

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА  
ШЕВЧЕНКА  
ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ  
Кафедра економічної кібернетики**



**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «Моделювання вартості акцій компанії Google за допомогою  
економетричних моделей та методів машинного навчання»

**Виконав**

студент 4 курсу

ОКР «Бакалавр»

спеціальності «Економіка»

ОП «Економічна кібернетика»

Шуба Даниїл Дмитрович

**Науковий керівник**

Доктор економічних наук, професор

Ставицький Андрій Володимирович

Роботу допущено до захисту перед ЕК рішенням  
кафедри економічної кібернетики від 12 червня 2025  
р., протокол № 15

Завідувач кафедри: доктор економічних наук,  
професор Ляшенко Олена Ігорівна

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ АНАЛІЗУ КОМПАНІЇ GOOGLE LLC.....	7
1.1. Компанія GOOGLE LLC та її місце на ринку .....	7
1.2. Аналіз основних перспектив розвитку компанії на сучасному етапі	11
1.3. Динаміка капіталізації компанії та роль фондового ринку	15
РОЗДІЛ 2. АНАЛІЗ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГОЮ КЛАСИЧНИХ ЕКОНОМЕТРИЧНИХ МЕТОДІВ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	20
2.1. Економетричні підходи до прогнозування часових рядів .....	20
2.1.1. Методи згладжування.....	20
2.1.2. Методи авторегресії для прогнозування часових рядів.....	24
2.1.3. Регресійні підходи .....	28
2.2. Методи машинного навчання для прогнозування часових рядів.....	31
2.2.1. Методи моделювання вартості цінових рядів за допомогою рекурентних нейронних мереж.....	33
2.2.2. Методи моделювання вартості цінових рядів за допомогою згорткових нейронних мереж.....	37
2.2.3. Методи моделювання вартості цінових рядів за допомогою багатошарового перцептрона Румельхарта.....	40
2.2.4. Методи моделювання вартості цінових рядів за допомогою моделі випадкового лісу.....	43
2.2.5. Методи моделювання вартості цінових рядів за допомогою довготривалої короткострокової пам'яті .....	46

2.2.6. Методи моделювання вартості цінових рядів за допомогою методу рекурсивного усунення ознак ..... 49

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕТОДІВ ЗГЛАДЖУВАННЯ, ЕКОНОМЕТРИЧНИХ МОДЕЛЕЙ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ ..... 52

3.1. Збір та підготовка даних..... 52

3.2. Моделювання вартості акцій компанії GOOGLE за допомогою економетричних методів..... 57

3.3. Моделювання вартості акцій компанії GOOGLE за допомогою методів машинного навчання ..... 65

ВИСНОВКИ..... 74

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 76

ДОДАТОК А. ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ МОДЕЛЕЙ..... 81

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ ДЛЯ РЕАЛІЗАЦІЇ МОДЕЛЕЙ ..... 110

ДОДАТОК В. ЛІСТИНГ ПРОГРАМИ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ЧАСОВОГО РЯДУ ..... 117

## ВСТУП

Фінансові системи окремих країн, разом з іншими секторами економіки, продовжують розвиватися та прогресувати в умовах глобалізації, поширення інформаційних технологій та загальної комп'ютеризації. Цей процес сприяє появі нових фінансових інституцій, інструментів та форм взаємодії між людьми. Виникнення постійної потреби в моніторингу руху грошових одиниць спричиняє сучасні тенденції грошово-валютних систем на світовому ринку, такі як: цифрові інновації та криптовалюти, електронні платіжні системи, фінансова технологічна інновація, розширення міжнародного обміну, тощо.

Однією з найбільш актуальних тем на сьогоднішній день є можливість прогнозування курсу цінних паперів, оскільки це дозволяє відстежувати тенденції розвитку цього сегменту валютних ринків. У сучасних дослідженнях широко використовуються моделі штучних нейронних мереж. Серед вітчизняних вчених слід зазначити наступні роботи: Кравець Т.В. та Ляшенко О.І. прогнозували вартість акцій за допомогою методів машинного навчання [46], Ставицький А.В., Черняк О.І. використовували GARCH підходи для моделювання вартості акцій [47].

Зазначені дослідники працюють над розробкою моделей та методів прогнозування ціни на акції, які базуються на аналізі великого обсягу даних і використанні штучних нейронних мереж. Ці методи можуть допомогти інвесторам, трейдерам та іншим зацікавленим сторонам у розумінні тенденцій ринку акцій та прийнятті обґрунтованих рішень.

Враховуючи швидкі зміни курсу акцій, завжди важливо звертатися до останніх досліджень та робіт вчених, які активно працюють у цій галузі, оскільки нові відкриття та методи постійно з'являються.

У даній роботі проведено порівняльний аналіз трьох підходів до прогнозування: методів згладжування, економетричних моделей та штучних нейронних мереж, що використовують алгоритми штучного інтелекту, з використанням відповідних бібліотек мови програмування Python.

**Об'єкт дослідження:** часові ряди, які містять інформацію щодо щоденного середнього курсу акцій компанії GOOGLE.

**Предмет дослідження:** моделювання середньої щоденної ціни акції компанії GOOGLE за допомогою методів згладжування, економетричних моделей та штучних нейронних мереж, що використовують алгоритми штучного інтелекту.

**Мета дослідження:** порівняти прогностичні можливості досліджуваних методів прогнозування та визначити, який метод є найкращим для моделювання середньої щоденної ціни акції компанії GOOGLE.

**Основні завдання:**

- Дослідження компанії GOOGLE та технології її функціонування.
- Аналіз статистичних даних.
- Дослідження архітектури методів згладжування, економетричних моделей та методів штучних нейронних мереж.
- Визначення найкращого методу для моделювання середньої щоденної ціни акції компанії GOOGLE

Інформаційною та аналітичною базою дослідження методів згладжування, економетричних моделей, методів нейронних мереж для моделювання середньої щоденної ціни акції компанії GOOGLE є наукові праці вітчизняних та зарубіжних вчених, а також Інтернет-ресурси.

**Практична цінність** дипломної роботи полягає у розробці та впровадженні ефективних моделей прогнозування середньої щоденної ціни акцій компанії GOOGLE, що може бути корисним для інвесторів, фінансових аналітиків та трейдерів. Використання методів згладжування, економетричних моделей та нейронних мереж дозволить підвищити точність прогнозів, зменшити ризики та покращити стратегічні рішення в інвестиційному процесі. Отримані результати можуть бути використані для створення програмного забезпечення, що автоматизує процес аналізу та прогнозування ринкових цін.

**Теоретична цінність** дипломної роботи полягає у розширенні наукових знань щодо методів аналізу фінансових даних та їх застосування до прогнозування ринкових цін. У дослідженні будуть проаналізовані існуючі підходи та запропоновані нові моделі, що сприятимуть подальшому розвитку економетрики та фінансових досліджень. Важливим аспектом є також інтеграція методів нейронних мереж у традиційні економетричні моделі, що відкриває нові можливості для покращення прогнозних моделей у фінансовій сфері.

**Структура роботи:** дипломна робота складається зі вступу та трьох розділів. У першому розділі наведено загальні відомості про компанію GOOGLE. У другому розділі наведений математичний опис методів згладжування, економетричних моделей та методів машинного навчання. У третьому розділі наведені результати досліджуваних методів та обрано найкращу модель прогнозування щоденної вартості акції компанії GOOGLE. Також дана робота містить висновки, перелік використаної літератури та додатки з кодами, реалізованими на мові програмування Python і посилання на базу даних.

# РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ АНАЛІЗУ КОМПАНІЇ GOOGLE LLC

## 1.1. Компанія GOOGLE LLC та її місце на ринку

Компанія Google була заснована Ларрі Пейджем та Сергієм Бріном, коли вони були аспірантами Стенфордського університету. Їхньою спільною метою було створення пошукової системи, яка могла б швидко та ефективно знаходити інформацію в Інтернеті. У 1997 році вони створили пошукову систему PageRank, де пошук базується на якісному, а не кількісному алгоритмі, тобто пріоритет надається джерелам з найбільшою кількістю посилань на інших сайтах. У 1998 році компанія була офіційно зареєстрована як Google Inc. Назва Google походить від математичного терміну «googol», що означає число, яке складається з 1 та 100 нулів. Це відображає їхнє бажання зробити доступ до великих обсягів інформації простим і доступним. [4]

Компанія продовжувала стрімко розвиватися: у 1999 році вони купили офісні приміщення і переїхали в Маунтін-В'ю, у 2000 році підписали угоду з Yahoo, великою американською компанією, що спеціалізується на інтернет-сервісах, у 2001 і 2002 роках підкорили Південну Америку і Австралію, а в 2013 році придбали найвідоміший в США видавничий сервіс Blogger. [4]

Одна з найважливіших подій в історії компанії сталася у 2004 році, коли акції надшвидкої та зручної пошукової системи вперше були розміщені на фондовому ринку, що зробило Сергія Бріна та Ларрі Пейджа справжніми мільярдерами. Це мотивувало їх розвивати компанію далі, випускаючи все нові і нові продукти: електронну пошту Gmail, популярний відеосервіс YouTube, карти Google Maps, соціальну мережу Google+, фоторедактор Picasa, онлайн-документи Google Docs, хмарне сховище Google Drive, перекладач Translate та інші. Окрім цих можливостей, в останні роки з'явилася низка Android-пристроїв, які також активно використовують і пропонують користувачам сервіси Google за замовчуванням. [3]

Відкриття офісу Google в Україні у 2010 році продемонструвало стратегічне прагнення компанії до міжнародної експансії та збереження глобальної присутності. 15 серпня 2011 року було оголошено, що Google придбав виробника мобільних

пристроїв Motorola Mobility за 12,5 мільярдів доларів США. Придбання Motorola Mobility у 2011 році дало Google можливість розширити свій вплив в індустрії мобільних технологій і зміцнити конкурентні позиції на ринку смартфонів.

10 серпня 2015 року Сундар Пічаї був оголошений наступним головним виконавчим директором (CEO) Google. Google став найбільшою дочірньою компанією Alphabet. Призначення Сундара Пічаї генеральним директором у 2015 році та перетворення Google на дочірню компанію Alphabet стало стратегічною зміною в управлінні та зміцненні корпоративної структури.

1 серпня 2018 року Google запустив Google Покупки в Україні. Google Shopping - це рекламне рішення, спрямоване на покращення користувацького досвіду, що дозволяє відстежувати покупки, порівнювати різні товари, їхні характеристики та ціни, а потім безпосередньо зв'язуватися з продавцем для здійснення покупки. Рішення працює на базі Merchant Centre - інструменту, який дозволяє завантажувати дані про магазини та продукти в Google і робити їх доступними для реклами інших продуктів і сервісів Google. Запуск Google Shopping у 2018 році демонструє постійне прагнення Google до покращення користувацького досвіду та розширення своїх сервісів. [2]

19 березня 2019 року Google оголосив про вихід на ринок відеоігор, запустивши хмарну ігрову платформу під назвою Google Stadia. Вийшовши на ринок відеоігор з хмарною ігровою платформою Google Stadia у 2019 році, компанія Google продемонструвала свою здатність до інновацій та визнала важливість розвитку галузі. [2]

У грудні 2019 року колишній операційний директор PayPal Білл Рід був призначений новим керівником комерційного відділу Google, який допоможе розвивати Google Pay. Призначення Білла Ріда на посаду керівника відділу комерції у 2019 році демонструє підвищену увагу компанії до розвитку платіжної системи Google Pay. [2]

У січні 2021 року уряд Австралії розробив законопроект, який зобов'язує Google і Facebook платити медіакомпаніям за права на використання їхнього контенту. У

відповідь Google пригрозив закрити доступ до своєї пошукової системи в Австралії. Законопроект австралійського уряду 2021 року та відповідь Google підкреслюють важливість глобальних правових та економічних викликів для компаній.

У вересні 2021 року компанія Google оголосила, що придбала офісну будівлю на Мангеттені за 2,1 мільярда доларів. За даними Real Capital Analytics, купівля Google офісної будівлі в Нью-Йорку є найдорожчою угодою в цьому сегменті з початку пандемії COVID-19 і однією з найдорожчих в історії США [1]. Придбання офісної будівлі на Манхеттені у 2021 році підкреслює стратегічне прагнення Google посилити свою присутність та інфраструктуру в ключових регіонах. [2]

У червні 2023 року Google відмовився від своєї хмарної ігрової платформи під назвою Google Stadia, яка вийшла на ринок відеоігор у 2019 році. Натомість Google почав тестувати онлайн-ігри безпосередньо на YouTube під назвою Playables. Це дозволить користувачам запускати ігри безпосередньо на платформі YouTube, як у веб-версіях, так і в мобільних додатках для Android та iOS. Згідно з повідомленням, до 25 червня 2023 року буде доступна лише одна гра - Stack Bounce. Відмова від хмарної ігрової платформи Google Stadia та запуск онлайн-ігор Playable на YouTube у 2023 році демонструє постійну прихильність Google до інновацій та гнучкість у реагуванні на зміни на ринку відеоігор. [1]

Висока конкурентоспроможність компанії (частка ринку 62,4%) пояснюється сильною корпоративною культурою, ретельним відбором фахівців та створенням найкомфортніших умов праці у світі. Компанія всіляко заохочує своїх співробітників до самостійного та креативного мислення. [5]

Google зараз є однією з найбільших технологічних компаній у світі, враховуючи широкий спектр продуктів і послуг, які вона надає, включаючи операційну систему Android, хмарні сервіси та штучний інтелект. Компанія відома своїми інноваціями, інвестиціями в дослідження і розробки та стратегією диверсифікації бізнесу. [5]

Аналіз заснування Google показує, як вдале поєднання технічного генія та підприємницького мислення може призвести до створення впливової компанії, яка змінює спосіб використання Інтернету та взаємодії з інформацією. Слід також

зазначити, що успіх Google базується не лише на технологічних рішеннях, а й на вмінні реагувати на потреби ринку та ефективному управлінні бізнесом.

Таким чином, Google є справжнім ІТ-гігантом, який сьогодні впливає практично на всі сфери життя суспільства. Без технологій, створених Google, не обходиться ні домашнє використання, ні мережа бізнес-імперії. Завдання, поставлене засновниками на зорі існування пошукової системи: зібрати і класифікувати всю інформацію про все на світі - з кожним днем стає все ближчим і ближчим до реалізації.

## 1.2. Аналіз основних перспектив розвитку компанії на сучасному етапі

Google LLC - це одна з найбільших технологічних компаній у світі, яка займається різноманітними сферами, включаючи пошукові послуги, хмарні технології, штучний інтелект, інтернет-рекламу, електронні товари та послуги та багато іншого. Основні теорії щодо перспектив розвитку компанії Google є наступними [8]:

Інновації в штучному інтелекті (AI): Google активно інвестує у розвиток штучного інтелекту, що відкриває нові можливості для автоматизації процесів, розробки нових продуктів та покращення існуючих послуг. Інвестиції в дослідження та розробку в галузі AI дозволяють Google вдосконалювати свої алгоритми та створювати нові, більш потужні інтелектуальні продукти. Інвестиції у розвиток штучного інтелекту (AI) мають значний вплив на вартість компанії, її конкурентоспроможність і перспективи цін. Розглянемо це в контексті Google та порівняємо з іншими компаніями. Ріст доходів і зниження витрат завдяки інноваціям в галузі штучного інтелекту можуть позитивно позначитися на фінансових показниках компанії. Крім того, інвестиції в AI допомагають зайняти стратегічну позицію на ринку та залишитися інноваційним лідером. Успішна реалізація AI-проектів може призвести до зростання вартості акцій компанії та збільшити інвестиційний інтерес до неї. Проте Google має сильну конкуренцію від інших технологічних гігантів, таких як Amazon, Microsoft, та Apple, які також активно інвестують у розвиток AI. Крім того, на ринку присутні багато стартапів, що спеціалізуються на розробці AI-технологій, які можуть стати потенційними конкурентами для Google у майбутньому. [6]

Розширення продуктового портфеля: Крім пошукової системи, Google розробляє здобуває нові продукти та сервіси, такі як Google Cloud Platform, YouTube, Android, Google Maps, Google Drive тощо. Розширення цього портфеля може допомогти компанії розширити свою аудиторію та збільшити прибуток. Розширення портфелю дозволяє Google більше конкурувати на ринку і забезпечує компанії додаткові джерела доходів та диверсифікацію бізнесу. Розширення портфеля дозволяє Google

розширити свою аудиторію та збільшити прибуток, що може позитивно вплинути на вартість компанії. Різноманітність продуктів дозволяє зменшити ризики, пов'язані з кон'юнктурними коливаннями в окремих секторах, що може зробити Google менш вразливою до змін на ринку та сприяти стабільнішій вартості компанії. Збільшення продуктового портфеля може збільшити інвестиційний інтерес до Google, що може призвести до зростання цін акцій компанії. Додавання нових продуктів може вплинути на ринкову оцінку компанії, особливо якщо ці продукти успішно використовуються та мають потенціал для майбутнього зростання.

Впровадження в галузі здоров'я та біотехнологій: Google активно досліджує можливості застосування технологій для поліпшення сфери охорони здоров'я та біотехнологій. Це може включати розробку медичного обладнання, програмного забезпечення для аналізу медичних даних, телемедицинські послуги тощо. Розробка медичного обладнання, програмного забезпечення для аналізу медичних даних та телемедицинських послуг може відкрити нові ринки та створити додаткові джерела доходу для Google. Активна участь у впровадженні технологій у сфері охорони здоров'я може підвищити імідж компанії та її репутацію як соціально відповідальної. Успішна реалізація ініціативи у галузі здоров'я та біотехнологій може збільшити інвестиційний інтерес до компанії, що може призвести до зростання цін акцій. Розширення діяльності в галузі охорони здоров'я може призвести до оцінки ринкових аналітиків щодо потенційних можливостей росту компанії.

Розвиток автономних систем: Google активно працює над розробкою автономних систем, таких як безпілотні автомобілі через свою дочірню компанію Waymo. Ця галузь може мати великий вплив на майбутнє транспортних послуг та мобільності. Впровадження автономних транспортних засобів може призвести до зменшення аварій, поліпшення транспортної ефективності та зменшення екологічного впливу. Розвиток автономних систем відкриває нові ринки для Google, що може призвести до збільшення прибутку компанії та позитивно вплинути на її вартість. Успішна реалізація проектів у сфері автономних систем може зміцнити позиції Google як технологічного лідера, що відображається на оцінці ринку та

інвестиційному інтересі. Якщо Google успішно впровадить автономні системи, це може призвести до зростання ціни її акцій, оскільки інвестори оцінять потенційні можливості у цьому секторі. Реалізація автономних систем може призвести до перегляду ринкових прогнозів та оцінок компанії, що може вплинути на її вартість на ринку капіталу.

Інші технологічні гіганти, такі як Tesla, Uber, та інші компанії, також активно працюють у сфері автономних систем, створюючи конкурентну середовище. Несподівані технічні або правові проблеми можуть затримати розвиток автономних систем, що може вплинути на конкурентну позицію Google у цій галузі. [6]

У порівнянні з іншими компаніями, Google має значні ресурси, технічні знання та досвід, які можуть допомогти їй зберегти конкурентну перевагу у цьому секторі. Однак успіх впровадження автономних систем залежатиме від ефективності виконання проєктів та вирішення ризиків і викликів, пов'язаних з цією технологією.

