів методом селективного зіставлення зі зразком. *Європейський журнал передових технологій*. 2015. Вип. №6 (78). С. 13 – 18.

17. Лесна А.С., Мінаков А.Г. Ефективність використання нейромережевих моделей для прогнозування руху цін, акцій компаній на ринку. *НаукаОнлайн*. 2019. Вип. № 7. С. 49 – 58.

18. Мاستицький С.Е. Аналіз часових рядів за допомогою R. Київ : Наука, 2020. 304 с.

19. Остапчук К.К. Основи машинного навчання. Харків : Фоліо, 2019. 464 с.

20. Полісна В.Й. Прогнозування та аналіз часових рядів. Чернігів : Видавництво ЧНТУ, 2018. 188 с.

21. Полісна В.Й. Моделі та методи аналізу часових рядів. Чернігів : Видавництво ЧНТУ, 2019. 147 с.
22. Томашевський В.М. Моделювання систем. Київ : Нора-Друк, 2015. 352 с.
23. Чабаненко Д.М. Математичні моделі та методи прогнозування часових рядів на основі складних ланцюгів Маркова Черкаси : Видавництво ЧНУ імені Богдана Хмельницького, 2012. 183 с.
24. Чернодуб А.М. Навчання рекурентних нейронних мереж методом псевдорегуляризації для багатокрокового прогнозування часових рядів. *Математичні машини і системи*. 2012. Вип. № 4. С. 41 – 60.
25. Шаркаді М.М., Роботишин М.В., Маляр М.М. Моделі і методи машинного навчання для завдань передбачення. *Науковий вісник Ужгородського університету*. 2020. Вип. № 36. С. 112 – 122.
26. Box G.P., Jenkins G.M. Time series analysis: Forecasting and control. San Francisco : Holden-Day, 1976. 575 p.
27. Chatfield C. The Analysis of Time Series. An Introduction with R. New York : Taylor & Francis Group, 2019. 414 p.
28. Chatfield C. The Analysis of Time Series. An Introduction with Python. New York : Taylor & Francis Group, 2021. 508 p.
29. Christopher O. Understanding LSTM Networks. San Francisco : Holden-Day, 2017. 459 p.
30. Dag O., Yozgatligil, C. GMDH: An R Package for Short Term Forecasting via GMDH-Type Neural Network Algorithms. London : McOreilly, 2020. 348 p.
31. Dag O. Learning Python: Powerful Object-Oriented Programming. London : McOreilly, 2022. 451 p.
32. How To Use the Yahoo Finance API in 2021. Massachusetts : MIT, 2021. 438 p.
33. Hyndman R. J. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*. 2008. Issue № 27(3). P. 2 – 22.

34. Jason B. Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Massachusetts : MIT, 2020. 337 p.
35. Park J Analysis of Recurrent Neural Network and Predictions. *Symmetry*. 2020. Issue №12. P 615 – 618.
36. Peters E. Fractal market analysis. Applying chaos theory to investment and economic. New York : John Wiley & Sons, 1994. 336 p.
37. Python 3.10.5 documentation. *Python*. URL: <https://docs.python.org/3/> (дата звернення: 05.05.2024).
38. Python 3.12.0 documentation. *Python*. URL: <https://docs.python.org/8/> (дата звернення: 05.05.2024).
39. Shumway R. Time Series Analysis and Its Applications. With R examples. Berlin : Springer Science, 2019. 202 p.
40. Siami-Namini S. Forecasting Economic and Financial Time Series: ARIMA vs. LSTM. *Cybernetics and Systems*. 2020. Vol. №27. P 32 – 41.
41. Singh S. Pattern Modeling in Time-Series Forecasting. *Cybernetics and Systems*. 2021. Vol. №31. P. 49 – 65.
42. Vercellis C. Business intelligence: Data mining and optimization for decision making. Cornwall : John Wiley & Sons, 2009. 417 p.
43. Visscher P.E. Forecasting Time Series with Artificial Neural Networks. *Mathematical Institute Utrecht University*. 2018. Issue №3. P. 76 – 81.
44. Wong F.S., Wang P.Z., Goh T.H. Fuzzy neural systems for stock selection. *Financial Analysts Journal*. 2021. Issue №4. P 47 – 52.
45. Yahoo Finance. *Yahoo*. URL: <https://finance.yahoo.com/> (дата звернення: 05.05.2024).

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

Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**  
галузь знань: 15 – Автоматизація та приладобудування  
спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної  
та прикладної системотехніки

д.т.н., проф. Шматков С. І.