Збільшення захоплення даних та рекламна ефективність: Google продовжує розвивати свої алгоритми та технології збору та аналізу даних, щоб покращити ефективність своєї рекламної платформи. Це включає в себе розвиток технологій машинного навчання для кращого співставлення реклами з інтересами користувачів та збільшення конверсії. Покращена ефективність рекламної платформи може привести до збільшення прибутку Google, оскільки рекламодавці будуть готові платити більше за ефективні рекламні кампанії. Покращення ефективності рекламних кампаній може зробити Google більш привабливою для інвесторів, що може вплинути на його вартість на ринку капіталу. Успішне вдосконалення рекламної платформи може призвести до зростання ціни акцій Google, оскільки інвестори оцінять потенційний зріст прибутку компанії. Ринок може реагувати позитивно на нові технології та підходи до рекламної ефективності, що може вплинути на ринкову оцінку Google. [7]

Екологічна ініціатива: За останні роки Google зосереджує увагу на екологічних ініціативах, таких як зменшення викидів CO<sub>2</sub>, використання відновлюваних джерел енергії для своїх дата-центрів тощо. Ця тенденція може продовжитися в майбутньому,

оскільки компанії все більше звертають увагу на свої екологічні впливи. Ця ініціатива спрямована на підтримку сталого розвитку та покращення екологічного статусу компанії. Сприяння екологічним ініціативам може позитивно вплинути на сприйняття компанії ринком та споживачами, що може підвищити її вартість. Використання відновлюваних джерел енергії та оптимізація енергоефективності може допомогти знизити витрати на енергію для компанії в майбутньому. Інвестори можуть реагувати позитивно на екологічні ініціативи, що може підвищити ціну акцій компанії. Зменшення екологічного впливу може позитивно вплинути на загальну ринкову оцінку компанії, оскільки інвестори та споживачі все більше звертають увагу на екологічну відповідальність підприємств. [7]

У порівнянні з іншими компаніями, Google має перевагу у великому обсязі інноваційних продуктів та широкому колі аудиторії. Однак конкурентна боротьба залишається високою, і успіх Google залежатиме від ефективності впровадження нових продуктів та здатності конкурувати на ринку. [8]

### 1.3. Динаміка капіталізації компанії та роль фондового ринку

Капіталізація компанії є одним з головних показників, що відображає її становище на фондовому ринку. Вона являє собою загальну ринкову вартість усіх випущених акцій компанії, помножену на їхню кількість. Цей показник важливий не лише для оцінки фінансового стану компанії, але й для інвесторів, які використовують його для прийняття рішень щодо вкладення в цінні папери. Для таких великих компаній, як Google, динаміка капіталізації є ключовим фактором. Зростання капіталізації свідчить про успіх компанії у залученні інвесторів і підтримці їхньої довіри, що може бути результатом високих фінансових показників, ефективного управління та інноваційних продуктів чи послуг. Крім того, стабільність капіталізації демонструє надійність компанії на фондовому ринку. Коливання цього показника може вплинути на інвесторів, змінюючи їхні інвестиційні рішення. Тому компанії прагнуть досягти стабільності капіталізації, щоб зберігати довіру ринку та забезпечувати стабільність свого фінансового становища.

У випадку Google, що належить до найбільших та найвпливовіших компаній світу, динаміка її капіталізації відображає її успіхи на фондовому ринку та довіру інвесторів до її бізнес-стратегій. На рис. 1 видно, що за останнє десятиліття Google продемонструвала значний ріст капіталізації, збільшивши вартість своїх акцій на 1294% [9]. Цей досягнутий успіх вважається одним з найбільш вражаючих в історії фондового ринку. Позитивну динаміку Google на фондовому ринку можна пояснити її стратегічними ініціативами та інноваційним підходом до ведення бізнесу. Компанія активно інвестує у дослідження та розробку нових технологій, таких як штучний інтелект, машинне навчання, хмарні обчислення та інші. Завдяки цьому Google випереджає конкурентів і забезпечує стабільне зростання своєї капіталізації. Додатково, компанія постійно розширює та вдосконалює свої продукти та послуги, такі як пошукова система Google, онлайн-реклама, хмарні сервіси, мобільні додатки тощо. Диверсифікація портфеля продуктів та послуг допомагає Google залишатися конкурентоспроможною і приваблювати нових інвесторів. Крім того, Google активно

використовує свої фінансові ресурси для стратегічних придбань та інвестування в інші компанії. Це дозволяє розширити свою присутність на ринку і входити в нові бізнес-сегменти, що сприяє подальшому зростанню капіталізації [10].

Google має сильні фундаментальні показники, такі як стійкий ріст доходів та прибутків. Згідно з даними, річний дохід компанії склав 65 мільярдів доларів, що ставить її на одному з перших місць серед американських компаній. Цей успіх відображається у високій маржинальності Google, що забезпечує стабільний ріст її капіталізації. При цьому, ця цифра ставить Google на одне з провідних місць серед американських компаній за обсягом доходів. Важливо відзначити, що постійний ріст доходів свідчить про ефективність бізнес-моделі компанії та її здатність адаптуватися до змін на ринку. Однак, така висока маржинальність Google, яка забезпечує стабільний ріст її капіталізації, є наслідком не лише обсягу доходів, але й ефективного управління витратами. Компанія постійно працює над оптимізацією своєї внутрішньої структури та бізнес-процесів, що дозволяє збільшувати маржинальність при збереженні високого рівня якості продуктів та послуг. Такий успіх Google на фондовому ринку може також бути пояснений його здатністю до інновацій та розвитку нових продуктів та послуг. Компанія постійно вкладає значні ресурси у дослідження та розвиток нових технологій, таких як штучний інтелект, машинне навчання, розширена реальність та інші, що дозволяє їй залишатися на передовій позиції у своїй галузі та приваблювати інвесторів. [10]

Розділення акцій - це стратегічний крок, який часто вживають компанії для збільшення доступності своїх акцій для широкого кола інвесторів та збільшення ліквідності на ринку. Ділити акції зазвичай зручно, коли їхні ціни стають дуже високими, що може змушувати багатьох інвесторів вагатися перед купівлею. Розділення акцій знижує ціну одиниці акції, зариблюючи їх більш доступними для інвестування. У випадку Google, розділення акцій відображає стратегічну вірність компанії до розширення свого акціонерного базису та створення більш привабливих умов для інвесторів. Крім того, це може також слугувати індикатором впевненості компанії у її майбутньому розвитку і зростанні, оскільки розділення акцій зазвичай

проводиться тоді, коли компанія очікує подальшого росту. З позиції інвесторів, розділення акцій може викликати позитивний настрій, оскільки це може вказувати на те, що компанія перебуває на шляху успіху та має намір продовжувати рости. Це може підвищити попит на акції компанії та підтримати позитивну динаміку цін. Google оголосила про створення нового класу капіталу без права голосу, або класу C акцій. Google розподілила акції свого класу C у формі дивіденду акціонерам із датою ведення списку дивідендів 27 березня 2014 року та датою виплати дивіденду 2 квітня 2014 року. Торгівля новими акціями класу C розпочалася 3 квітня 2014 року, датою відміни дивідендів. Таким чином, розділення акцій Google є одним із стратегічних кроків компанії, спрямованих на підвищення привабливості своїх акцій для інвесторів та стимулювання позитивного настрою на ринку. Google активно використовує нові технології та інновації для розвитку свого бізнесу, що підсилює інтерес інвесторів. Висока конкурентоспроможність та лідерство у сфері технологій дозволяють компанії зберігати свою позицію як одного з лідерів на ринку. [11]

Включення Google до індексу S&P 500 є значущою подією для будь-якої компанії, оскільки цей індекс вважається одним з найбільших і найвпливовіших показників стану американського фондового ринку [12]. Індекс S&P 500 складається з 500 найбільших компаній за ринковою капіталізацією, які торгуються на американських фондових біржах, і він є важливим барометром для інвесторів і фахівців у фінансовій галузі. Для Google включення до індексу S&P 500 означало визнання його статусу як однієї з найбільших і найбільш впливових компаній у світі. Це подія, яка підтвердила довіру інвесторів та інституційних фондів до бізнесу Google та його перспектив на майбутнє. Включення до індексу S&P 500 відкрило нові можливості для інституційних інвесторів, які, керуючись стратегіями інвестування, можуть використовувати індекс як основний джерело для формування своїх портфелів. Це сприяло зростанню попиту на акції Google серед інституційних інвесторів, що призвело до додаткового зростання капіталізації компанії. [12]

З позиції інвесторів, включення Google до індексу S&P 500 створило додаткові можливості для різноманітних стратегій інвестування, зокрема індексного

інвестування або портфельного управління. На рис.1, можна побачити, що це сприяло збільшенню притоку капіталу до компанії та підтримці позитивного настрою на ринку. [13]

Отже, включення Google до індексу S&P 500 в 2006 році стало важливим кроком, який підвищив привабливість компанії для інституційних інвесторів, що в свою чергу сприяло додатковому зростанню її капіталізації. Історія успіху Google на фондовому ринку свідчить про сильність компанії та довіру інвесторів до її майбутнього. З річною капіталізацією у 397 мільярдів доларів, Google продовжує зберігати свою позицію як одна з найбільших та найбільш впливових компаній у світі, ставши однією з ікон сучасного фінансового ринку.

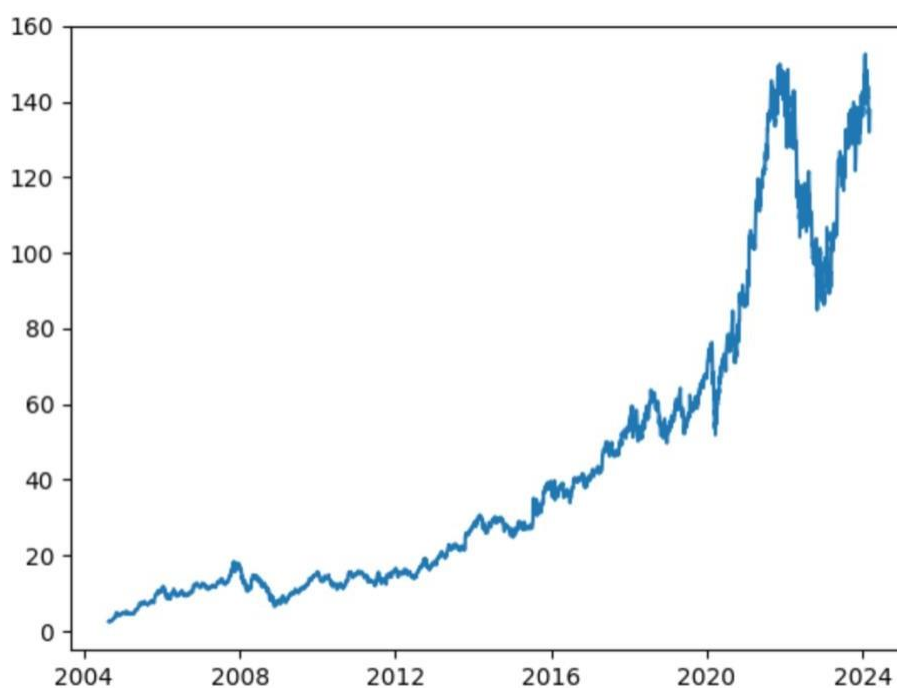


Рис. 1.3.1 Динаміка зміни вартості акцій компанії GOOGLE LLC

Джерело: складено автором

## Висновки до розділу 1

За останні роки акції компанії Google продемонстрували значний ріст, що відображає успішний розвиток бізнесу та довіру ринку до стратегії компанії. Однак, вартість акцій Google піддається впливу різноманітних факторів, що варіюються від фундаментальних показників до зовнішніх економічних та політичних подій.

Перш за все, фундаментальні фактори, такі як фінансові показники компанії, її прибутковість, інноваційність та стратегія розвитку, впливають на вартість акцій Google. Крім того, ринкові тенденції, включаючи загальну економічну ситуацію, кон'юнктуру галузей, конкуренцію та рішення регуляторів, також мають суттєвий вплив.

У зв'язку з такою складною динамікою ринку, для прогнозування вартості акцій Google необхідно розглядати різні моделі, які охоплюють широкий спектр аспектів. Економетричні моделі дозволяють аналізувати фундаментальні показники та їх вплив на ціну акцій. Моделі авторегресії допомагають передбачати майбутні тенденції на основі минулих даних. Згладжування може бути корисним для виділення основних трендів та зменшення шуму в даних. Нарешті, методи машинного навчання можуть допомогти виявити складні неявні зв'язки між різними факторами та ціновими змінами акцій, що можуть бути важкі для виявлення за допомогою традиційних методів.

Отже, успіх компанії Google на фондовому ринку визначається не лише її власними досягненнями та стратегією, але й широким спектром факторів, що впливають на динаміку ринку. Для прогнозування вартості акцій Google необхідно враховувати різноманітні моделі та фактори, що вимагає глибокого аналізу та дослідження.

## **РОЗДІЛ 2. АНАЛІЗ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГОЮ КЛАСИЧНИХ ЕКОНОМЕТРИЧНИХ МЕТОДІВ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ**

### **2.1. Економетричні підходи до прогнозування часових рядів**

Прогнозування часових рядів - це процес аналізу часових рядів даних за допомогою статистики та моделювання для здійснення передбачень та підтримки стратегічного прийняття рішень. Це не завжди точний прогноз, ймовірність точності прогнозів може широко змінюватися, особливо коли маємо справу з постійно змінюваними змінними у часових рядах та факторами, над якими ми не маємо контролю. Проте прогнозування дає уявлення про те, які результати більш ймовірні або менш ймовірні, ніж інші можливі результати. Зазвичай, чим більше у нас повна інформація, тим точніші можуть бути прогнози. Хоча терміни "прогнозування" та "прогноз" загалом мають однакове значення, є помітна відмінність. У деяких галузях прогнозування може стосуватися даних на конкретний майбутній момент часу, тоді як прогнозування відноситься до майбутніх даних загалом. Прогнозування часових рядів часто використовується в поєднанні з аналізом часових рядів. Аналіз часових рядів передбачає розробку моделей для отримання розуміння даних та виявлення причин. Прогнозування потім переходить до наступного кроку - використання цих знань і передбачуваних екстраполяцій того, що може трапитися в майбутньому. [15]

#### **2.1.1. Методи згладжування**

Згладжування є методом прогнозування фінансових ринків, який має велике значення та широке застосування. Зазвичай використовуються різні методи згладжування, зокрема ковзні середні. Вони допомагають знизити вплив випадкових факторів у часовому ряді. В даному розділі будуть представлені 4 звичайних методи згладжування, які використовуються для прогнозування фінансових часових рядів.

Основне припущення цих методів полягає в тому, що коливання попередніх значень є випадковими відхиленнями від деякої плавної кривої, яку можна використовувати для екстраполяції та створення прогнозу. [18]

Один з видів змінного середнього, що використовується, - експоненціальне ковзне середнє (ЕМА). Цей метод можна розглядати як зважене ковзне середнє, де кожен компонент часового ряду має свою вагу, яка зменшується залежно від віддаленості від поточного торгового періоду. Таке розподілення дозволяє зосередитися на поточних цінах і не пропустити важливі торговельні сигнали. Експоненціальне ковзне середнє зменшує затримку, приділяючи більше уваги останнім цінам порівняно з віддаленими цінами. Це дозволяє швидше реагувати на поточні зміни цін, порівняно з простим ковзним середнім. Вага, яку отримує остання ціна, залежить від періоду експоненціального ковзного середнього. З коротшим періодом ЕМА, більша вага надається останній ціні. [18]

Математична формула для розрахунку ЕМА є рекурсивною і при використанні коефіцієнта згладжування, що дорівнює  $n$ , має наступний вигляд:

$$EMA_t = \alpha \cdot p_t + (1 - \alpha)EMA_{t-1}$$

де  $EMA_t$  - значення експоненціального змінного середнього в точці  $t$  (останнє значення, в разі тимчасового ряду),

$EMA_{t-1}$  - значення експоненціального змінного середнього в точці  $t - 1$  (попереднє значення в разі тимчасового ряду),

$p_t$  - значення вихідної функції в момент часу  $t$  (останнє значення, в разі тимчасового ряду),

$\alpha$  (згладжуючи константа від англ. *smoothing constant*) - коефіцієнт що характеризує швидкість зменшення ваг.

Перше значення експоненціального змінного середнього, зазвичай приймається рівним першому значенню вихідної функції:

$$EMA_0 = p_0$$

Коефіцієнт, може бути обраний довільним чином, в межах від 0 до 1, наприклад, виражений через величину вікна усереднення:

$$\alpha = \frac{2}{(n + 1)}$$

У звичайному експоненціальному ковзному середньому згладжування піддаються значення вихідної функції, однак, згладжування можуть піддаватися і значення результуючої функції. Тому деякі автори надають поняття експоненціальні ковзне середнє довільного порядку, які обраховуються за формулою:

$$EMA_t^{(n)} = EMA_t^{(n-1)} + (1 - \alpha) \cdot EMA_{t-1}^{(n)}$$

де  $EMA_t^{(n)}$ - значення експоненціального змінного середнього n-го порядку в точці t (останнє значення, в разі тимчасового ряду),  $EMA_{t-1}^{(n)}$  - значення експоненціального змінного середнього n-го значення в точці t – 1(попереднє значення в разі часового ряду),  $EMA_t^{(n-1)}$  - значення експоненціального змінного середнього (n – 1) -го значення в точці t (останнє значення, в разі тимчасового ряду),  $\alpha$  - згладжує константа. [20]

В даній роботі досліджувалося подвійне (метод Холта-Вінтерса) та потрійне (метод Брауна) експоненціальне ковзне середнє.

Експоненціально зважені ковзні середні другого і третього порядку позначають іноді як, відповідно DMA (від англ. Double exponential moving average - подвійне (дворазове) експоненціальне ковзне середнє) і TMA (від англ. triple exponential moving average - потрійне (триразове)) експоненціальне ковзне середнє:

$$DMA_t = EMA_t + (1 - \alpha)DMA_{t-1}$$

$$TMA_t = DMA_t + (1 - \alpha)TMA_{t-1}$$

**Метод ковзного середнього** використовується для позначення набору значень, які постійно змінюються в часі. Ковзне середнє відображає загальну тенденцію зміни цін і приглушує незначні коливання. Оскільки ковзне середнє представляє середнє значення цін в минулому, графіки ковзних середніх зазвичай "відстають" від поточних змін у часовому ряді. На ринках з вираженим трендом ковзні середні дають добрі результати, але на ринках без вираженої тенденції цей метод може призводити до значних похибок. Формально цей метод описується так: [14]

$$y_t = \frac{1}{N} \sum_{i=1}^N y_{k-i},$$

Де  $N$  – число попередніх моментів, що було взято до уваги при побудові прогнозу,  $y_{k-i}$  – реальні значення показника в момент часу  $t_{k-i}$ .

**Метод зваженого ковзного середнього** розширює концепцію простого ковзного середнього шляхом призначення вагових коефіцієнтів для кожного компонента часового ряду. Це означає, що значення кожного компонента осереднюється залежно від його ваги: [14]

$$y_t = \frac{1}{N} \sum_{i=1}^N w_i y_{k-i},$$

де  $N$  – число попередніх моментів, що було взято до уваги при побудові прогнозу,  $y_{k-i}$  – реальні значення показника в момент часу  $t_{k-i}$ ,  $w_i$  – ваговий коефіцієнт для  $i$ -го компоненту ряду.

Даний метод краще використовувати, оскільки в результаті отримана оцінка тренду є набагато гладкою. Кожне вхідне спостереження різко не змінює своє середнє значення за допомогою вагових коефіцієнтів.

### 2.1.2. Методи авторегресії для прогнозування часових рядів

Методи авторегресії є статистичними моделями, які використовуються для аналізу та прогнозування часових рядів. Основна ідея за методами авторегресії полягає в тому, що значення змінної у поточний момент часу залежить від попередніх значень цієї змінної.

Одна з основних моделей авторегресії - авторегресія AR(p), де "p" вказує на кількість попередніх значень, що використовуються для прогнозу. У моделі AR(p) кожне значення в часовому ряді представляється як лінійна комбінація попередніх значень цього ряду.