«21» грудня 2023 року



**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**Кузинської Катерини Дмитрівни**

(прізвище, ім'я, по батькові студентки)

1. Тема роботи: **«Автоматизована система аналізу та прогнозування фінансових ринків на основі машинного навчання»**

Керівник роботи: Чуб Ольга Ігорівна – к.е.н., доцент кафедри ТПС  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити:

- 1) Дослідження інструментарію прогнозування фінансових показників методами машинного навчання.
- 2) Класифікація нейронних мереж, аналіз принципів їх функціонування.
- 3) Обґрунтування вибору типу нейронної мережі як інструменту прогнозування.
- 4) Розробка архітектури автоматизованої системи аналізу та прогнозування фінансових ринків.
- 5) Опис методики навчання нейронної мережі.
- 6) Експериментальне дослідження поведінки фінансових ринків за допомогою нейронної мережі.
- 7) Підготовка практичних рекомендацій щодо впровадження та використання системи аналізу та прогнозування фінансових ринків.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Формулювання наукової проблеми, обґрунтування актуальності, мети та завдань кваліфікаційної роботи, узгодження об'єкта, предмета та методів дослідження, а також формулювання очікуваних практичних результатів	19.12.2023 – 02.01.2024
2	Аналіз актуальних наукових досліджень та публікацій з теми кваліфікаційної роботи	02.01.2024 – 25.01.2024
3	Опис сучасного інструментарію інтелектуального аналізу даних фінансових ринків	25.01.2024 – 14.02.2024
4	Характеристика типів нейронних мереж	14.02.2024 – 18.02.2024
5	Виклад матеріалу першого розділу кваліфікаційної роботи	18.02.2024 – 26.02.2024
6	Побудова UML-діаграм предметної області	26.02.2024 – 03.03.2024
7	Розробка архітектури автоматизованої системи прогнозування поведінки фінансових ринків	03.03.2024 – 21.03.2024
8	Виклад матеріалу другого розділу кваліфікаційної роботи	21.03.2024 – 28.03.2024
9	Реалізація методики навчання нейронної мережі в середовищі розробки	28.03.2024 – 15.04.2024
10	Експериментальне дослідження поведінки фінансових ринків в автоматизованій системі	15.04.2024 – 22.04.2024
11	Тестування системи та підготовка практичних рекомендацій для користувачів	22.04.2024 – 30.04.2024
12	Виклад матеріалу третього розділу кваліфікаційної роботи	30.04.2024 – 08.05.2024
13	Підготовка презентації до пояснювальної записки кваліфікаційної роботи	08.05.2024 – 14.05.2024
14	Оформлення пояснювальної записки кваліфікаційної роботи та проходження перевірки на антиплагіат	14.05.2024 – 25.05.2024
15	Передзахист кваліфікаційної роботи	14.05.2024 – 25.05.2024
16	Підготовка супровідних документів та презентації до захисту кваліфікаційної роботи перед атестаційною комісією.	30.04.2024 – 27.05.2024

5. Дата видачі завдання: 21.12.2023

Студент К.Д. Кузінська



Керівник роботи О.І. Чуб



**Технічне завдання  
на розробку програмного виробу**

**«Автоматизована система аналізу та прогнозування фінансових ринків  
на основі машинного навчання»**

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва програмного виробу: Автоматизована система аналізу та прогнозування фінансових ринків на основі машинного навчання. 1.2. Галузь застосування: Фінансовий сектор, біржовий сектор.
2. Підстава для розробки	2.1. Робочий навчальний план спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології. 2.2. Завдання на дипломну роботу бакалавра, затверджено наказом по університету № 4101-5/909 від 3 травня 2024 року.
3. Призначення розробки	3.1. Мета розробки: Підвищення прогнозної точності цін на акції. 3.2. Призначення розробки: Аналіз та прогнозування біржової інформації методами машинного навчання. 3.3. Початкові дані для розробки: бази фінансових даних аналітичних платформ Yahoo Finance та Google Finance.
4. Технічні вимоги до програмного виробу	4.1. Функціональні характеристики: Система забезпечує збір, аналіз, візуалізацію даних, а також побудову прогнозів у реальному часі. 4.2. Надійність: Система забезпечує безперервну роботу, високу точність прогнозів, стійкість від збоїв, обробку великих обсягів даних. 4.3. Умови експлуатації: Система працює на основних операційних системах, таких як Windows, macOS і Linux. 4.4. Технічні засоби: Система забезпечує можливість отримувати фінансову інформацію за допомогою API. 4.5. Сумісність: Система має здатність інтегруватися з іншими фінансово-аналітичними системами. 4.6. Маркування та упаковка: Система не має спеціальних вимог. 4.7. Транспортування та зберігання: Немає вимог, оскільки продукт цифровий. 4.8. Спеціальні вимоги: Система забезпечує конфіденційність фінансових даних та захищає від несанкціонованого доступу результату аналізу.
5. Вимоги до програмної документації	Програмою документацією системи є: 1) Технічне завдання на розробку програмного виробу (Додаток Б); 2) Програма та методика випробувань програмного виробу (Додаток В); 3) Опис програмного виробу (Розділ 2 та Розділ 3 Пояснювальної записки до кваліфікаційної роботи бакалавра).