Позначення AR (p) відноситься до авторегресійної моделі порядку p. Модель AR (p) записують як:

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t$$

де  $\varphi_1, \dots, \varphi_p$  – параметри авторегресії,  $c$  – постійна, а випадкова величина  $\varepsilon_t$  – білий шум. Деякі обмеження необхідні на значеннях параметрів, так що модель залишається стаціонарною. Наприклад, процеси в моделі AR(1) з  $|\varphi_1| \geq 1$  не є стаціонарними. У статистиці та обробці сигналів модель авторегресії (AR) є поданням типу випадкового процесу; як таке, воно описує певні зміни в часі, процеси в природі, економіці тощо. [17]

Додатково, існує розширена модель авторегресії ARMA(p, q), яка комбінує авторегресію з рухомим середнім. В цій моделі крім попередніх значень часового ряду також враховується вплив попередніх значень помилок моделі.

Модель ARMA - це просто злиття між моделями AR(p) і MA(q):

Моделі AR(p) намагаються пояснити ефекти імпульсу та середнього реверсії, які часто спостерігаються в стохастичних процесах.

Моделі MA(q) намагаються захопити ударні ефекти, що спостерігаються в умовах білого шуму. Ці шокові ефекти можна розглядати як несподівані події, що впливають на процес спостереження[17].

Модель ARMA намагається охопити обидва ці аспекти при моделюванні часових рядів. [20]

Часто цю модель називають моделлю ARMA(p, q), де:

- p - порядок полінома авторегресії,
- q - порядок багаточлена ковзної середньої.

$$X_t = c + \varepsilon_t + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

де  $\varphi$  - параметри моделі авторегресії,

$\theta$  - параметри моделі ковзної середньої.

c - константа,

$\varepsilon$  - терміни помилок (білий шум).

Алгоритм побудови моделі ARMA(p, q) складається з наступних кроків:

1. Визначаємо порядок авторегресійної складової p
2. Обчислюємо ACF та PACF
3. По значенням ACF та PACF визначаємо p
4. Обчислюємо AR(p)
5. Визначаємо порядок ковзного середнього MA – q
6. Знаходимо ACF та PACF залишків моделі AR(p)
7. По отриманим значенням знаходимо q порядок MA

Більш складним варіантом є модель авторегресії з інтегрованим рухом ARIMA(p, d, q), де "d" вказує на кількість похідних, які потрібно взяти, щоб зробити ряд стаціонарним. Ця модель включає в себе як авторегресійну, так і рухому середню складові, а також може враховувати інтегровану складову для корекції тренду.

Різниця між ARMA-моделлю і ARIMA. Ці дві моделі мають багато спільного. Насправді компоненти AR і MA ідентичні, об'єднуючи загальну авторегресійну модель AR(p) і загальну ковзну середню модель MA(q). AR(p) робить прогнози з використанням попередніх значень залежної змінної. MA (q) робить прогнози з використанням середніх і попередніх помилок. [19]

Модель ARMA є стаціонарною моделлю; Якщо ваша модель не є стаціонарною, то можна досягти стаціонарності, взявши ряд відмінностей. "I" в моделі ARIMA означає інтегрований; Це міра того, як багато несезонних відмінностей потрібні для досягнення стаціонарності. Якщо ніякої диференціації не бере участь в моделі, то вона стає просто ARMA.

В основі методології, запропонованої Боксом і Дженкінсом, лежить ідея про те, що поведінка змінної в теперішньому і майбутньому часі з достатнім ступенем точності пояснюється її поведінкою в минулому.

Модель авторегресії з ковзним середнім ARMA є методологічний апарат для ідентифікації, оцінки та діагностики моделей стаціонарних часових рядів.

Подання про послідовні вигляді моделі змінного середнього MA(q) і авторегресії AR(p) в комбінації з послідовним взяттям різниць d дає можливість моделювати нестаціонарні процеси, зводячи їх до стаціонарних. Модель авторегресії з проінтегровати змінного середнім для нестаціонарних часових рядів отримала назву ARIMA.

Короткий запис моделі ARIMA має вигляд [19]:

$$\varphi(B)(1 - B)^d X_t = \theta(B)\varepsilon_t,$$

де  $\varphi$  - поліноми ступенів p, q, B - лаговий оператор, d - порядок взяття послідовної різниці.

Завдання моделювання процесу за допомогою ARIMA зводиться до знаходження оптимального значення параметрів p, d, q, причому ці значення повинні бути мінімально можливими щоб уникнути надмірності моделі.

Алгоритм побудови моделі ARIMA складається з наступних кроків:

1. Ідентифікація моделі - процес вибору моделі, яка найкращим чином відповідає досліджуваного тимчасового ряду.
2. Оцінювання моделі - використання різних методів для отримання оцінок параметрів, включених в модель.
3. Тестування моделі, зокрема, перевірка залишків на нормальність і незалежність розподілу.
4. Виконання прогнозу за допомогою отриманої моделі.

Відповідні значення  $p$ ,  $d$  та  $q$  можуть бути знайдені за допомогою побудови часткових автокореляційних функцій для оцінки  $p$ , а також з використанням автокореляційних функцій для оцінки  $q$ . Для цього використовують інформаційний критерій Акайке (Akaike information criterion - AIC). При оцінці кількості інформації, втраченої моделлю, цей критерій шукає компроміс між точністю та простотою моделі, тому моделі, які краще підходять при використанні меншої кількості функцій, отримують кращу (нижчу) оцінку AIC, ніж аналогічні моделі, які використовують більше можливостей. [20]

В загальному критерій Акайке має такий вигляд:

$$AIC = 2k - 2 \ln L,$$

де  $k$  – кількість оцінених параметрів у моделі,

$L$  – максимальне значення функції правдоподібності для моделі.

Модель із мінімальним значенням AIC вважається найкращою. Таким чином, AIC сприяє точності моделі (оцінюючи функцію правдоподібності), але також “штрафує” за збільшення кількості оцінюваних параметрів. Такий штраф перешкоджає перенавчанню моделі, оскільки збільшення кількості параметрів у моделі майже завжди покращує точність відтворення моделі на тестових даних. У сучасному середовищі для аналізу та прогнозування, цей критерій реалізовано у відповідних бібліотеках та функціях.

Методи авторегресії дозволяють аналізувати залежності та прогнозувати майбутні значення часових рядів на основі їх попередніх значень. Вони знаходять

широке застосування у фінансовому аналізі, економічному прогнозуванні, прогнозуванні попиту та інших галузях, де важливо виявити та спрогнозувати патерни та тенденції в часових рядах.

### 2.1.3. Регресійні підходи

Основна концепція полягає в тому, що прогноз значення відбувається за допомогою припущення, що він має лінійну залежність з іншими часовими рядами.

Наприклад, прогнозування місячних продажів може відбуватися з використанням загальних рекламних витрат як передбачувальний фактор. Або можна прогнозувати денний попит на електроенергію, використовуючи температуру і день тижня як передбачувальні фактори. [17]

Прогнозна змінна іноді також називається регресандом, залежною або пояснювальною змінною. Передбачувальні змінні іноді також називають регресорами, незалежними або пояснювальними змінними. У цій книзі ми завжди будемо посилатися на них як «прогнозу» змінну і «передбачувальні» змінні.

**Модель простої лінійної регресії** - це простий підхід до прогнозування цінкових рядів, де ми розглядаємо лише одну незалежну змінну (предиктор) для прогнозу цінкових змін. Загальний вигляд даної моделі має наступний вигляд:

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

де  $\beta_0$  – точка перетину регресії з віссю  $y$ , коли значення  $x$  дорівнює 0,

$\beta_1$  – коефіцієнт регресії, який відображає наскільки змінюється значення  $y$  при зміні одиниці значення  $x$

$\varepsilon$  – помилка, яка відображає випадкову неточність моделі.

Мета полягає в тому, щоб знайти оптимальні значення  $\beta_0$  та  $\beta_1$  які найкраще відображають взаємозв'язок між  $x$  та  $y$ . Це може бути зроблено шляхом мінімізації суми квадратів помилок між спостереженими значеннями  $y$  та значеннями, передбаченими моделлю.

Отже, модель простої лінійної регресії допомагає прогнозувати цінові ряди, використовуючи лише один предиктор, що дає змогу робити прості та ефективні прогнози. [16]

**Модель множинної лінійної регресії** є статистичним методом, який використовується для аналізу залежності між однією залежною змінною і двома або більше незалежними змінними. В основі цієї моделі лежить припущення, що залежна змінна може бути виражена як лінійна комбінація незалежних змінних з додаванням помилки. Ця помилка враховується, оскільки реальні дані часто містять випадкові відхилення. Модель виражається наступною формулою:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon,$$

де  $Y$  - залежна змінна, яку ми намагаємось передбачити,

$X_1, X_2, \dots, X_n$  - незалежні змінні, які використовуються для передбачення змінної  $Y$

$\beta_0, \beta_1, \beta_2, \dots, \beta_n$  - коефіцієнти регресії, які визначають вагу кожної незалежної змінної,

$\varepsilon$  – помилка моделі, яка враховує випадкові відхилення.

Мета множинної лінійної регресії полягає у встановленні зв'язку між незалежними і залежною змінними, а також у прогнозуванні значень залежної змінної на основі значень незалежних змінних. Для цього кожен коефіцієнт регресії оцінюється за допомогою статистичних методів, таких як метод найменших квадратів, з метою мінімізації суми квадратів відхилень між фактичними і передбаченими значеннями залежної змінної.

**Методи нелінійної регресії** використовуються, коли залежність між змінними не може бути адекватно виражена лінійною моделлю. Вони застосовуються для аналізу та передбачення залежностей, які можуть бути більш складними або не мають прямого лінійного зв'язку між змінними.

Найбільш поширеною трансформацією є метод натурального логарифма. Даний метод можна записати наступною формулою:

$$y = \beta_0 + \beta_1 \ln(x) + \varepsilon$$

Одним із методів нелінійної регресії є метод множинної регресії з використанням перетворення логарифмів. Цей метод полягає в тому, щоб замінити оригінальні змінні (зазвичай прогножуючу  $x$  та/або прогнозовану  $y$ ) на їх логарифми, а потім застосувати лінійну регресію до цих перетворених змінних. Логарифмічне перетворення дозволяє уникнути нелінійності у вихідних даних, тим самим роблячи модель більш адекватною для аналізу складних нелінійних залежностей між змінними.

Метод нелінійної регресії "piecewise linear" (частково лінійна) є способом моделювання нелінійних залежностей, де використовується кілька лінійних сегментів для апроксимації складної функції. Замість того, щоб використовувати одну лінію для всього діапазону даних, діапазон даних розбивається на кілька підінтервалів, і на кожному підінтервалі застосовується окрема лінійна функція.

Основна ідея полягає в тому, щоб підібрати лінійну функцію для кожного підінтервалу так, щоб вона найкращим чином відповідала даним у цьому конкретному діапазоні. Це дозволяє адаптувати модель до нелінійних залежностей, які можуть бути складними та неоднорідними по всьому діапазону.

Процес моделювання часових рядів за допомогою методу нелінійної регресії включає наступні кроки:

- Вибір точок розбиття: Діапазон даних розбивається на підінтервали за допомогою точок розбиття. Ці точки можуть бути вибрані на основі експертного досвіду, або можуть бути визначені автоматично на основі аналізу даних.
- Підгонка лінійних моделей: Для кожного підінтервалу виконується підгонка лінійної моделі, такої як проста лінійна регресія. Це означає, що кожен підінтервал має свої власні коефіцієнти регресії.
- Оцінка моделі: Після підгонки всіх лінійних моделей проводиться оцінка їхньої адекватності та якості підгонки до даних. Це може включати аналіз залишкових діаграм, оцінку критеріїв адекватності моделі тощо.
- Інтерпретація результатів: Після побудови "piecewise linear" моделі важливо інтерпретувати кожен знайдений підінтервал та його відповідну лінійну функцію, щоб зрозуміти, як змінні взаємодіють у різних частинах діапазону даних.

## 2.2. Методи машинного навчання для прогнозування часових рядів

Методи машинного навчання для прогнозування часових рядів використовуються для аналізу та передбачення змін у часових рядах на основі різноманітних моделей та алгоритмів, які навчаються на вхідних даних.

Методи машинного навчання можуть використовувати дуже складні моделі, такі як нейронні мережі, дерева рішень, ансамблі моделей тощо. Це дозволяє їм виявляти складні залежності в часових рядах, що може бути складно або неможливо зробити за допомогою методів згладжування. Методи машинного навчання можуть автоматично виявляти важливі ознаки або закономірності у вхідних даних, що дозволяє покращити точність прогнозування. У методах згладжування часто використовуються певні передбачені шаблони (наприклад, середнє значення або тренди), що можуть бути менш гнучкими. Методи машинного навчання можуть ефективно працювати з великими обсягами даних, що дозволяє їм працювати з великими часовими рядами або враховувати багатомірність даних. Деякі методи машинного навчання можуть автоматично адаптуватися до змін у часових рядах, що робить їх більш гнучкими у порівнянні з традиційними методами згладжування. Методи машинного навчання потребують навчання на історичних даних для побудови моделі, що може бути часом трудомістким. У той час як методи згладжування можуть бути більш простими у використанні, оскільки вони часто базуються на аналізі попередніх значень часового ряду без необхідності в навчанні. [25]

В цілому, методи машинного навчання можуть бути потужним інструментом для прогнозування часових рядів, зокрема у випадках, коли вимоги до точності високі або коли присутні складні необхідності в аналізі даних. Однак вони також можуть вимагати більшого обсягу даних та обчислювальних ресурсів для ефективної роботи.

Найпопулярнішим методом прогнозування часових рядів є за допомогою використання штучних нейронних мереж. Штучні нейронні мережі (Artificial Neural Networks - ANN) є корисним інструментом для прогнозування часових рядів, оскільки вони можуть ефективно вирішувати складні проблеми прогнозування. [27]

Однією з важливих особливостей ANN у контексті прогнозування часових рядів є їх здатність до нелінійного моделювання без потреби в будь-яких припущеннях про статистичний розподіл часового ряду. Кожна модель нейронної мережі адаптивна формується на основі доступних даних. Таким чином, штучні нейронні мережі керуються даними і є самоадаптивними за своєю природою. [27]

Структура штучної нейронної мережі ґрунтується на з'єднаних вузлах, які називаються нейронами, подібно до біологічних нейронів у людському мозку.

Модель штучної нейронної мережі (ANN) може включати три типи шарів: вхідний шар, прихований шар та вихідний шар, які з'єднані без циклів. Деякі моделі можуть мати більше одного прихованого шару. Простою прикладом є тришарова архітектура ANN, яка може бути зображена схематично наступним чином.

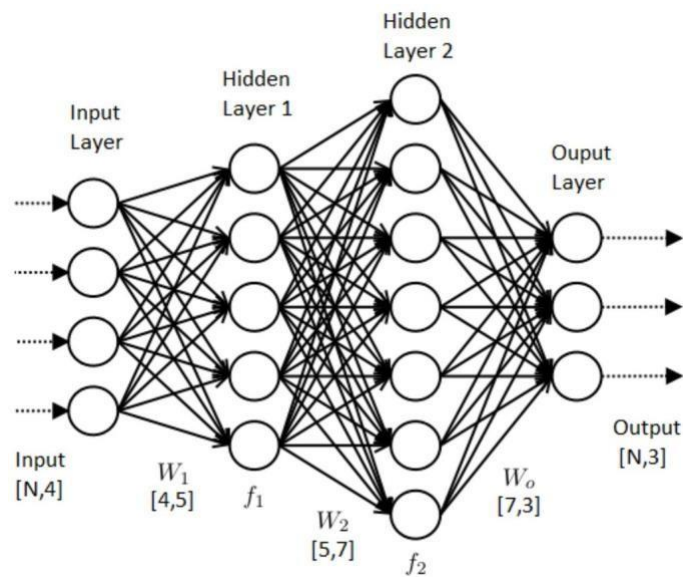


Рис 2.2.1. Схематичне відображення штучної нейронної мережі

Джерело: <https://www.datasciencecentral>

### 2.2.1. Методи моделювання вартості цінкових рядів за допомогою рекурентних нейронних мереж

Рекурентна нейронна мережа (RNN) вміє обробляти послідовності, такі як речення чи дані, що змінюються в часі. Вона працює, як розумну клітину зі своєю власною пам'яттю. Наприклад, розглянемо задачу передбачення слів у реченні. RNN не лише розуміє кожне слово, але також пам'ятає, що було перед тим, використовуючи свою внутрішню пам'ять. Ця пам'ять допомагає їй зафіксувати шаблони та зв'язки у послідовностях. Це робить RNN чудовим інструментом для завдань, таких як передбачення майбутніх значень в часових рядах даних, наприклад, цін на акції чи погодних умов, де минула інформація відіграє важливу роль. [27]

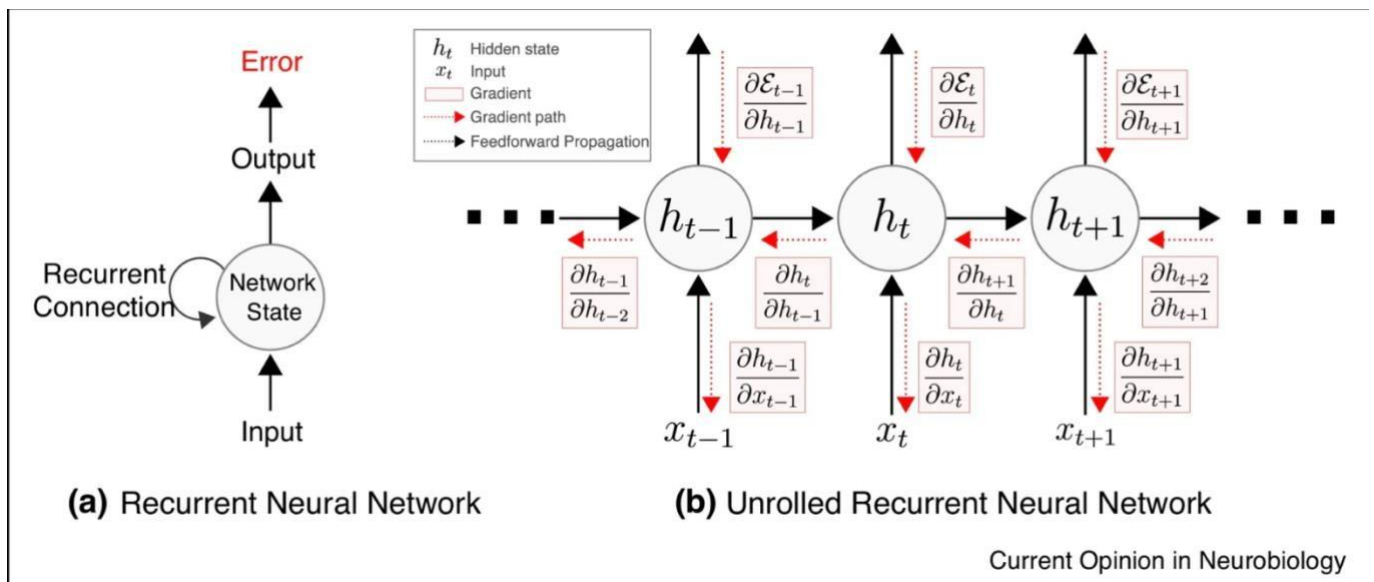


Рис. 2.2.1.1. Зворотній зв'язок у рекурентній нейронній мережі

Джерело: <https://encord.com/blog/time-series-predictions-with-recurrent-neural-networks/>

Рекурентні нейронні мережі (RNNs) мають кілька переваг для завдань прогнозування часових рядів. Вони можуть ефективно обробляти послідовні дані різної довжини, захоплюючи довгострокові залежності та часові шаблони. RNNs враховують неправильно розміщені проміжки часу та адаптуються до різних задач прогнозування з вхідними та вихідними послідовностями різної довжини.

Проте, у RNNs є обмеження, такі як проблема зникаючого або вибухаючого градієнту, що впливає на їхню здатність захоплювати довгострокові залежності. Ця проблема виникає, коли RNNs розгортаються на велику кількість кроків у часі, що

може призвести до обмежень пам'яті та обмежити їх продуктивність з дуже довгими послідовностями. Проблема зникнення градієнта - це виклик, який впливає на навчання глибоких нейронних мереж, включаючи Рекурентні Нейронні Мережі (RNNs). Вона виникає, коли градієнти, які показують напрям та величину оновлень ваг мережі під час навчання, стають дуже малими, коли вони поширюються назад через шари. Це явище ускладнює можливість RNNs вивчати довгострокові залежності та може призводити до повільного або неефективного навчання. Проблема зникнення градієнта особливо проблематична в послідовностях, де інформацію потрібно пам'ятати або передавати протягом тривалого часу, що впливає на можливість мережі захопити важливі закономірності. Щоб боротися з проблемою зникнення градієнта, яка ускладнює ефективне навчання у нейронних мережах, з'явилися кілька стратегій. Техніки, такі як належна ініціалізація ваг, пакетна нормалізація, обмеження градієнту, пропускні з'єднання та розкладка швидкості навчання, відіграють вирішальну роль у стабілізації потоку градієнту та запобіганні їхньої передчасної загибелі.

Проблема зникнення градієнта особливо проблематична в послідовностях, де інформацію потрібно пам'ятати або передавати протягом тривалого часу, що впливає на можливість мережі захопити важливі закономірності. Щоб боротися з проблемою зникнення градієнта, яка ускладнює ефективне навчання у нейронних мережах, з'явилися кілька стратегій. Техніки, такі як належна ініціалізація ваг, пакетна нормалізація, обмеження градієнту, пропускні з'єднання та розкладка швидкості навчання, відіграють вирішальну роль у стабілізації потоку градієнту та запобіганні їхньої передчасної загибелі. Хоча техніки, такі як LSTM та GRU, допомагають подолати деякі з цих проблем, інші високорівневі архітектури, такі як трансформатори, можуть перевершувати RNNs у певних складних сценаріях часових рядів, що вимагає уважного вибору моделі.

Будування та тренування ефективної моделі RNN для прогнозування часових рядів потребує підходу, який збалансовує архітектуру моделі та техніки навчання. Розглянемо всі необхідні кроки для побудови та тренування моделі RNN. Процес

включає підготовку даних, визначення архітектури моделі, побудову моделі, налаштування гіперпараметрів та оцінку продуктивності моделі.

Підготовка даних є ключовою для точних прогнозів часових рядів за допомогою RNNs. Обробка відсутніх значень та викидів, масштабування даних та створення відповідних вхідно-вихідних пар є невід'ємними. Вилучення сезонності та тренду допомагає виявити закономірності, тоді як вибір відповідної довжини послідовності збалансовує короткострокові та довгострокові залежності. Інженерія ознак, така як затримки, покращує продуктивність моделі. Належна підготовка даних забезпечує, що RNNs вивчають значущі закономірності та роблять точні прогнози на невидимих даних.

Побудова архітектури RNN включає вибір шарів та нейронів у мережі. Типова структура для прогнозування часових рядів включає вхідний шар, один або декілька прихованих шарів з клітинами LSTM або GRU та вихідний шар. Вибір оптимальної кількості шарів та нейронів є критичним етапом в побудові моделі RNN. Замало шарів або нейронів може призвести до недонавчання, тоді як занадто багато може призвести до перенавчання. Важливо знайти баланс між складністю моделі та узагальненням. Такі техніки як перехресна перевірка та пошук по сітці, щоб знайти оптимальні гіперпараметри. Налаштування гіперпараметрів включає пошук найкращого набору гіперпараметрів для моделі RNN. Гіперпараметри включають швидкість навчання, розмір пакета, кількість епох та силу регуляризації. Можна використовувати перехресну перевірку або випадковий пошук для дослідження різних комбінацій гіперпараметрів та ідентифікації конфігурації, яка дає найкращі результати.

Навчання RNN передбачає подання послідовних даних з використанням алгоритмів навчання моделі та ітеративне оновлення її параметрів для мінімізації помилки прогнозування. Шляхом подачі історичних послідовностей у RNN вона навчається захоплювати закономірності та залежності у даних. Процес зазвичай включає пряме розповсюдження для обчислення прогнозів та зворотне розповсюдження для оновлення ваг моделі за допомогою алгоритмів оптимізації, таких як стохастичний градієнтний спуск (SGD) або Adam. Зауважимо, що

перенавчання - це загальна проблема в моделях глибокого навчання, включаючи RNNs. Використовуються техніки регуляризації, такі як L1 і L2 регуляризація, виключення (dropout) та раннє зупинення, щоб запобігти перенавчанню та покращити загальну продуктивність моделі.

Нестаціонарні часові ряди виявляють змінні статистичні властивості, такі як зміна середнього або дисперсії, з часом. Робота з нестаціонарністю є ключовою, оскільки традиційні моделі припускають стаціонарність. Техніки, такі як віднімання, детрендинг або сезонне розкладання, можуть допомогти перетворити дані в стаціонарну форму. Крім того, передові методи, такі як сезонний авторегресійний інтегрований ковзний середній (SARIMA) або Prophet, можуть бути використані для моделювання та прогнозування нестаціонарних часових рядів. Часові ряди в реальному світі часто мають нерегулярні частоти та відсутні мітки часу, що руйнує здатність моделі вивчати шаблони. Можна застосувати техніки ресемплінгу (наприклад, інтерполяція, агрегація), щоб перетворити дані на регулярну частоту. Для відсутніх міток часу застосовуються техніки імпутації, такі як заповнення вперед і назад або більш складні методи, наприклад моделі імпутації часових рядів.

RNN може ефективно моделювати часові залежності в даних, що робить його корисним для прогнозування цінових рядів, які мають складні темпоральні взаємозв'язки. У випадках, коли цінові ряди представлені у вигляді послідовностей, RNN може бути ефективним інструментом для аналізу, оскільки він може враховувати порядок даних у вхідних послідовностях. RNN може автоматично виявляти та використовувати як короткострокові, так і довгострокові залежності в даних, що дозволяє йому краще адаптуватися до різних трендів та паттернів у цінових рядах.

Проте, є деякі сценарії, де метод рекурентних нейронних мереж може працювати гірше. Наприклад, у разі обмежених або неякісних даних RNN може виявитися неефективним, оскільки він може недостатньо точно вивчати закономірності в даних та здійснювати точний прогноз. Якщо в цінових рядах є значна кількість шуму або непередбачуваних факторів, які впливають на ціни, RNN може мати складнощі з

усуненням цього шуму та точним прогнозуванням. Якщо умови ринку раптово змінюються або різко коливаються, RNN може мати проблеми з адаптацією до цих змін, особливо якщо йому не надано достатньої інформації про такі зміни.

### **2.2.2. Методи моделювання вартості цінових рядів за допомогою згорткових нейронних мереж**

У сфері глибокого навчання Згорткові Нейронні Мережі (CNN) революціонізували область візуального аналізу. З їх здатністю видобувати витончені шаблони та ознаки зображень, CNN стали невід'ємною частиною завдань, таких як класифікація зображень, виявлення об'єктів та розпізнавання облич. Архітектура згорткових нейронних мереж (CNN) складається з кількох шарів, які спільно витягують та навчаються важливим ознакам зображень. Цей унікальний дизайн дозволяє CNN винятково справлятися з такими завданнями, як класифікація зображень, виявлення об'єктів та семантична сегментація.

Згорткові нейронні мережі (CNN) застосовуються в прогнозуванні акцій шляхом аналізу та використання великих обсягів фінансових даних, таких як цінові ряди, обсяги торгів, індикатори технічного аналізу тощо. Основні способи використання CNN у цій області включають:

**Аналіз графіків цінових рядів:** CNN можуть бути навчені аналізувати графіки цінових рядів акцій та визначати патерни та тренди, що можуть бути корисними для прогнозування майбутніх цінових рухів.

**Використання новин та соціальних медіа:** CNN можуть бути застосовані для аналізу новинних статей, соціальних медіа та інших джерел інформації, щоб виявляти новісні події та тренди, які можуть впливати на ціни акцій.

**Поєднання з іншими методами прогнозування:** CNN часто використовуються разом з іншими методами машинного навчання та статистичного аналізу, такими як рекурентні нейронні мережі (RNN), дерева рішень тощо, для покращення точності прогнозів.

Використання CNN у прогнозуванні акцій дозволяє автоматизувати аналіз великих обсягів даних та виявляти складні залежності між різними факторами, що впливають на ціни акцій. Однак важливо пам'ятати, що ринки фінансових активів дуже складні та непередбачувані, тому навіть найкращі моделі прогнозування можуть допускати помилки.

Ядро CNN - це згортковий шар. Він застосовує набір вивчених фільтрів до вхідного зображення, конволюючи їх по простору зображення. Кожен фільтр вивчає виявлення конкретних шаблонів або ознак, таких як краї, кути або текстури. Вихід цього шару - це набір карт ознак, де кожна карта представляє активацію певного фільтра. Після згорткових шарів зазвичай застосовують функції активації, такі як Rectified Linear Units (ReLU). Вони вводять нелінійність в мережу, дозволяючи CNN вивчати складні взаємозв'язки між ознаками. Наприклад, ReLU встановлює від'ємні значення на нуль, зберігаючи позитивні значення, що підвищує здатність мережі моделювати нелінійні трансформації. Шари пулінгу зменшують просторові розміри карт ознак, наприклад, за допомогою техніки максимального пулінгу, де вибирається максимальне значення в певній області. Пулінг допомагає зменшити обчислювальну складність, поліпшити інваріантність до трансляції та захопити найбільш суттєві ознаки. Повністю зв'язані шари, також відомі як плотні шари, відповідають за зроблення остаточних прогнозів на основі витягнутих ознак. Вони з'єднують кожен нейрон з попереднього шару з кожним нейроном у поточному шарі, інтегруючи інформацію з карт ознак та вивчаючи високорівневі представлення, що дозволяє виконувати класифікаційні або регресійні завдання.

Дропаут - це техніка регуляризації, часто використовувана в CNN для запобігання перенавчанню. Під час навчання випадково вибрані нейрони в мережі тимчасово вимикаються, тобто їх виходи встановлюються на нуль. Це змушує мережу полагатися на залишені нейрони та запобігає співадаптації нейронів, покращуючи узагальнення. У завданнях класифікації зазвичай використовується softmax-шар в кінці архітектури CNN. Він нормалізує виходи останнього повністю зв'язаного шару, призначаючи ймовірності кожному класу. Клас з найвищою

ймовірністю вважається передбаченою міткою. Архітектура CNN зазвичай слідує послідовному зразку, починаючи з чергуються згорткових та пулінгових шарів, а потім - повністю зв'язаних. Кількість шарів, їх розміри та розташування можуть варіюватися залежно від складності завдання та доступних обчислювальних ресурсів.

Згорткові нейронні мережі (CNN) трансформували візуальний аналіз завдяки їх здатності видобувати складні патерни та ознаки зображень. Від класифікації зображень до виявлення об'єктів та розпізнавання облич, CNN стали ключовим інструментом для розуміння та інтерпретації візуальних даних. Імітуючи людську візуальну систему та використовуючи техніки глибокого навчання, CNN пропонують небачену точність та ефективність у аналізі складних зображень. Поступово розвиваючись та інтегруючись у різноманітні сфери, вплив CNN на комп'ютерне зорове сприйняття та штучний інтелект буде лише зростати, відкриваючи нові можливості для інновацій та прогресу в галузі.

Метод моделювання вартості цінних рядів за допомогою згорткових нейронних мереж (CNN) може бути корисним і ефективним підходом у ряді ситуацій. Ось кілька випадків, коли використання CNN може бути вигідним. Якщо дані представлені у вигляді графіків або зображень, наприклад, свічки OHLC (Open-High-Low-Close), то CNN може ефективно використовувати локальні шаблони для виявлення важливих паттернів, які можуть бути корисними для прогнозування. Якщо в цінних рядах існують важливі просторові залежності між даними, CNN може ефективно виявляти ці залежності, що робить його корисним для аналізу цінних рядів. CNN може бути використаний для фільтрації шуму в цінних рядах та виділення важливих ознак, що може поліпшити точність моделі та прогнозу. У випадку великих обсягів даних CNN може бути ефективним інструментом, оскільки він може обробляти великі об'єми інформації швидше за інші моделі.

Проте, існують певні обмеження та ситуації, коли використання CNN може бути менш ефективним. Якщо цінні ряди не мають просторової структури або важливих локальних залежностей, використання CNN може бути зайвим і неефективним. Якщо вимірність даних надто велика, CNN може виявитися менш ефективним у

порівнянні з іншими методами, оскільки він може вимагати значних обсягів обчислювальних ресурсів та часу для навчання. CNN може мати тенденцію до перенавчання у випадку, коли він має велику кількість параметрів, що може змусити його недооцінювати нові дані.

Отже, у кожному конкретному випадку використання CNN для моделювання вартості цінних рядів слід враховувати особливості даних та характеристики задачі прогнозування.

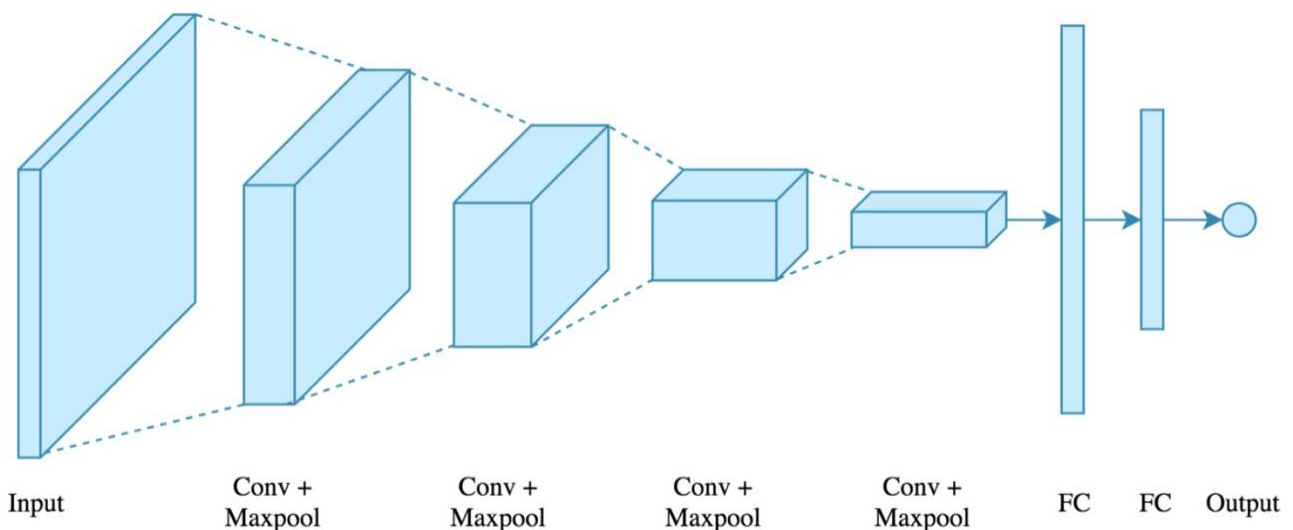


Рис. 2.2.2.1. Схема згорткової нейронної мережі (CNN)

Джерело: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00599-4>

### 2.2.3. Методи моделювання вартості цінних рядів за допомогою багат шарового перцептрона Румельхарта

Штучні нейронні мережі представляють собою електронні моделі, інспіровані структурою та функціонуванням нейронної системи людського мозку. Вони використовуються для моделювання та аналізу складних взаємозв'язків у великих наборах даних. Найпростішим типом нейронної мережі є одношаровий перцептрон, який складається з одного шару вихідних вузлів, що зв'язані з входами. Однак така

мережа має обмежену здатність до розв'язання складних завдань через її лінійну структуру.

Багатошарові нейронні мережі складаються з декількох шарів обчислювальних вузлів, що зв'язані між собою в прямому напрямку. Кожен вузол одного шару пов'язаний з вузлами наступного шару. У таких мережах нейрони можуть використовувати різні функції активації, такі як сигмоїдна або ReLU, для нелінійного моделювання складних залежностей в даних.

Ці мережі можуть навчатися за допомогою алгоритмів навчання, таких як зворотне розповсюдження помилки, які коригують ваги зв'язків між нейронами з метою мінімізації помилки прогнозування на тренувальних даних.

Важливою властивістю багатошарових нейронних мереж є їх універсальність у наближенні будь-якої функції з високою точністю за умови належного навчання та належного підбору параметрів мережі. Ця універсальність гарантує, що мережі можуть адаптуватися до різних завдань і видаляти складні залежності в даних, що робить їх потужним інструментом для багатьох областей, включаючи прогнозування акційних ринків.

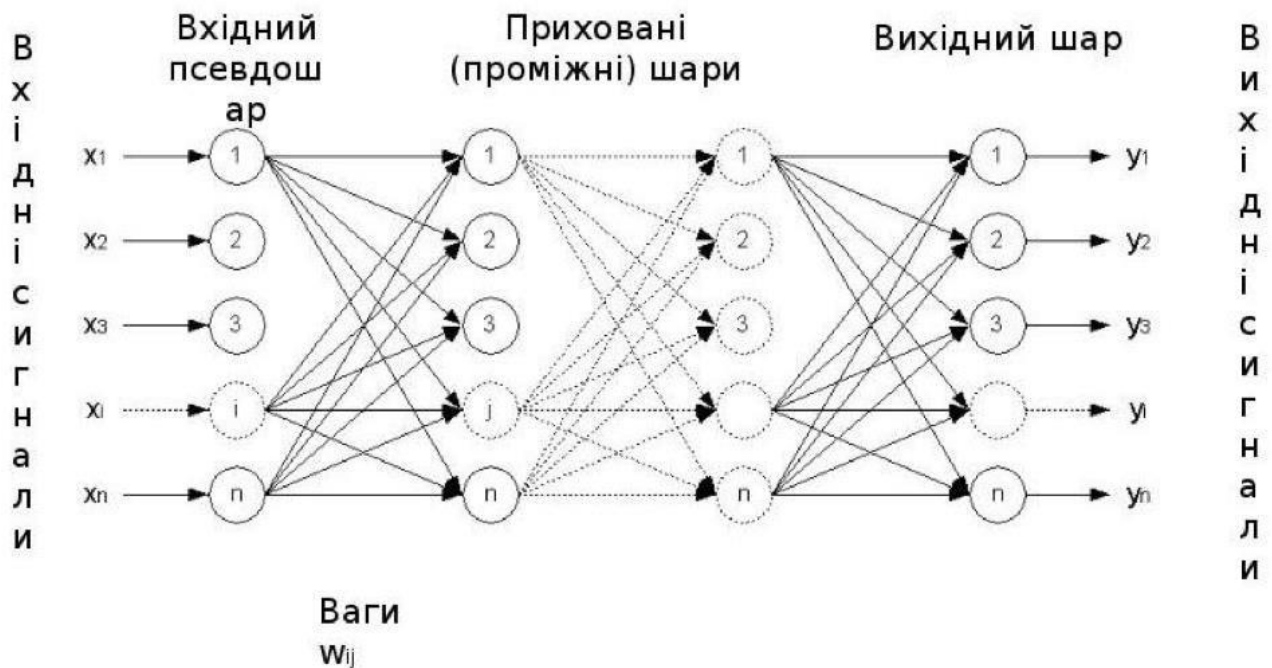


Рис. 2.2.3.1. Приклад багатошарової нейронної мережі

Джерело: [https://link.springer.com/chapter/10.1007/978-3-642-13318-3\\_70](https://link.springer.com/chapter/10.1007/978-3-642-13318-3_70)

Активаційні функції нейронів у багатошарових нейронних мережах використовуються зазвичай у вигляді сигмоїдальної функції, такої як логістична або гіперболічний тангенс. Ці мережі демонструють можливість знаходити наближені рішення для надзвичайно складних задач і вважаються універсальним апроксиматором функцій, здатним до побудови регресійних моделей. Пік популярності багатошарових перцептронів у машинному навчанні припав на 1980-ті роки, особливо в галузях, таких як розпізнавання мови та зображень, а також системи машинного перекладу.