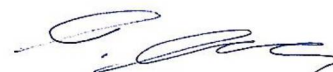
6. Техніко-економічні показники	Орієнтовна оцінка ефективності системи та якості аналізу, що виконується, повинна бути вище 90%		
7. Стадії і етапи розробки	№ з/п	Назви етапів роботи	Термін виконання
	1	Формулювання наукової проблеми, обґрунтування актуальності, мети та завдань кваліфікаційної роботи, узгодження об'єкта, предмета та методів дослідження, а також формулювання очікуваних практичних результатів	19.12.2023 – 02.01.2024
	2	Аналіз актуальних наукових досліджень та публікацій з теми кваліфікаційної роботи	02.01.2024 – 25.01.2024
	3	Опис сучасного інструментарію інтелектуального аналізу даних фінансових ринків	25.01.2024 – 14.02.2024
	4	Характеристика типів нейронних мереж	14.02.2024 – 26.02.2024
	5	Побудова UML-діаграм предметної області	26.02.2024 – 03.03.2024
	6	Розробка архітектури автоматизованої системи прогнозування поведінки фінансових ринків	03.03.2024 – 28.03.2024
	7	Реалізація методики навчання нейронної мережі в середовищі розробки	28.03.2024 – 15.04.2024
	8	Експериментальне дослідження поведінки фінансових ринків в автоматизованій системі	15.04.2024 – 22.04.2024
	9	Тестування системи та підготовка практичних рекомендацій для користувачів	22.04.2024 – 30.04.2024
8. Порядок контролю і приймання	<p>1) Внутрішній контроль: Проводиться керівником кваліфікаційної роботи на кожному етапі розробки для забезпечення відповідності технічному завданню і стандартам якості;</p> <p>2) Тестування програмного виробу: Проводиться керівником кваліфікаційної роботи разом з виконавцем після остаточного виконання Технічного завдання на розробку виробу;</p> <p>3) Приймання роботи: Проводиться на засіданні Атестаційної комісії № 44 на базі Харківського національного університету імені В.Н. Каразіна.</p>		

**Виконавець:**

студентка групи КУ-41  
Кузинська К.Д.


**Замовник:**

к.е.н., доцент кафедри теоретичної  
та прикладної системотехніки  
Чуб О.І.



## **Програма і методика випробувань програмного виробу**

«Автоматизована система аналізу та прогнозування фінансових ринків на основі машинного навчання»

### **1. Об'єкт випробувань**

1.1. Назва програмного виробу: «Автоматизована система аналізу та прогнозування фінансових ринків на основі машинного навчання»

1.2. Область застосування: Фінансовий сектор, біржовий сектор.

### **2. Мета випробувань**

Перевірка відповідності функціональності програмної системи заявленим у Технічному завданні функціональним можливостям (Додаток Б до пояснювальної записки).

### **3. Загальні положення**

3.1. Підстави для проведення випробувань: Наказ по університету про призначення Атестаційної комісії № 44.

3.2. Місце і тривалість випробувань: Випробування проводяться у віртуальному лабораторному середовищі протягом роботи Атестаційної комісії № 44.

3.3. Обсяг випробувань: Приймання та тестування програмного виробу проводяться в обсягах, які відповідають пункту 3.2 цієї Програми.

3.4. Організації, які беруть участь у випробуваннях: Приймання проводиться в процесі засідання Атестаційної комісії № 44 за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

### **4. Вимоги до програми або програмного виробу**

Система повинна задовольняти наступні вимоги:

1) Функціональні характеристики: Система забезпечує збір, аналіз, візуалізацію даних, а також побудову прогнозів у реальному часі.

2) Надійність: Система забезпечує безперервну роботу, високу точність прогнозів, стійкість від збоїв, обробку великих обсягів даних.