Хоча в подальшому ці мережі зазнали конкуренції з іншими технологіями машинного навчання, такими як машини опорних векторів, інтерес до них повернувся завдяки успіхам глибокого навчання. Багатошаровий перцептрон, спочатку запропонований Ф. Розенблатом, у своєму сучасному вигляді був розроблений Д. Румельхартом.

Підхід Румельхарта відрізняється від підходу Розенבלата за наступними властивостями: використання нелінійних активаційних функцій, можливість використання більше ніж одного прихованого шару (зазвичай не більше трьох), а також кодування вхідних сигналів десятковими числами, нормованими до інтервалу  $[0,1]$ . Навчання в таких мережах здійснюється з метою стабілізації ваг мережі, що дозволяє уникнути перенавчання.

Сучасні багатошарові перцептрони залишаються популярним інструментом аналізу даних і використовуються у багатьох платформах бізнес-аналітики.

Ось деякі ситуації, коли використання MLP може бути вигідним. У випадку, коли цінові ряди мають прості лінійні залежності або нелінійні залежності, які можна розділити гіперплощиною, MLP може ефективно моделювати такі зв'язки. У випадку, коли ви маєте велику кількість даних із зазначеними властивостями, MLP може бути ефективним варіантом, оскільки він може вивчити складні залежності з великим обсягом даних. MLP - це досить простий алгоритм, який легко реалізувати з використанням бібліотек машинного навчання, таких як TensorFlow або PyTorch. Він

не вимагає великої кількості гіперпараметрів для налаштування, що може спростити процес моделювання.

Проте, є деякі обмеження та ситуації, коли використання MLP може бути менш ефективним. У випадку, коли цінові ряди мають складні нелінійні залежності, які важко апроксимувати за допомогою лінійних моделей, MLP може бути недостатньо потужним для точного моделювання таких зв'язків. Якщо в цінових рядах є значна кількість шуму або непередбачуваних факторів, які впливають на ціни, MLP може мати тенденцію до перенавчання або недооцінювання, що призведе до непрогнозованих результатів. У порівнянні з іншими методами, MLP може вимагати більше обчислювальних ресурсів, особливо у випадку великої кількості шарів та нейронів.

Важливо враховувати контекст та характеристики даних при виборі методу моделювання цінових рядів з використанням багат шарового перцептрона Румельхарта.

#### **2.2.4. Методи моделювання вартості цінових рядів за допомогою моделі випадкового лісу**

Випадкові ліси були вперше представлені у 2001 році Брейманом і є одним із найпопулярніших алгоритмів у машинному навчанні. Популярність впливає з широкого спектру застосувань, на яких вони відомі за хорошу продуктивність навіть у високорозмірних даних, швидкість обчислень та легкість налаштування. Успішні застосування можуть бути зазначені у хімоінформатиці, екології, розпізнаванні 3D-об'єктів та прогнозуванні часових рядів. [36]

Модель випадкового дерева для прогнозування цінових рядів використовується для аналізу та прогнозування цін на акції, товари, валютні курси та інші фінансові інструменти. Ця модель базується на алгоритмі випадкового лісу, який є ансамблем дерев рішень.

Основна ідея випадкового лісу полягає в тому, що він комбінує декілька дерев рішень із випадковим вибором підмножини ознак для побудови кожного дерева. Кожне дерево випадкового лісу навчається на випадковій підмножині даних та випадково вибраній підмножині ознак. Потім прогнози кожного дерева узгоджуються або усереднюються, щоб отримати кінцевий прогноз.

У випадку прогнозування цінових рядів, модель випадкового дерева може аналізувати історичні дані цін, обсягів торгів та інших факторів, що можуть впливати на цінову динаміку. Вона може допомогти інвесторам та трейдерам у прийнятті рішень щодо торгівлі та управління ризиками, шляхом надання прогнозів цін на майбутнє. Однак важливо пам'ятати, що фінансові ринки динамічні та непередбачувані, і жодна модель не може гарантувати точний прогноз цінових рухів.

Випадкові ліси можна пов'язати з двома основними джерелами: деревами регресії та методом усіпання. Дерева регресії будуються шляхом рекурсивного розбиття простору вхідних даних на основі якогось критерію для оцінки регресійної функції  $f$ . На кожному кроці побудови дерева вибирається розбиття (змінна і місце на змінній) на основі оцінки критерію серед усіх допустимих розбиттів на основі всіх змінних. Клітина розбивається на дві на вибраному розбитті, і попередній крок повторюється на нових клітинах. Дерево є частково сталою декомпозицією простору вхідних даних. До вихідної розділізації простору вводиться бінарне дерево. Кожен вузол відповідає тесту, який відображає, як простір був розрізаний. На рис. 1 показано поділ у двовимірному просторі та відповідне бінарне дерево. Принцип усіпання (скорочено від "бутстреп-агрегування") полягає у створенні  $M$  випадково згенерованих наборів навчальних даних шляхом випадкового вибору  $a_n$  спостережень з або без повторення з набору  $D_n$ , і побудові на кожному наборі прогнозувальника. Після побудови прогнозувальників усіпання для нового спостереження  $x$  полягає в агрегації, як правило, емпіричного середнього, прогнозів, отриманих від  $M$  прогнозувальників для точки  $x$ . Ця процедура спрямована на покращення стабільності та точності базового прогнозувальника. В контексті

випадкових лісів прогнозувальниками є дерева регресії. Для пояснення процедури випадкових лісів нам потрібно докладніше розглянути побудову одного дерева.

Перший крок - це бутстреп/підвибірка: обираються  $\alpha_n$  точок з або без повторень серед  $n$  реалізацій. Потім на цих  $\alpha_n$  вибраних точках будується дерево. На кожному вузлі дерева найкращий розподіл (змінна і місце на цій змінній) визначається шляхом мінімізації внутрішньо-вузлової дисперсії. Це загально відомо як критерій *CART*, введений у літературі. Замість мінімізації цього критерію серед всіх можливих розподілів на основі всіх змінних вибір вхідних даних обмежується випадковим підмножиною фіксованого розміру  $mtry$ . Ця процедура потім повторюється на кожному вузлі, який утворюється після бінарного розбиття, до виконання умов зупинки. Перше правило зупинки - це коли дисперсія в вузлі дорівнює нулю. Оскільки це вкрай рідкісний випадок, друга умова полягає в тому, що кількість спостережень в вузлі повинна бути більшою за заданий поріг. [36]

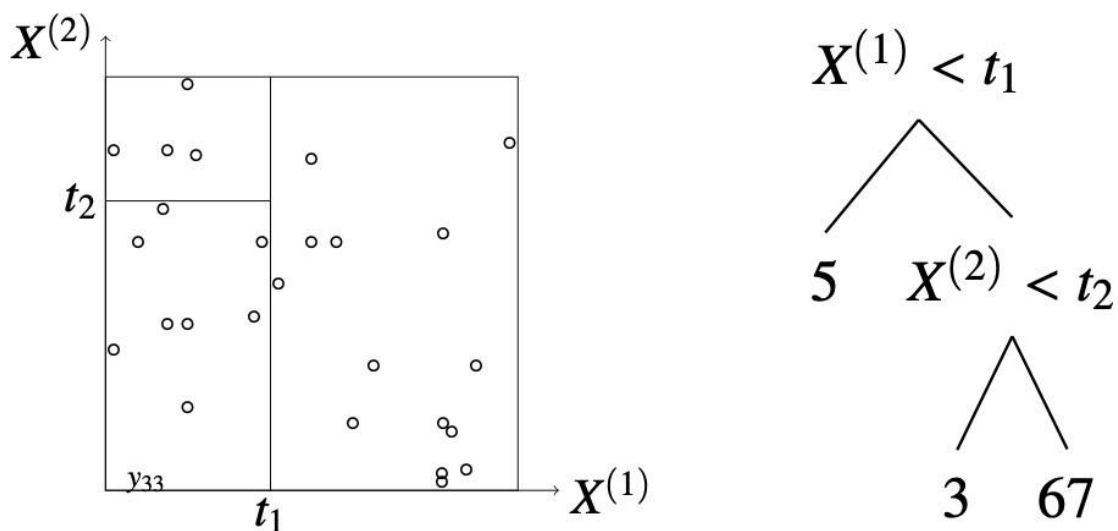


Рис. 2.2.4.1. Частинний випадок моделі випадкового дерева

Джерело: <https://www.ine.pt/revstat/pdf/Randomforestsfortimeseries.pdf>

Розглянемо декілька випадків, коли доцільно використовувати модель випадкового лісу. Модель випадкового лісу може ефективно працювати з великою кількістю ознак, що може бути важливим у фінансовому аналізі, де досліджуються

багато потенційно впливових факторів. Випадковий ліс може моделювати складні нелінійні залежності між цінними рядами та їх впливом на ціни, що може бути корисним для точного прогнозування. Випадковий ліс надає оцінку важливості ознак, що дозволяє виявити найбільш впливові фактори в моделі, що може бути корисним для розуміння та інтерпретації результатів. Модель випадкового лісу має механізми для уникнення перенавчання, що дозволяє ефективно працювати з наборами даних різної складності та розміру без значного погіршення результатів.

Проте, є деякі обмеження та ситуації, коли використання моделі випадкового лісу може бути менш ефективним. Якщо в цінних рядах є значна кількість шуму або непередбачуваних факторів, які впливають на ціни, модель випадкового лісу може показати менш точні результати. У випадку обмежених обсягів даних модель випадкового лісу може мати обмежену ефективність, особливо якщо потрібна велика кількість ознак для точного прогнозу. Хоча модель випадкового лісу може давати точні прогнози, її результати можуть бути складні для інтерпретації через велику кількість дерев у моделі.

У загальному, модель випадкового лісу є потужним та ефективним інструментом для моделювання вартості цінних рядів, особливо у випадках, коли потрібно враховувати багато ознак та нелінійні залежності.

### **2.2.5. Методи моделювання вартості цінних рядів за допомогою довготривалої короткострокової пам'яті**

LSTM (Long Short-Term Memory) - це тип рекурентної нейронної мережі, яка добре підходить для аналізу послідовних даних, таких як часові ряди. LSTM здатний зберігати та використовувати інформацію з попередніх кроків часового ряду, що робить його ефективним інструментом для прогнозування.

Побудова даної моделі включає декілька кроків. Спочатку необхідно підготувати дані. Це може включати очищення даних від пропущених значень, нормалізацію даних для зручності навчання нейронної мережі та розділення даних на тренувальний

і тестовий набори. Після підготовки даних слід побудувати модель LSTM. Це може включати визначення кількості шарів та нейронів в кожному шарі, вибір функції втрат та оптимізатора для навчання. Навчання моделі включає передачу тренувальних даних через LSTM та налаштування параметрів моделі за допомогою зворотного поширення помилок. Після завершення навчання необхідно оцінити ефективність моделі за допомогою тестового набору даних. Це дозволяє з'ясувати, наскільки добре модель прогнозує ціни на ціновому ряді. За допомогою навченої моделі можна прогнозувати майбутні ціни на ціновому ряді. Для цього використовується навчена модель, яка бере вхідні дані, що представляють собою останні точки даних, і генерує прогнози на основі цих даних. Після прогнозування необхідно оцінити результати, порівняти їх з реальними даними та здійснити візуалізацію прогнозів і фактичних значень для оцінки точності моделі.

LSTM (Long Short-Term Memory) має відносну нечутливість до довжини проміжків, що є його перевагою порівняно з іншими рекурентними нейронними мережами, прихованими моделями Маркова та іншими методами вивчення послідовностей. Він має на меті надати короткочасну пам'ять для RNN, яка може тривати тисячі кроків часу, отже "довготривала короткочасна пам'ять". Він застосовний до класифікації, обробки та прогнозування даних на основі часових рядів, таких як рукопис, розпізнавання мови, машинний переклад, виявлення мовної активності, керування роботами, відеоігри та охорона здоров'я. [45]

Загальний блок LSTM складається з комірки, входної воронки, вихідної воронки та воронки забуття. Комірка запам'ятовує значення протягом довільних проміжків часу, а три воронки регулюють потік інформації до та з комірки. Воронки забуття вирішують, яку інформацію викинути з попереднього стану, присвоюючи попередньому стану, порівняно з поточним входом, значення між 0 та 1. Значення, близьке до 1, означає зберегти інформацію, а 0 - викинути її. Воронки входу вирішують, які частини нової інформації зберігати в поточному стані, використовуючи ту саму систему, що і воронки забуття. Воронки виходу контролюють, яку інформацію з поточного стану вивести, присвоюючи значення від

0 до 1 інформації, враховуючи попередній та поточний стани. Вибірковий вивід важливої інформації з поточного стану дозволяє мережі LSTM зберігати корисні довгострокові залежності для здійснення прогнозів як на поточні, так і на майбутні кроки часу. [29]

Розглянемо ситуації, коли доцільно використовувати дану модель для прогнозування часових рядів. LSTM ефективно працює з послідовними даними, такими як цінові ряди, оскільки він може зберігати та використовувати інформацію з попередніх кроків для прогнозування майбутніх значень. У фінансових ринках існують довготривалі залежності та цикли, які можуть впливати на ціни активів у майбутньому. LSTM може ефективно враховувати ці залежності та використовувати їх для прогнозування. LSTM може бути корисним для прогнозування цінових рядів у реальному часі, оскільки він може швидко адаптуватися до нової інформації та змінюючихся умов ринку. LSTM може допомагати трейдерам та інвесторам приймати рішення щодо купівлі, продажу або утримання активів, враховуючи історичні та поточні дані цінових рядів.

Проте, важливо враховувати, що LSTM може мати обмеження та ситуації, коли його використання може бути менш ефективним. Наприклад, у разі недостатньої кількості даних LSTM може показати менш точні результати, оскільки він вимагає великої кількості даних для ефективного навчання. Якщо у цінових рядах є значна кількість аномалій або шуму, LSTM може мати складнощі з врахуванням цих непередбачуваних величин у прогнозах. LSTM може вимагати значних обчислювальних ресурсів, особливо у випадку великої кількості нейронів та шарів.

У цілому, LSTM є потужним інструментом для моделювання вартості цінових рядів, здатним враховувати складні динаміки та залежності на фінансових ринках. Однак ефективність його застосування залежить від якості даних та правильного налаштування моделі.

## 2.2.6. Методи моделювання вартості цінових рядів за допомогою методу рекурсивного усунення ознак

Метод рекурсивного усунення ознак (Recursive Feature Elimination, RFE) є стратегією відбору ознак, яка використовується для покращення моделей машинного навчання шляхом поступового видалення ознак з набору даних. Цей метод особливо корисний у випадках, коли у вас є велика кількість ознак і необхідно вибрати найбільш інформативні з них для побудови моделі. [25]

Побудова даної моделі починається із побудови моделі машинного навчання, наприклад, лінійної регресії, дерева рішень або нейронної мережі. Ця модель повинна бути навчена на вихідних даних без видалення жодних ознак. Після навчання моделі оцінюємо важливість кожної ознаки для прогнозу. Це може бути здійснено, наприклад, за допомогою коефіцієнтів регресії у випадку лінійної регресії або значень важливості ознак у випадку дерева рішень. Обираємо ознаку з найменшим впливом на прогноз та вилучаємо її з моделі. Після видалення ознаки перенавчаємо модель на залишкових ознаках. Перевіряємо критерій зупинки, такий як кількість залишених ознак або покращення метрики якості моделі. Якщо критерій не виконаний, повертаємося до кроку 2. Коли критерій зупинки виконаний, завершуємо процес. Останні ознаки, що залишилися, є найважливішими для моделі. [29]

У разі лінійної регресії важливість ознаки може бути оцінена за допомогою абсолютних значень коефіцієнтів регресії. Наприклад, якщо ми маємо лінійну регресійну модель з ознаками  $X_1, X_2, \dots, X_n$  і коефіцієнтами регресії  $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ , то важливість ознаки може бути оцінена за допомогою відповідного коефіцієнта регресії. У випадку дерева рішень важливість ознаки може бути оцінена за допомогою значень важливості, які зазвичай надаються алгоритмом під час навчання моделі. [45]

Рекурсивна елімінація ознак є потужним методом відбору ознак, який може бути успішно використаний для прогнозування часових рядів, дозволяючи покращити якість моделі та зменшити перенавантаження шляхом використання лише найважливіших ознак. [34]

Ми задаємо параметр для збереження  $I$  числа ознак, і на кожній ітерації відбору ознак зберігаємо набір ознак, які отримали найвищий рейтинг, потім підганяємо модель і знову оцінюємо продуктивність, щоб почати нову ітерацію. У підсумку алгоритм ранжирування визначає набір кращих ознак.

Відомо, що алгоритм RFE страждає від проблеми надмірної підгонки. Щоб усунути проблему надмірної підгонки, ми запустимо алгоритм RFE кілька разів на випадково обраних акціях в якості навчального набору і переконаємося, що всі вибрані ознаки мають високу вагу.

Якщо розглядати набори даних, які містять велику кількість ознак, RFE може допомогти ідентифікувати найбільш інформативні ознаки, що є ключовими для прогнозування вартості цінового ряду. RFE допомагає уникнути перенавчання, забираючи менш важливі ознаки з набору даних, що може покращити загальну здатність моделі до узагальнення на нові дані. Використання меншої кількості ознак може зменшити обсяг обчислень та пам'яті, необхідний для тренування моделі, особливо в випадку великих наборів даних.

Проте, варто розглянуто деякі обмеження. Під час поступового вилучення ознак може втрачатися частина інформації, що може призвести до втрати точності моделі. RFE може бути часомістким процесом, особливо якщо ви працюєте з великими наборами даних та складними моделями. Для ефективного використання RFE може бути необхідним налаштування гіперпараметрів, таких як кількість ознак для вилучення на кожному кроці, що може вимагати додаткового часу та експертного знання.

Узагальнюючи, метод рекурсивного усунення ознак може бути корисним і ефективним підходом для моделювання вартості цінних рядів, здатним до ефективного відбору найбільш інформативних ознак і покращення точності моделі. Однак перед його застосуванням важливо враховувати контекст задачі та виконувати належне налаштування параметрів для досягнення оптимальних результатів.

## Висновки до розділу 2

Прогнозування акцій компанії Google вимагає використання різноманітних методів та моделей для аналізу ринкових даних, трендів та інших факторів, що впливають на їхню вартість. В основі такого прогнозування лежить потреба у точних та надійних прогностичних моделях, які допоможуть інвесторам та фінансовим аналітикам у прийнятті обґрунтованих рішень. Розглянемо, чому використання економетричних моделей, авторегресії, згладжування та методів машинного навчання є критично важливим для прогнозування акцій Google.

Економетричні моделі дозволяють аналізувати складні взаємозв'язки між різними економічними факторами та ціновими змінами акцій. Вони допомагають інтерпретувати вплив фундаментальних показників, таких як доходи компанії, фінансовий стан, кон'юнктура ринку тощо, на ціну акцій. Моделі авторегресії використовуються для прогнозування майбутніх значень на основі попередніх спостережень. У випадку акцій Google це може означати аналіз часових рядів цін акцій та їхніх змін з метою передбачення майбутніх тенденцій. Техніки згладжування допомагають зменшити шум та випадкові коливання в даних, що може бути особливо важливим при аналізі цінових динамік акцій. Використання згладжування дозволяє виділити основні тренди та патерни, що полегшує їхнє прогнозування.

Методи машинного навчання, такі як нейронні мережі, випереджаючі моделі та кластерний аналіз, можуть допомогти виявити складні неявні зв'язки між різними факторами та ціновими змінами акцій, які можуть бути непоміченими за допомогою традиційних методів аналізу.

Отже, використання економетричних моделей, авторегресії, згладжування та методів машинного навчання є критично важливим для прогнозування акцій компанії Google. Ці методи дозволяють аналізувати великі обсяги даних, виявляти складні зв'язки та робити точні прогнози, що є ключовими для успішного інвестування та прийняття фінансових рішень.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕТОДІВ ЗГЛАДЖУВАННЯ, ЕКОНОМЕТРИЧНИХ МОДЕЛЕЙ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ

### 3.1. Збір та підготовка даних

Для реалізації методів прогнозування вартості акцій компанії Google LLC було використано дані з офіційного сайту компанії. Показники взято за 5 років, починаючи з 30 січня 2020 р. по 30 січня 2025 р.

У табл. 1 наведено частину бази даних, яка була використана для реалізації подальших методів.

Таблиця 1

Зразок бази даних, яка була використана для прогнозування вартості акцій компанії

Google LLC

Date	Price	Open	High	Low	Vol.	Change %
01/30/2025	200.87	198.00	201.40	197.67	24.35M	2.79%
01/29/2025	195.41	195.55	196.79	193.43	18.22M	0.06%
01/28/2025	195.30	192.74	195.48	190.68	24.16M	1.82%
01/27/2025	191.81	192.41	196.88	190.73	41.73M	-4.20%
01/24/2025	200.21	198.10	200.90	198.00	23.88M	1.13%
01/23/2025	197.98	198.14	200.30	195.20	26.95M	-0.20%
01/22/2025	198.37	199.06	200.48	197.53	26.20M	0.16%
01/21/2025	198.05	199.07	202.29	197.87	29.97M	1.05%
01/17/2025	196.00	196.53	197.23	193.75	27.74M	1.60%
01/16/2025	192.91	194.14	195.48	192.81	17.82M	-1.35%
01/15/2025	195.55	193.09	196.36	191.86	21.78M	3.11%
01/14/2025	189.66	191.24	191.98	188.31	17.17M	-0.71%
01/13/2025	191.01	190.07	191.18	187.36	21.82M	-0.54%
01/10/2025	192.04	194.29	196.52	190.31	26.67M	-0.98%
01/08/2025	193.95	192.57	196.29	192.38	24.86M	-0.79%

Джерело: <http://www.googlers.ua/videnie/1.php>

Спочатку за допомогою відповідних бібліотек Python відсортовано дані за спаданням дати. Потім видалено стовпчики з непотрібними в аналізі даними (volume,

close). Після цього обраховано середнє значення за кожен день (на основі нього побудовано подальші моделі) за відповідною формулою:

$$\text{average} = \frac{\text{high} + \text{low}}{2}$$

Для оптимізації роботи програми середні значення нормалізуються, використовуючи наблизений до нормального розподіл. Потім ряд розбивається на два ряди: тренувальні та тестові у відношенні 70% до 30%. На рис. 6 представлений графік цін протягом періодів 30 січня 2020 р. по 30 січня 2025 р.

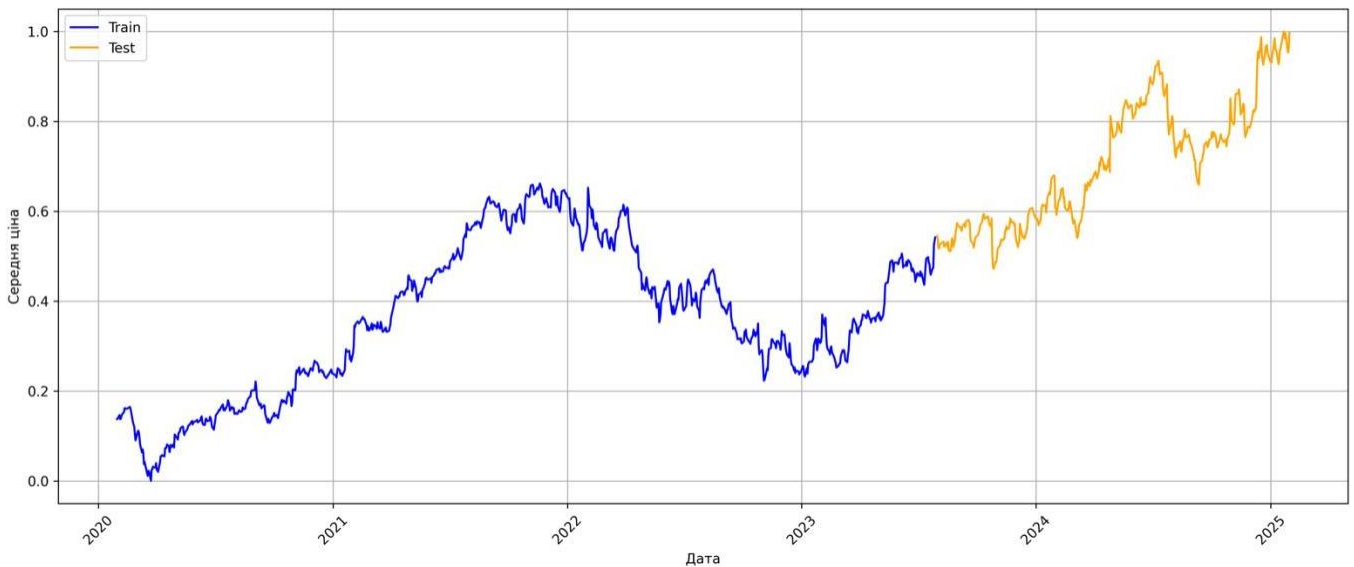


Рис. 3.1.1. Розбиття даних на тестову та тренувальну вибірки

Джерело: розроблено автором

Як видно з рис. 6, наш ряд не має сезонних коливань (ціна не повторюється в залежності від дня, тижня або місяця). Далі розглянемо доведення цього твердження іншими методами. Для цього часовий ряд був розбитий на вибірку та підвибірки: 5 років, 3 роки, 2 роки та 1 рік.

Стаціонарність є ключовою частиною аналізу часових рядів. Простіше кажучи, стаціонарність означає, що спосіб зміни даних часових рядів є постійним. Стаціонарний часовий ряд не матиме жодних тенденцій або сезонних моделей. Варто перевірити стаціонарність, оскільки це не тільки полегшує моделювання часових рядів, але й є основним припущенням у багатьох методах часових рядів. Зокрема, стаціонарність передбачається для широкого спектру методів прогнозування часових

рядів, включаючи авторегресійне ковзне середнє (ARMA), ARIMA та сезонне ARIMA (SARIMA).

Використаємо тест Дікі-Фуллера, щоб перевірити стабільність наших даних. Цей тест створить критичні значення та р-значення, що дозволить нам прийняти або відхилити нульову гіпотезу про відсутність стаціонарності. Якщо ми відкидаємо нульову гіпотезу, це означає, що ми приймаємо альтернативу, яка стверджує, що існує стаціонарність.

Ці значення дозволяють нам перевірити ступінь зміни теперішніх цінностей порівняно з минулими. Якщо в наборі даних немає стаціонарності, зміна поточних значень не спричинить істотної зміни минулих значень.

Таблиця 3.1.1

Тест Дікі-Фуллера для середньої ціни акції компанії Google LLC в залежності від періодів

Метрика	Значення (5 років)	Значення (3 роки)	Значення (2 роки)	Значення (1 рік)
Test Statistics	-0.37	-0.25	-0.48	-1.25
p-value	0.91	0.93	0.90	0.65
No. of lags used	3	1	3	1
Number of observation used	1254	751	499	250
critical value (1%)	-3.44	-3.44	-3.44	-3.46
critical value (5%)	-2.86	-2.87	-2.87	-2.87
critical value (10%)	-2.57	-2.57	-2.57	-2.57

Джерело: складено автором

З табл. 3.1.1 видно, що усі p-values значно більше за 0.05, а також test statistics більше за critical values у всіх періодах. Вищесказане свідчить про не стаціонарність ряду, тобто відсутність повторень в залежності від періоду.

Наступним етапом аналізу нашого ряду буде перевірка автокореляції. Перевірка даних часових рядів на автокореляцію в Python є ще однією важливою частиною аналітичного процесу. Це міра того, наскільки дані часових рядів у певний момент часу співвідносяться з минулими значеннями, що має величезне значення для багатьох галузей. Наприклад, якщо наші ціни на акції мають сильну автокореляцію, ми можемо припустити, що висока ціна сьогодні вказує на велику ймовірність того, що вона буде високою і завтра.

Таблиця 3.1.2

## Автокореляція середньої щоденної вартості акцій компанії Google LLC

Період автокореляції	Значення автокореляції
1 місяць	0.95
3 місяці	0.81
6 місяців	0.45
9 місяців	0.35

Джерело: складено автором

Можна спостерігати на табл. 2, що ряд значно пов'язаний з даними з лагом 30 днів. Однак слабка залежність між даними за 9 місяців. Це доводить наявність коротко або довго тривалих трендів в даних.

Декомпозиція є важливим інструментом для аналізу часових рядів, оскільки вона дозволяє розкрити складові частини ряду і виявити патерни, що лежать в їх основі. Декомпозиція дозволяє виділити трендову складову часового ряду, яка показує загальний напрямок змін у ряді протягом тривалого періоду. Це дозволяє аналізувати загальний тенденції в даних та розуміти, чи є зростання, спад або стійкість у часовому ряді. Декомпозиція також дозволяє виявити сезонні коливання або патерни у даних, які повторюються з певною періодичністю. Це важливо для розуміння циклічності або регулярності в даних та може бути корисним для прогнозування майбутніх значень. Декомпозиція допомагає виділити складову випадкової змінної, яка є

випадковою частиною даних, не підпорядкованою трендам або сезонним паттернам. Це може бути корисно для виділення шуму та виявлення непередбачуваних варіацій у даних. На рис. 9 відображені результати декомпозиції.

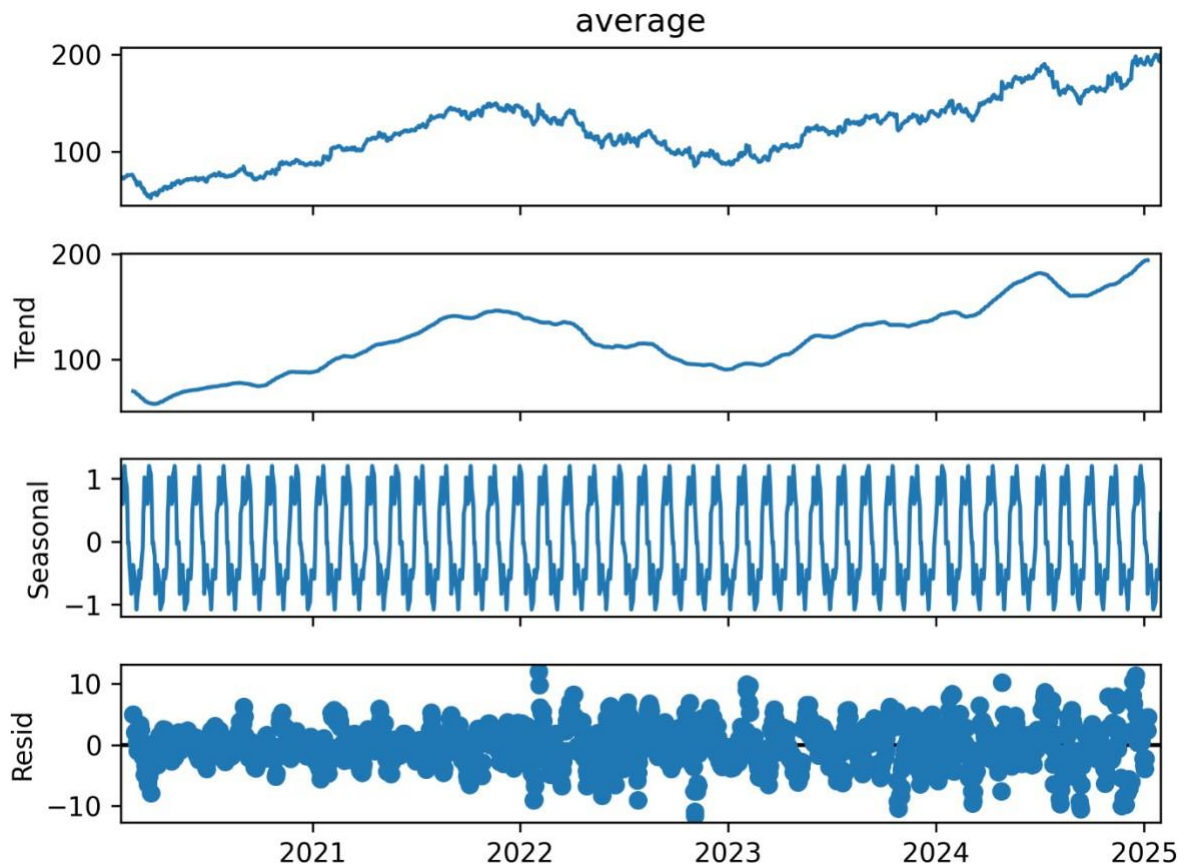


Рис. 3.1.2. Декомпозиція середньої щоденної вартості за допомогою логарифмування акцій компанії Google LLC

Джерело: розроблено автором

З рис. 9 видно зростаючий тренд, що свідчить про те, що ціна акції компанії значно зросла у 2025 році, порівнюючи з 2020 роком. Також бачимо сезонні коливання в рості та падінні вартості кожні півроку. Зростаючий тренд у ціні акцій свідчить про те, що компанія має позитивну динаміку розвитку, що може бути важливим фактором для прогнозування майбутніх значень. Економетричні моделі, наприклад, можуть використовувати лінійну регресію для моделювання цього тренду.

Сезонні коливання в рості та падінні цін акцій кожні півроку можуть бути викликані різними факторами, такими як сезонні зміни в попиті на продукцію

компанії або ринкові тенденції. Для ефективного прогнозування таких коливань можуть бути використані моделі часових рядів або моделі машинного навчання, які можуть враховувати ці сезонні ефекти.

Варіабельність цін акцій збільшується протягом часу. Це може бути важливим фактором для моделювання ризику та для прийняття рішень щодо інвестування. Моделі машинного навчання, такі як моделі глибокого навчання, можуть бути корисними для прогнозування варіабельності цін акцій.

Отже, для ефективного прогнозування цін акцій компанії Google LLC корисними будуть як економетричні моделі, так і моделі машинного навчання, які можуть враховувати зростаючий тренд, сезонні коливання та зміну варіабельності цін акцій.

### **3.2. Моделювання вартості акцій компанії GOOGLE за допомогою економетричних методів**

В даній роботі вартість акцій компанії Google LLC була змодельована за допомогою чотирьох методів згладжування: метод простого експоненціального згладжування, метод подвійного експоненціального згладжування, метод потрійного експоненціального згладжування (метод Холта-Вінтерса). Також було розглянуто метод ковзного середнього та метод зваженого ковзного середнього. Після цього було використано методи авторегресії та регресійні методи. Усі моделі були перевірені на чотирьох різних проміжках: 5 років, 3 роки, 2 роки та 1 рік. Та перевірено, на якому проміжку кожна із моделей справляється краще з моделюванням середньої ціни акції компанії Google LLC.

Спершу, було реалізовано модель експоненціального ковзного середнього. Результати в залежності від періоду можна побачити у табл. 3 та на рис. 8 – рис. 11. В якості параметра  $\alpha$  було взято три різних значення: 0.1, 0.6 та 0.9.

Результати впровадження моделі простого експоненціального ковзного середнього

Термін	$\alpha$	MSE	MAE	R2	MAPE
5 років	0.1	0.002	0.037	0.895	0.069
	0.6	0.000	0.017	0.976	0.032
	0.9	0.000	0.015	0.981	0.028
3 роки	0.1	0.004	0.047	0.881	0.093
	0.6	0.001	0.021	0.974	0.044
	0.9	0.001	0.019	0.981	0.038
2 роки	0.1	0.005	0.054	0.732	0.094
	0.6	0.001	0.024	0.948	0.044
	0.9	0.001	0.022	0.959	0.040
1 рік	0.1	0.005	0.059	0.558	0.069
	0.6	0.002	0.032	0.855	0.040
	0.9	0.001	0.029	0.883	0.036

Джерело: складено автором

У табл. 3.2.1 наведено результати впровадження моделі експоненціального ковзного середнього (ЕМА) для прогнозування середньої щоденної ціни акцій компанії Google за різні періоди та коефіцієнти згладжування. Параметри оцінки моделі включають середньоквадратичну похибку (MSE), середню абсолютну похибку (MAE), коефіцієнт детермінації (R2) та середню абсолютну процентну похибку (MAPE).

Результати показують, що модель ЕМА з більш високими значеннями (0.6 та 0.9) має кращу прогнозну точність за всіма періодами, що підтверджується нижчими значеннями MSE, MAE та MAPE, а також вищими значеннями R2. Це свідчить про те, що модель з більшою чутливістю до нових даних краще прогнозує середню щоденну ціну акцій Google.

Модель ЕМА можна використовувати для прогнозування короткострокових і довгострокових тенденцій на фондовому ринку. Інвестори та аналітики можуть застосовувати цей метод для визначення оптимальних моментів купівлі або продажу акцій. Значення  $\alpha$  можна налаштовувати в залежності від необхідного рівня чутливості до останніх змін ринкових даних.

Далі було реалізовано моделювання середньої щоденної вартості акцій компанії Google LLC методом подвійного експоненціального згладжування, де включається параметр, що відповідає за значення тренду. У табл. 3.2.2 наведені результати реалізації методу подвійного експоненціального згладжування в залежності від періоду та параметру тренда.

Таблиця 3.2.2

Результати впровадження моделі подвійного експоненціального згладжування

Термін	$(\alpha, \beta)$	MSE	MAE	R2	MAPE
5 років	(0.2, 0.1)	0.001	0.028	0.939	0.053
	(0.3, 0.5)	0.001	0.023	0.956	0.044
	(0.9, 0.9)	0.001	0.018	0.974	0.033
3 роки	(0.2, 0.1)	0.001	0.027	0.939	0.053
	(0.3, 0.5)	0.001	0.022	0.956	0.044
	(0.9, 0.9)	0.001	0.018	0.974	0.042
2 роки	(0.2, 0.1)	0.002	0.040	0.871	0.073
	(0.3, 0.5)	0.002	0.320	0.909	0.058
	(0.9, 0.9)	0.001	0.026	0.945	0.048
1 рік	(0.2, 0.1)	0.003	0.044	0.724	0.054
	(0.3, 0.5)	0.003	0.046	0.713	0.057
	(0.9, 0.9)	0.002	0.036	0.834	0.046

Джерело: складено автором

Результати свідчать про те, що модель подвійного експоненціального згладжування (DEMA) показує досить високу точність у прогнозуванні ціни акцій Google. Значення  $R^2$  набагато більше за 0.9, що вказує на високу здатність моделі пояснювати варіацію в даних. Крім того, низькі значення MAPE свідчать про добру точність прогнозів.

Модель DEMА може бути використана для прогнозування цін акцій Google на різні періоди часу. Її можна застосовувати як у короткострокових, так і у довгострокових аналізах ринку. Відповідно до вибраних параметрів модель буде здатна підлаштовуватися під різні ринкові умови.

Застосування моделі DEMА може допомогти інвесторам та аналітикам приймати обґрунтовані рішення щодо купівлі, утримання або продажу акцій компанії Google.

Високі значення коефіцієнта детермінації та низькі значення середньої абсолютної процентної похибки свідчать про ефективність моделі в прогнозуванні цінових тенденцій.

Порівняно з іншими методами прогнозування, такими як просте експоненціального згладжування або нейронні мережі, модель ДЕМА показує схожу або навіть кращу точність.

Після цього було реалізовано метода Холта-Вінтерса, де включався третій параметр сезонності. Результати впровадження моделі потрійного експоненціального згладжування можна побачити у табл. 3.2.3.

Таблиця 3.2.3

## Результати впровадження моделі Холта-Вінтерса

Термін	Параметри	MSE	MAE	R2	MAPE
5 років	[10, 'add', 'add']	0.000	0.014	0.982	0.027
3 роки	[10, 'add', 'add']	0.000	0.018	0.982	0.036
2 роки	[10, 'add', 'add']	0.000	0.020	0.963	0.038
1 рік	[10, 'add', 'add']	0.001	0.026	0.902	0.033

Джерело: розрахунки автора

У таблиці наведено результати впровадження методу Холта-Вінтерса з третім параметром сезонності для прогнозування середньої щоденної ціни акцій компанії Google за різні періоди.