3) Умови експлуатації: Система працює на основних операційних системах, таких як Windows, macOS і Linux.

4) Технічні засоби: Система забезпечує можливість отримувати фінансову інформацію за допомогою API.

5) Сумісність: Система має здатність інтегруватися з іншими фінансово-аналітичними системами.

6) Спеціальні вимоги: Система забезпечує конфіденційність фінансових даних та захищає від несанкціонованого доступу результати аналізу.

## 5. Вимоги до програмної документації

Програмною документацією системи є:

- 1) Технічне завдання на розробку програмного виробу (Додаток Б);
- 2) Програма та методика випробувань програмного виробу (Додаток В);
- 3) Опис програмного виробу (Розділ 2 та Розділ 3 Пояснювальної записки до кваліфікаційної роботи бакалавра).

## 6. Засоби і порядок випробувань

6.1. Засоби випробувань: Для проведення випробування необхідне середовище, яке підтримує мову програмування Python та має відповідні бібліотеки.

6.2. Порядок проведення випробувань: Випробування програмного засобу відбувається за допомогою проведення 3 тестів.

*Тест 1: Випробування здатності системи до декомпозиції часових рядів*

- Мета тесту: Перевірка алгоритму декомпозиції ряду на сезонність, тренд та шумові компоненти.
- Опис процедури: Система отримує на вхід часовий ряд з історичними даними фінансового ринку, проходить процес декомпозиції, після чого результати представляються у формі графіка декомпозиції ряду.
- Використання графіка: «графік декомпозиції ряду» використовується для візуалізації результатів декомпозиції (рис. В.1).

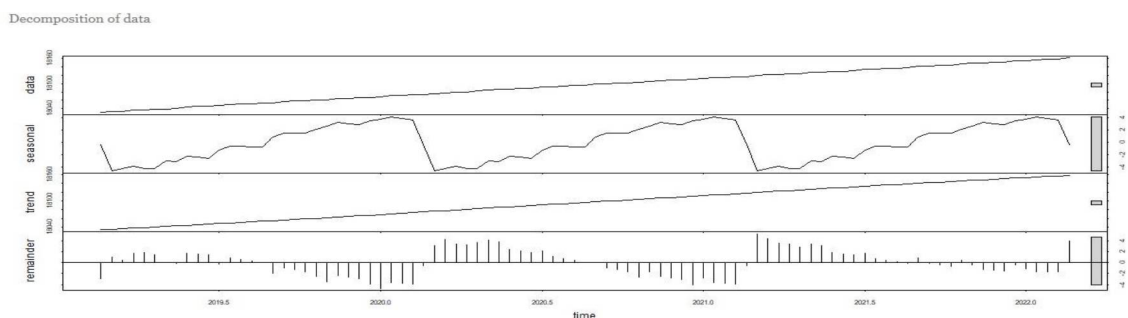


Рис. В.1 – Графік декомпозиції ряду

*Тест 2: Тестування інтерфейсу системи та взаємодії з користувачем*

- Мета тесту: Перевірка зручності інтерфейсу системи, легкості завантаження даних та налаштування параметрів нейромережі.
- Опис процедури: Користувач через головну сторінку інформаційної системи вибирає файли для аналізу, налаштовує параметри нейромережі в конфігураційному вікні та ініціює процес навчання.
- Використання графіків та вікон:
  1. «Головна сторінка інформаційної системи» для демонстрації входу в систему (рис. В.2);
  2. «Панель вибору файлу та колонки з даними» для відображення процесу завантаження даних (рис. В.3);
  3. «Вікно конфігурації параметрів нейромережі» (рис. В.4) та «Схематичний вигляд структури розробленої нейромережі» для демонстрації налаштувань моделі (рис. В.5).