Для п'ятирічного, трьохрічного, дворічного та однорічного періодів були використані однакові параметри [10, 'add', 'add'], що вказує на використання адитивної моделі з трендом та сезонністю.

Метод Холта-Вінтерса з третім параметром сезонності враховує як тренд, так і сезонні коливання у даних. Це дозволяє моделі адаптуватися до сезонних змін у часовому ряді, що може бути важливим для прогнозування цін акцій, особливо в галузі фінансів. Однією з ключових переваг моделі Холта-Вінтерса з третім параметром сезонності є її здатність ефективно управляти сезонними коливаннями у

часових рядах. Це особливо важливо в контексті прогнозування цін акцій, де сезонні впливи можуть мати значний ефект на цінові рухи.

Результати показують дуже низькі значення MSE та MAE, а також високі значення R2, що свідчить про дуже точні прогнози моделі. Це може бути корисним для інвесторів та фінансових аналітиків для управління ризиками та прийняття обґрунтованих рішень на ринку.

Порівняно з іншими методами прогнозування, модель Холта-Вінтерса з сезонністю може бути більш ефективною в управлінні сезонними коливаннями у часових рядах. Вона зазвичай демонструє кращі результати, коли є чітко виражений тренд та сезонні зміни в даних, як це часто буває на фінансових ринках. Хоча модель Холта-Вінтерса з третім параметром сезонності може бути більш складною для реалізації і вимагати більше обчислювальних ресурсів, вона може надати кращі прогнози відносно до ЕМА або ДЕМА, особливо у випадках, коли присутня явна сезонність.

Модель Холта-Вінтерса з третім параметром сезонності є потужним інструментом для прогнозування цін акцій компанії Google. Вона дозволяє враховувати як тренд, так і сезонність у даних, що призводить до точних прогнозів і може бути корисним для прийняття рішень на фінансових ринках.

Було також реалізовано метод ковзного середнього з різними кроками. Результати в залежності від періоду можна спостерігати у табл. 6

Таблиця 3.2.4

## Результати впровадження методу ковзного середнього

Термін	Вікно	MSE	MAE	R2	MAPE
5 років	30	0.044	0.191	-1.197	0.343
	10	0.055	0.213	-1.704	0.383
	5	0.058	0.218	-1.859	0.394
3 роки	30	0.109	0.307	-2.646	0.642
	10	0.109	0.303	-2.676	0.656
	5	0.112	0.301	-2.712	0.657
2 роки	30	0.074	0.211	-2.882	0.412
	10	0.068	0.203	-2.557	0.414
	5	0.067	0.199	-2.521	0.413

1 рік	30	0.030	0.162	-1.501	0.201
	10	0.037	0.169	-2.071	0.212
	5	0.040	0.172	-2.345	0.217

Джерело: складено автором

Табл. 3.2.4 представляє результати впровадження методу ковзного середнього для прогнозування середньої щоденної ціни акцій компанії Google за різні періоди та різні розміри вікон (30, 10, 5).

Для п'ятирічного, трирічного, дворічного та однорічного періодів було розглянуто вікна розміром 30, 10 та 5.

Результати показують, що зменшення розміру вікна призводить до збільшення значень MSE, MAE та MAPE, а також до зниження значень R2. Це може свідчити про те, що зменшення вікна призводить до втрати істотних даних та погіршення точності прогнозування. У порівнянні з іншими методами прогнозування, такими як модель Холта-Вінтерса або метод подвійного експоненціального згладжування, метод ковзного середнього може виявитися менш ефективним у випадку складних часових рядів зі значними коливаннями. Величина вікна може впливати на точність прогнозів, і великі вікна можуть призводити до згладжування важливих даних та погіршення точності.

Метод ковзного середнього є одним із найпростіших методів прогнозування, але його ефективність може залежати від величини вікна та складності часового ряду. У випадку складних часових рядів, таких як ціни акцій, із застосуванням методу ковзного середнього слід бути обережними та уважними до вибору відповідного розміру вікна для досягнення найкращих результатів прогнозування.

Далі було реалізовано три методи авторегресії. Спочатку було обрано оптимальні параметри за допомогою тесту Дікі-Фулера та критерію Акайке, і здійснено прогнозування майбутніх цін. Результатами даного методу було виявлено наступні параметри: (2, 2, 1). Результати впровадження цих моделей видно у табл. 7.

## Реалізація моделей авторегресії

Термін	Параметри	MSE	MAE	R2	MAPE
5 років	(2, 0, 0)	0.000	0.014	0.983	0.027
	(2, 0, 1)	0.000	0.014	0.982	0.026
	(2, 2, 1)	0.000	0.014	0.982	0.026
3 роки	(2, 0, 0)	0.001	0.017	0.982	0.035
	(2, 0, 1)	0.001	0.017	0.982	0.035
	(2, 2, 1)	0.001	0.017	0.982	0.035
2 роки	(2, 0, 0)	0.001	0.020	0.961	0.037
	(2, 0, 1)	0.001	0.020	0.961	0.038
	(2, 2, 1)	0.001	0.020	0.961	0.038
1 рік	(2, 0, 0)	0.001	0.026	0.890	0.033
	(2, 0, 1)	0.001	0.027	0.889	0.033
	(2, 2, 1)	0.001	0.026	0.890	0.033

Джерело: розрахунки автора

Табл. 3.2.5 представляє результати впровадження трьох моделей авторегресії для прогнозування середньої щоденної ціни акцій компанії Google за різні періоди. Моделі авторегресії використовують попередні значення цільової змінної для прогнозування майбутніх значень. Параметри (p, d, q) відповідають порядку авторегресії, порядку диференціювання та порядку ковзного середнього.

У даному випадку, підбором оптимальних параметрів за допомогою тесту Дікі-Фулера та критерію Акайке, було обрано (2, 2, 1) як оптимальний набір параметрів для всіх моделей.

У порівнянні з іншими моделями, такими як метод ковзного середнього чи модель Холта-Вінтерса, моделі авторегресії показали кращу точність прогнозування для розглянутих періодів.

Отже, моделі авторегресії з оптимальними параметрами (2, 2, 1) є ефективними для прогнозування цін акцій компанії Google на різні періоди. Вони демонструють низьку помилку прогнозування та високу точність прогнозів, що може бути корисним для інвесторів та фінансових аналітиків у прийнятті рішень на фондовому ринку.

Далі було розглянуто регресійні підходи. За допомогою відповідних бібліотек Python було реалізовано метод простої лінійної регресії (в якості незалежної змінної була середня щоденна ціна з одним кроком назад), метод множинної лінійної регресії (в якості незалежних змін були взяті: середня щоденна ціна з одним кроком назад, ціна відкриття, найменші та найбільші щоденні ціни, та щоденна зміна вартості) та метод нелінійної регресії, де в якості параметрів використовувалися поліноміальні ознаки. Дані моделі мають гарні результати, про це свідчить табл. 3.2.6.

Таблиця 3.2.6

Впровадження регресійних підходів для моделювання середньої вартості акцій компанії Google

Термін	Модель	MSE	MAE	R2	MAPE
5 років	Проста лінійна регресія	0.000	0.015	0.982	0.027
	Множинна регресія	0.000	0.000	0.996	0.000
	Нелінійна регресія	0.000	0.000	0.996	0.000
3 роки	Проста лінійна регресія	0.000	0.018	0.982	0.037
	Множинна регресія	0.000	0.000	0.996	0.000
	Нелінійна регресія	0.000	0.000	0.996	0.000
2 роки	Проста лінійна регресія	0.001	0.021	0.960	0.039
	Множинна регресія	0.000	0.000	0.991	0.000
	Нелінійна регресія	0.000	0.000	0.991	0.000
1 рік	Проста лінійна регресія	0.001	0.029	0.876	0.036
	Множинна регресія	0.000	0.000	0.990	0.000
	Нелінійна регресія	0.000	0.000	0.990	0.000

Джерело: складено автором

У даному випадку було розглянуто три різних підходи до регресійного моделювання: проста лінійна регресія, множинна лінійна регресія та нелінійна регресія, застосовані для прогнозування середньої вартості акцій компанії Google на різних періодах часу.

Модель простої лінійної регресії використовує лише один предиктор (середню щоденну ціну з одним кроком назад) для прогнозування цільової змінної (середня вартість акцій). Цей підхід показав хороші результати, з низькими значеннями MSE та MAE, а також високим значенням коефіцієнта детермінації R2.

Модель множинної лінійної регресії використовує кілька предикторів (середню щоденну ціну з одним кроком назад, ціну відкриття, найменшу та найбільшу щоденні ціни, та щоденну зміну вартості) для прогнозування цільової змінної. Цей підхід також показав вражаючі результати з дуже низькими значеннями MSE та MAE, а також майже ідеальним коефіцієнтом детермінації  $R^2$ , що свідчить про дуже високу точність моделі.

Модель нелінійної регресії використовує поліноміальні ознаки для побудови моделі, що дозволяє моделі адаптуватися до нелінійних залежностей між предикторами та цільовою змінною. Так само, як і модель множинної лінійної регресії, цей підхід показав вражаючі результати з дуже низькими значеннями MSE та MAE, а також майже ідеальним коефіцієнтом детермінації  $R^2$ .

На різних періодах часу різні підходи можуть показувати кращі результати. Наприклад, проста лінійна регресія може бути ефективною для короткострокового прогнозування, тоді як множинна лінійна регресія та нелінійна регресія можуть бути більш ефективними для довгострокових прогнозів або в умовах складних залежностей між змінними.

Порівнюючи з попередніми моделями, такими як експоненціальне згладжування або авторегресія, регресійні підходи показали схожу або кращу точність прогнозування. Вони можуть бути особливо корисними у випадках, коли присутні складні залежності між змінними та коли необхідно врахувати більш широкий спектр факторів для прогнозування.

### **3.3. Моделювання вартості акцій компанії GOOGLE за допомогою методів машинного навчання**

В даній роботі було розглянуто декілька моделей машинного навчання для моделювання щоденної середньої вартості акцій компанії Google LLC. Кожна із моделей була навчена на різних проміжках (5, 3, та 2 роки), оскільки для навчання моделей машинного навчання необхідно використати якомога більше даних.

Зауважимо, що на рік не будувалося, оскільки така кількість даних не є достатньою для моделі.

У даному випадку було розглянуто використання моделі довго-коротко строкової пам'яті (LSTM) для прогнозування середньої щоденної ціни акцій компанії Google. Модель LSTM є потужним інструментом для обробки часових рядів, здатним уловлювати довгострокові залежності в даних.

Таблиця 3.3.1

## Результати впровадження моделі довго-коротко строкової пам'яті

Термін	Кількість епох	MSE	MAE	R2	MAPE
5 років	1	0.019	0.115	0.997	0.241
	10	0.020	0.121	0.998	0.254
	20	0.024	0.133	0.999	0.278
3 роки	1	0.050	0.198	0.996	0.558
	10	0.074	0.247	0.996	0.681
	20	0.057	0.211	0.997	0.595
2 роки	1	0.051	0.210	0.995	0.546
	10	0.038	0.167	0.995	0.454
	20	0.058	0.219	0.995	0.577

Джерело: розрахунки автора

Модель LSTM навчається на часових рядах, оптимізуючи свої ваги для мінімізації помилки прогнозу. Після кожної епохи модель проходить через весь навчальний набір даних, оновлюючи свої параметри. Збільшення кількості епох зазвичай покращує здатність моделі вловлювати складні патерни в даних, але також може призвести до перенавчання, особливо якщо дані містять шум або якщо модель стає занадто складною.

У порівнянні з іншими моделями, такими як проста лінійна регресія, множинна лінійна регресія, нелінійна регресія та методи експоненціального згладжування, модель LSTM показала наступні особливості:

- **Висока точність:** Модель LSTM демонструє дуже високі значення R2 (приблизно 0.998-0.999) для довгих періодів (п'ять років), що свідчить про те, що модель дуже добре підходить для прогнозування довгострокових тенденцій.
- **Складність навчання:** Хоча LSTM модель показала хороші результати, вона потребує більш тривалого навчання (більше епох) для досягнення оптимальних результатів.
- **Чутливість до параметрів:** Модель може бути чутливою до кількості епох навчання, що може призвести до коливань у точності прогнозування, особливо для коротших періодів.

Модель LSTM є потужним інструментом для прогнозування часових рядів, особливо для довгострокових прогнозів, завдяки своїй здатності враховувати як короткострокові, так і довгострокові залежності в даних. Однак, вона потребує ретельної настройки та тривалого навчання для досягнення оптимальних результатів. У порівнянні з більш простими моделями, такими як регресійні підходи або методи експоненціального згладжування, LSTM може запропонувати кращу точність, але за рахунок більшої складності та обчислювальних ресурсів.

Далі було реалізовано модель багатошарового парцептрона Румельхарта на різні періоди з кількістю епох 50 на кожен із них. Результати впровадження даної моделі можна побачити у табл. 10. Зауважимо, що при збільшенні кількості епох модель буде перенавчатися

Таблиця 3.3.2

Результати впровадження моделі багатошарового парцептрона Румельхарта

Термін	Кількість епох	MSE	MAE	R2	MAPE
5 років	50	0.022	0.128	0.888	0.267
3 роки	50	0.049	0.193	0.654	0.551
2 роки	50	0.050	0.204	0.654	0.538

Джерело: складено автором

Модель багатошарового перцептрона (MLP) є одним з основних видів штучних нейронних мереж, призначених для класифікаційних і регресійних задач. Ця модель

складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Вона використовує функцію активації для кожного нейрона, що дозволяє моделі навчатися нелінійним залежностям у даних.

Модель MLP показала наступні результати для різних періодів:

- П'ятирічний період: Значення MSE і MAE вказують на те, що модель здатна уловлювати загальні тенденції, однак  $R^2$  негативний, що свідчить про те, що модель не підходить для даного терміну або що вона може бути перенавченою або недонавченою. Значення MAPE вказує на середню абсолютну процентну похибку, яка також є досить високою.
- Трьохрічний період: Подібні результати спостерігаються і тут, з негативним значенням  $R^2$ , що свідчить про низьку точність моделі. Значення MAE і MAPE свідчать про високий рівень похибок.
- Дворічний період: Результати для цього періоду також показують негативний  $R^2$  і високі значення MAE та MAPE, що вказує на значну похибку моделі.

Невисокі значення  $R^2$  вказують на те, що модель може бути перенавченою або недонавченою. Це може бути через недостатню кількість епох навчання або занадто складну структуру моделі для даних, які використовувалися. Хоча кількість епох (50) здається достатньою, для деяких даних цього може бути недостатньо, щоб модель могла належним чином навчитися залежностям у даних. Акції компанії Google можуть мати складні залежності, які важко уловити за допомогою простої структури MLP, особливо з урахуванням сезонності, трендів та інших нелінійних факторів.

Модель LSTM показала значно кращі результати, ніж MLP, з високими значеннями  $R^2$  та низькими похибками, що свідчить про її здатність уловлювати довгострокові залежності в даних.

Модель багат шарового перцептрона (MLP) може бути ефективною для моделювання складних нелінійних залежностей, однак у даному випадку вона показала значно гірші результати порівняно з іншими моделями. Це може бути пов'язано з проблемами перенавчання або недонавчання, а також з недостатньою кількістю епох навчання. Для покращення результатів може бути необхідно

збільшити кількість епох або використовувати більш складну архітектуру нейронної мережі.

Також було впроваджено модель випадкового лісу. Результати даної моделі можна побачити у табл. 3.3.3.

Таблиця 3.3.3

## Результати впровадження моделі випадкового лісу

Термін	MSE	MAE	R2	MAPE
5 років	0.012	0.569	0.999	0.241
3 роки	0.015	0.651	0.998	0.245
2 роки	0.017	0.752	0.998	0.247

Джерело: складено автором

Випадковий ліс (Random Forest) є потужним методом ансамблевого навчання, який поєднує прогнози численних дерев рішень для покращення загальної точності та стабільності. Цей метод є особливо ефективним у випадках, коли є багато змінних і потенційно складні взаємозв'язки між ними.

Модель випадкового лісу показала найкращі результати серед усіх розглянутих моделей для прогнозування середньої щоденної ціни акцій компанії Google. Значення R2 близьке до 1 (0.999 для п'ятирічного періоду і 0.998 для трьохрічного та дворічного періодів) свідчить про те, що модель випадкового лісу дуже точно пояснює варіації в даних. Це означає, що майже вся варіація в середній щоденній ціні акцій компанії Google пояснюється моделлю. Низьке значення середньоквадратичної похибки (MSE) вказує на те, що модель робить дуже точні передбачення з мінімальними відхиленнями від фактичних значень. Середня абсолютна похибка (MAE) показує, що середнє абсолютне відхилення передбачуваних значень від фактичних є досить низьким, що також підтверджує високу точність моделі. Хоча середня абсолютна процентна похибка (MAPE) трохи вища, вона все одно знаходиться на прийнятному рівні, що вказує на відносно низькі відносні похибки.

Акції компанії Google характеризуються складною динамікою і високою варіативністю. Модель випадкового лісу, завдяки своїй здатності вловлювати

нелінійні залежності та взаємодії між змінними, є особливо ефективною для таких типів даних. Випадковий ліс створює безліч дерев рішень на основі випадкових підмножин даних і ознак, що зменшує ризик перенавчання і підвищує здатність моделі узагальнювати дані. Це особливо важливо для фінансових даних, які можуть мати шум і випадкові коливання. Модель випадкового лісу показала стабільно високі результати на всіх розглянутих періодах (2, 3 і 5 років). Це свідчить про її здатність ефективно працювати на різних вибірках даних і з різними обсягами історичних даних. Випадковий ліс може враховувати безліч змінних, що впливають на ціну акцій, включаючи історичні ціни, обсяги торгів, макроекономічні індикатори тощо. Це дозволяє моделі краще розуміти і передбачати складні ринкові динаміки.

Модель випадкового лісу продемонструвала найкращі результати серед усіх розглянутих моделей для прогнозування середньої щоденної ціни акцій компанії Google завдяки своїй здатності вловлювати складні нелінійні взаємозв'язки, високу узагальнюючу здатність і стабільність результатів на різних періодах. Це робить її потужним інструментом для аналізу і прогнозування фінансових даних у мінливому ринковому середовищі.

Також було впроваджено модель рекурсивного усунення ознак на основі попередньої моделі випадкового лісу. Дана модель дала значно кращі результати, прибравши ті значення, що негативно впливали на дані. Результати впровадження даної моделі можна побачити у табл. 3.3.4

Таблиця 3.3.4

Результати впровадження методу рекурсивного усунення ознак на основі моделі випадкового лісу

Термін	MSE	MAE	R2	MAPE
5 років	0.009	0.569	0.999	0.241
3 роки	0.010	0.651	0.998	0.245
2 роки	0.010	0.752	0.998	0.247

Джерело: складено автором

Рекурсивне усунення ознак (Recursive Feature Elimination, RFE) – це метод, який використовується для вибору найбільш значущих змінних для моделі. Цей метод ітеративно тренує модель, усуваючи найменш значущі ознаки на кожному етапі, поки не залишиться набір найбільш важливих ознак. Поєднання цього методу з моделлю випадкового лісу дозволяє не тільки підвищити точність прогнозування, але й зробити модель більш інтерпретованою.

Метод рекурсивного усунення ознак на основі моделі випадкового лісу показав ще кращі результати порівняно з базовою моделлю випадкового лісу. Розглянемо детальніше, чому саме цей підхід виявився найбільш ефективним. Зниження значень MSE для всіх періодів свідчить про те, що модель стала більш точною в своїх прогнозах. Наприклад, для п'ятирічного періоду MSE зменшилася з 0.012 до 0.009. Значення R2 залишилися на дуже високому рівні (0.999 для п'ятирічного періоду і 0.998 для трьохрічного та дворічного періодів), що вказує на майже повну відповідність передбачених значень фактичним. Метод RFE допоміг виділити найбільш значущі ознаки, що сприяло покращенню загальної продуктивності моделі. Усунення менш важливих ознак зменшило шум в даних і дозволило моделі більш точно вловлювати основні патерни.