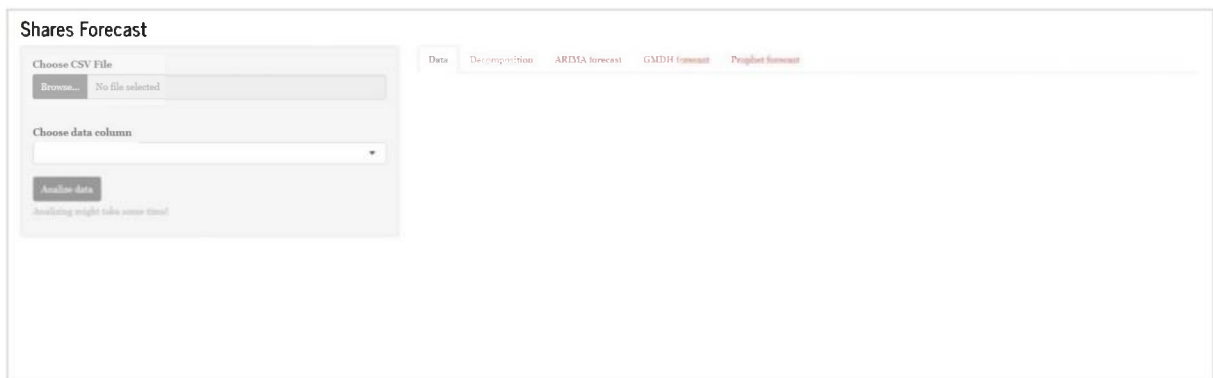


Рис. В.2 – Головна сторінка інформаційної системи

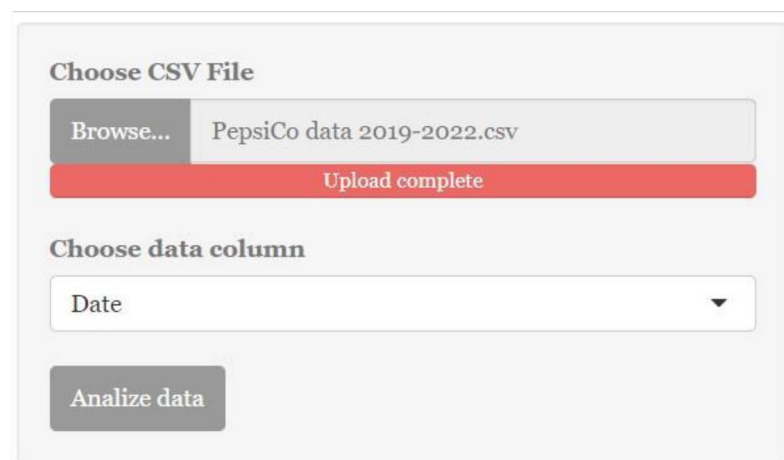


Рис. В.3 – Панель вибору файлу та колонки з даними

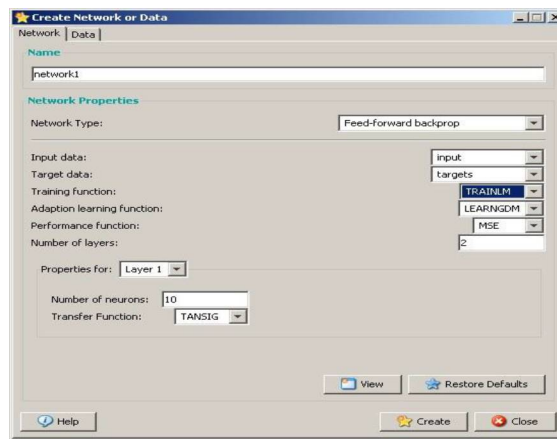


Рис. В.4 – Вікно конфігурації параметрів неймережі

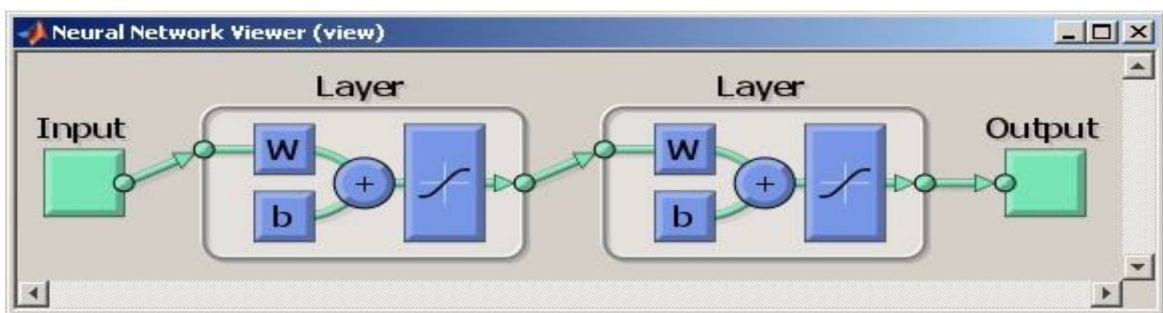


Рис. В.5 – Схематичний вигляд структури розробленої неймережі

### *Тест 3: Валідація та тестування ефективності неймережі*

- Мета тесту: Оцінка точності моделі на валідаційному та тестовому наборах даних.
- Опис процедури: Система проводить навчання неймережі, після чого виконує прогнозування на валідаційному та тестовому наборах даних. Результати візуалізуються у відповідних графіках.
- Використання графіків:
  1. «Графік відображення результатів валідації неймережі» (рис. В.6) та «Графік відображення результатів тестування неймережі на перевіірочній вибірці даних» (рис. В.7) для відображення ефективності моделі.
  2. «Графік залежності середньоквадратичної помилки навченої неймережі від кількості пройдених епох» (рис. В.8) та «Графіки відображення стану тренування неймережі» (рис. В.9) для аналізу процесу навчання.

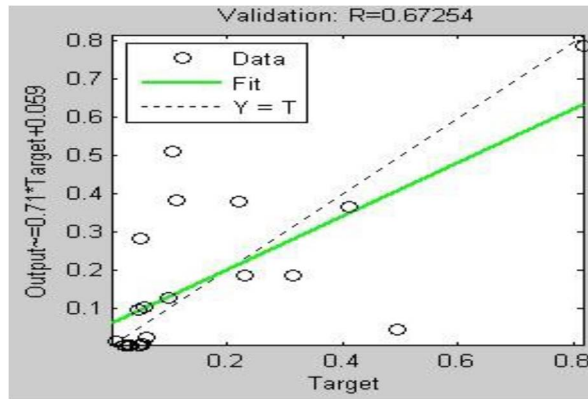


Рис. В.6 – Графік відображення результатів валідації нейромережі

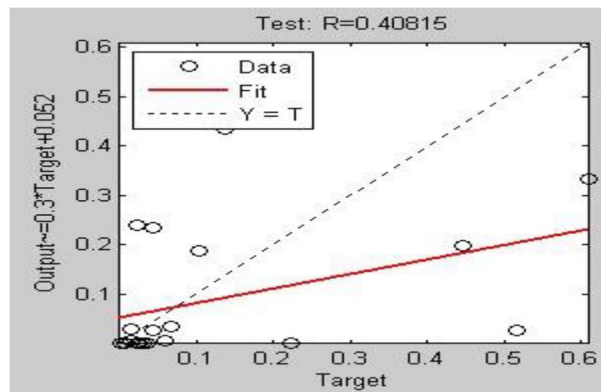


Рис. В.7 – Графік відображення результатів тестування нейромережі на перевіірчній вибірці даних

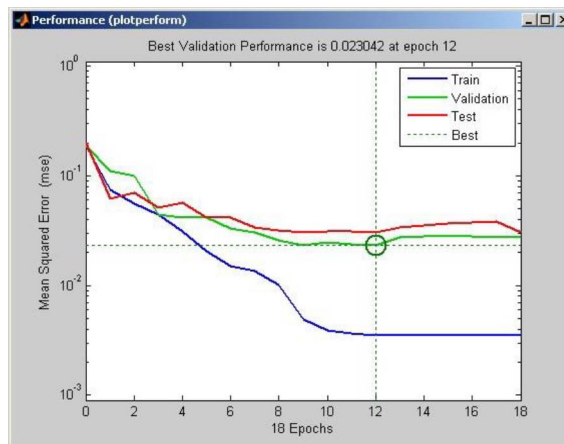


Рис. В.8 – Графік залежності середньоквадратичної помилки навченої нейромережі від кількості пройдених епох

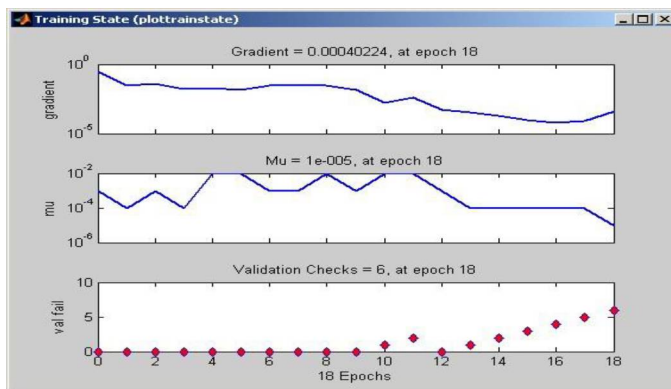


Рис. В.9 – Графіки відображення стану тренування нейромережі

**Висновки:** при вдалому проходженні всіх трьох тестів випробування розробленого додатка вважаються успішними.

**Виконавець:**

студентка групи КУ-41

Кузинська К.Д.