Використання RFE допомогло зменшити кількість незначущих або корельованих ознак, що могло б вносити шум у модель. Це дозволило моделі фокусуватися на найбільш інформативних змінних. Модель випадкового лісу в поєднанні з RFE зменшує ризик перенавчання, оскільки вона використовує тільки найважливіші ознаки для побудови своїх дерев рішень, що сприяє кращому узагальненню на нових даних. Використання меншої кількості ознак знижує обчислювальні витрати і час навчання моделі, що є важливим фактором для обробки великих обсягів даних.

Метод рекурсивного усунення ознак на основі моделі випадкового лісу продемонстрував найкращі результати для прогнозування середньої щоденної ціни акцій компанії Google. Це пов'язано з його здатністю оптимально вибирати найбільш значущі ознаки, зменшувати шум в даних і покращувати загальну точність моделі.

Високі значення  $R^2$ , низькі значення MSE і MAE свідчать про високу ефективність і надійність цього підходу.

### Висновки до розділу 3

Різні методи прогнозування мають свої переваги та обмеження в залежності від специфіки завдання та доступних даних. Найкращі результати показала модель випадкового лісу з використанням методу рекурсивного усунення ознак (RFE). Ця модель забезпечила найвищу точність прогнозування, про що свідчать низькі значення середньоквадратичної помилки (MSE), середньої абсолютної помилки (MAE) та високі значення коефіцієнта детермінації ( $R^2$ ). Висока ефективність цієї моделі зумовлена здатністю оптимально вибирати найбільш значущі ознаки, що зменшує шум у даних та покращує загальну точність прогнозування.

Прості та експоненціальні згладжування, хоча й легкі у використанні та інтерпретації, не можуть враховувати складні взаємозв'язки між змінними. Вони виявилися менш точними у порівнянні з моделями на основі випадкового лісу та нейронних мереж, особливо при прогнозуванні на тривалі періоди. Регресійні моделі, такі як проста та множинна лінійна регресія, показали гарні результати, але також мають обмеження у врахуванні нелінійних залежностей та взаємодій між змінними. Це робить їх менш ефективними у порівнянні з моделями машинного навчання.

Моделі нейронних мереж, зокрема модель довго-коротко строкової пам'яті (LSTM), продемонстрували високу точність у короткостроковому прогнозуванні, але потребують більше даних та часу на навчання. Вони ризикують перенавчатися, особливо при недостатній кількості даних або при невірно обраних гіперпараметрах. Модель багатошарового перцептронну, хоча й ефективна, також виявилася менш точною у порівнянні з випадковим лісом з RFE.

Таким чином, найкращі результати прогнозування середньої щоденної ціни акцій компанії Google показала модель випадкового лісу з рекурсивним усуненням ознак. Ця модель варто використовувати, коли важливо досягти найвищої точності

прогнозування та є можливість оптимізувати вибір ознак. Просте та експоненціальне згладжування може бути корисним для попереднього аналізу та вивчення основних тенденцій, особливо коли даних недостатньо для складніших моделей. Регресійні моделі доцільно застосовувати для базового прогнозування та аналізу лінійних залежностей. Нейронні мережі підходять для задач з великою кількістю даних і складними нелінійними залежностями, але потребують ретельного налаштування та значних обчислювальних ресурсів.

Отже, вибір моделі залежить від специфіки завдання, доступності даних та необхідного рівня точності. Модель випадкового лісу з RFE наразі демонструє найвищу ефективність для прогнозування цін акцій компанії Google, що обумовлює її пріоритетне використання у відповідних задачах.

## ВИСНОВКИ

У роботі проведено порівняльний аналіз трьох підходів до прогнозування середньоденної ціни акцій Google: методів згладжування, економетричних моделей та штучних нейронних мереж з використанням алгоритмів штучного інтелекту. Дослідження базувалося на аналізі часових рядів, що містять інформацію про середньоденну ціну акцій Google, з використанням відповідних бібліотек мови програмування Python.

Виконані завдання дозволили досягти поставленої мети дослідження. Спочатку було проведено детальне вивчення компанії Google і технологій її роботи. Це дозволило зрозуміти основні чинники, що впливають на коливання курсу акцій компанії, та врахувати ці фактори під час моделювання.

По-друге, було проведено аналіз статистичних даних, який включав попередню обробку та вивчення поведінки часових рядів. Це дозволило забезпечити високу якість даних для подальшого моделювання та прогнозування.

По-третє, проведено ґрунтовне дослідження архітектури методів згладжування, економетричних моделей і методів штучних нейронних мереж. Кожен із підходів було детально розглянуто, проаналізовано його переваги та недоліки, а також проведено емпіричне тестування на історичних даних цін акцій Google.

Результати порівняльного аналізу показали, що моделі на основі штучних нейронних мереж і алгоритмів машинного навчання, зокрема модель випадкового лісу з використанням методу рекурсивного усунення ознак (RFE), продемонстрували найвищу точність прогнозування. Ці моделі забезпечили найнижчу середню квадратичну похибку (MSE) і середню абсолютну похибку (MAE), а також найвищі значення коефіцієнта детермінації ( $R^2$ ), що вказує на їх здатність ефективно моделювати складні зв'язки в даних.

Методи згладжування, хоча й прості у використанні, виявилися менш точними порівняно зі складнішими моделями через їх обмежену здатність враховувати нелінійні залежності. Економетричні моделі, такі як моделі авторегресії, працювали краще, але все ще були менш точними, ніж моделі машинного навчання.

Таким чином, найкращим методом моделювання середньоденної ціни акцій Google була модель випадкового лісу з рекурсивним усуненням ознак. Такий підхід забезпечує високу точність і надійність прогнозування, що є критично важливим для прийняття зважених інвестиційних рішень. Інші методи можуть бути корисними в певних умовах або для початкового аналізу даних, але сучасні методи машинного навчання рекомендуються для найбільш точного прогнозування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google [Електронний ресурс].  
<https://uk.wikipedia.org/wiki/Google>
2. Ростислав Камерістов. (14 квітня 2018). Старший брат слідкує за тобою: що знає про тебе Google.
3. Vise, D. A. (2005). The Google Story: Inside the Hottest Business, Media, and Technology Success of Our Time. Delta, 1-336.
4. Mayer, M. (2014). Google AdWords: A Historical Review and Current Practices. Journal of Advertising Research, 54(2), 1-15.
5. Офіси GOOGLE [Електронний ресурс].  
<https://www.google.com/intl/ua/about/locations/>
6. Про компанію GOOGLE [Електронний ресурс].  
<https://www.google.com/intl/ua/about/>
7. Про продукти GOOGLE [Електронний ресурс].  
<https://www.google.com/intl/ua/about/products/>
8. Google – вчора, сьогодні, завтра [Електронний ресурс].  
<http://www.googlers.ua/videnie/1.php>
9. Акції Google за 10 років зросли в 13 разів [Електронний ресурс].  
<https://www.epravda.com.ua/news/2014/08/20/484330/>
10. Yahoo finance [Електронний ресурс].  
[https://finance.yahoo.com/quote/GOOG/history/?guccounter=1&guce\\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\\_referrer\\_sig=AQAAANKWMnGPxa6\\_129VBEIY48L9RTKBP8sJejBIK3sFGgBPjrnBvuhE-6V-zgYtv-NHwUD-LKji7HjPY3eYzR9h\\_zGTiXJcKm6Pt7De85zyrSb9UAkl9sVeZqi2F\\_VC2pW](https://finance.yahoo.com/quote/GOOG/history/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAANKWMnGPxa6_129VBEIY48L9RTKBP8sJejBIK3sFGgBPjrnBvuhE-6V-zgYtv-NHwUD-LKji7HjPY3eYzR9h_zGTiXJcKm6Pt7De85zyrSb9UAkl9sVeZqi2F_VC2pW)
11. If You Had Invested Right After Google's IPO [Електронний ресурс].  
<https://www.investopedia.com/articles/active-trading/081315/if-you-would-have-invested-right-after-googles-ipo.asp>
12. Yijie Zhang (Січень, 2024). The comprehensive analysis of Googles stock using ARIMA model (ст. 1 - 2).

13. Afees Salisu, Ahamuefula Ephiraim Ogbonna, Idris A. Adedrian (Липень, 2020). Stock-induced Google trends and predictability of sectoral stock returns (ст. 1).
14. Rolling Window Regression: a Simple Approach for Time Series Next value Predictions [Электронный ресурс].  
<https://medium.com/making-sense-of-data/time-series-next-valueprediction-using-regression-over-a-rolling-window-228f0acae363>
15. Financial Forecasting Preparing a prediction of the future [Электронный ресурс].  
<https://corporatefinanceinstitute.com/resources/knowledge/modeling/financialforecasting-guide/>
16. Predicting Stock Returns Using Financial Statement Information [Электронный ресурс].  
<https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5957.t01-1-00205>
17. Engle R. F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*. 1982. Т. 50, № 4. С. 987. [Электронный ресурс].  
<https://doi.org/10.2307/1912773>
18. Cuthbertson K. *Introductory econometrics for finance*, Chris Brooks, Cambridge University Press, Cambridge, 2002. *International Journal of Finance & Economics*. 2004. Т. 9, № 1. С. 82–83. [Электронный ресурс].  
<https://doi.org/10.1002/ijfe.223>
19. How-To Guide on Exploratory Data Analysis for Time Series Data [Электронный ресурс].  
<https://medium.com/analytics-vidhya/how-to-guide-on-exploratory-dataanalysis-for-time-series-data-34250ff1d04f>
20. Hyper parameter optimization in Python [Электронный ресурс].  
<https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-hyperopt-5f661db91324>

21. Time Series Analysis Tutorial Using Financial Data URL: <https://towardsdatascience.com/time-series-analysis-tutorial-using-financial-data4d1b846489f9>
22. Financial, Economic and Alternative Data. Quandl URL: <https://www.quandl.com/>
23. Python Release Python 3.9.2 URL: <https://www.python.org/downloads/release/python-392/>
24. Financial Forecasting Preparing a prediction of the future URL: <https://corporatefinanceinstitute.com/resources/knowledge/modeling/financialforecasting-guide/>
25. Predicting the Stock Market with Machine Learning. Introduction. URL: <https://towardsdatascience.com/predicting-the-stock-market-with-machine-learningintroduction-310cd6069ffa>
26. Predicting Stock Returns Using Financial Statement Information URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5957.t01-1-00205>
27. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005-2019 URL: <https://arxiv.org/pdf/1911.13288.pdf>
28. Introduction to interactive Time Series Visualizations with Plotly in Python URL: <https://towardsdatascience.com/introduction-to-interactive-time-seriesvisualizations-with-plotly-in-python-d3219eb7a7af>
29. Machine Learning models for 100% better return in Algo-trading URL: <https://medium.datadriveninvestor.com/machine-learning-models-for-marketbeating-trading-strategies-c773ba46db66>
30. Hyperparameter optimization in Python URL: <https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-hyperopt-5f661db91324>
31. Error Analysis to Your Rescue! URL: <https://towardsdatascience.com/erroranalysis-to-your-rescue-773b401380ef>

32. How to Improve my ML algorithm? Lessons from Andrew Ng's experience - I URL: <https://medium.com/all-things-data-science/how-to-improve-my-mlalgorithm-lessons-from-andrew-ngs-experience-i-551ca1a32634>
33. Regularization: Ridge, Lasso and Elastic Net URL: <https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net>
34. Deep Learning in Neural Networks: An Overview [Електроннийресурс] – URL: <https://arxiv.org/pdf/1404.7828.pdf>
35. Вибір статистичних методів прогнозування виробництва продукції <https://lib.chmnu.edu.ua/pdf/naukpraci/economy/2005/38-25-14.pdf>
36. Mathematics Behind Random forest and XGBoost URL: <https://medium.com/analytics-vidhya/mathematics-behind-random-forest-andxgboost-ea8596657275>
37. Cuthbertson K. Introductory econometrics for finance, Chris Brooks, Cambridge University Press, Cambridge, 2002. International Journal of Finance & Economics. 2004. Т. 9, № 1. С. 82–83. URL: <https://doi.org/10.1002/ijfe.223>
38. NumPy Tutorial. URL: <https://realpython.com/numpy-tutorial/>
39. Mean Absolute Error. URL: <https://www.statisticshowto.com/absoluteerror/>
40. Cuthbertson K. Introductory econometrics for finance, Chris Brooks, Cambridge University Press, Cambridge, 2002. International Journal of Finance & Economics. 2004. Т. 9, № 1. С. 82–83. URL: <https://doi.org/10.1002/ijfe.223>
41. Офіційний сайт фондової біржі «Українська біржа». URL: <http://www.ux.ua>
42. How-To Guide on Exploratory Data Analysis for Time Series Data URL: <https://medium.com/analytics-vidhya/how-to-guide-on-exploratory-dataanalysis-for-time-series-data-34250ff1d04f>
43. Hyper parameter optimization in Python URL: <https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-hyperopt-5f661db91324>
44. Error Analysis to Tour Rescue! URL: <https://towardsdatascience.com/erroranalysis-to-your-rescue-773b401380ef>

45. How to Improve my ML algorithm? Lessons from Andrew Ng's experience - II URL: <https://medium.com/all-things-data-science/how-to-improve-my-mlalgorithm-lessons-from-andrew-ngs-experience-ii-f66926926f88>
46. МОДЕЛЮВАННЯ КРЕДИТНОГО РИЗИКУ З ВИКОРИСТАННЯМ ГІБРИДНОЇ МОДЕЛІ RNN-LSTM ДЛЯ ЦИФРОВИХ ФІНАНСОВИХ УСТАНОВ. URL: <https://www.nayka.com.ua/index.php/ee/article/view/6224>
47. Черняк О.І та Ставицький А.В. Динамічна економетрика.

## ДОДАТОК А. ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ МОДЕЛЕЙ

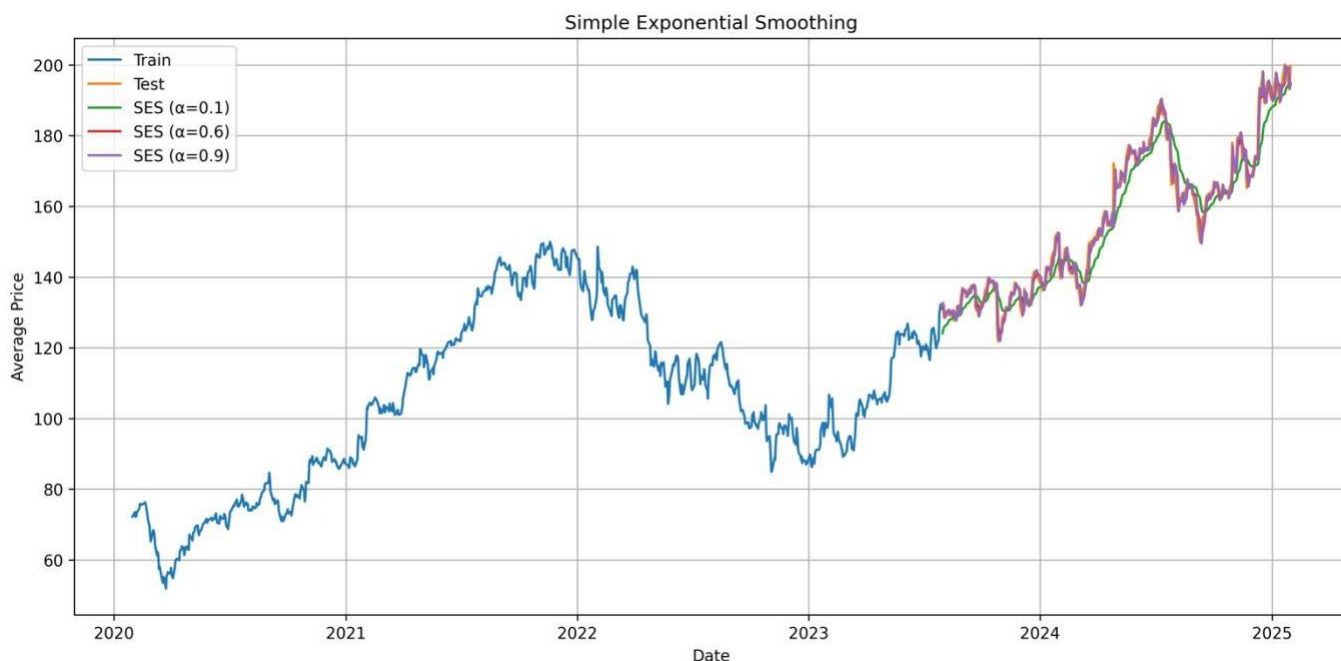


Рис. 1. Реалізація методу простого експоненціального згладжування на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: складено автором

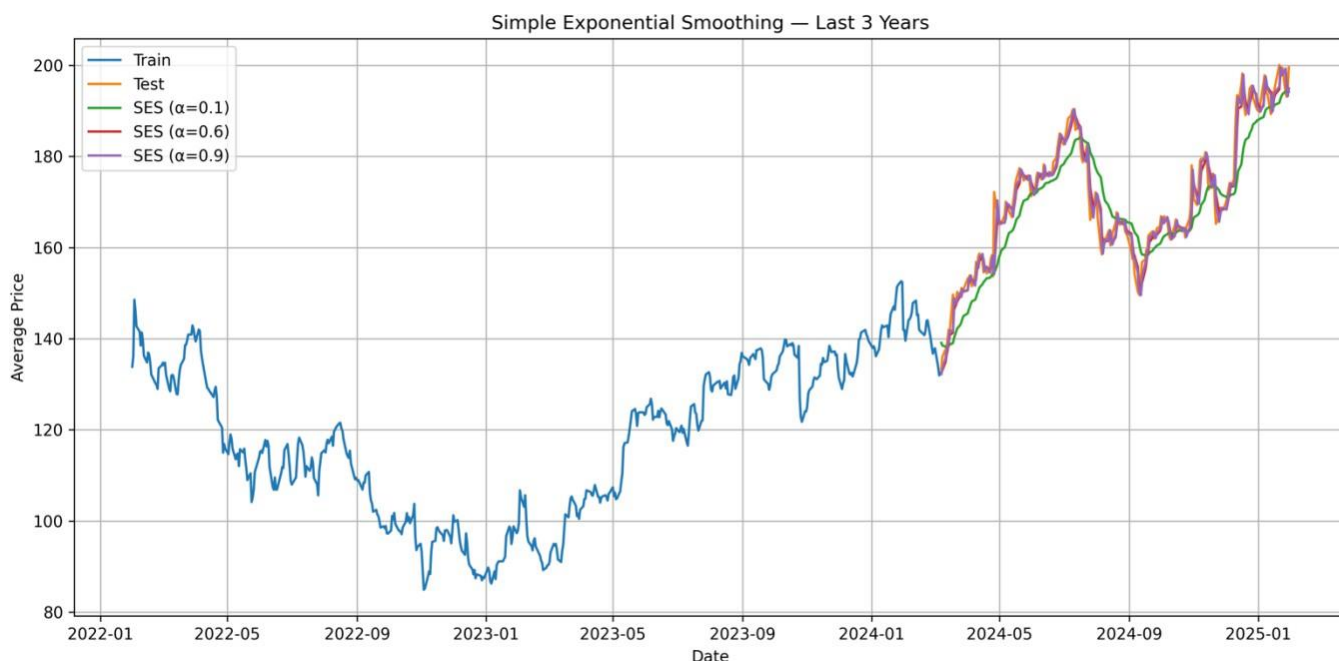


Рис. 2. Реалізація методу простого експоненціального згладжування на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: складено автором



Рис. 3 Реалізація методу простого експоненціального згладжування на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: складено автором

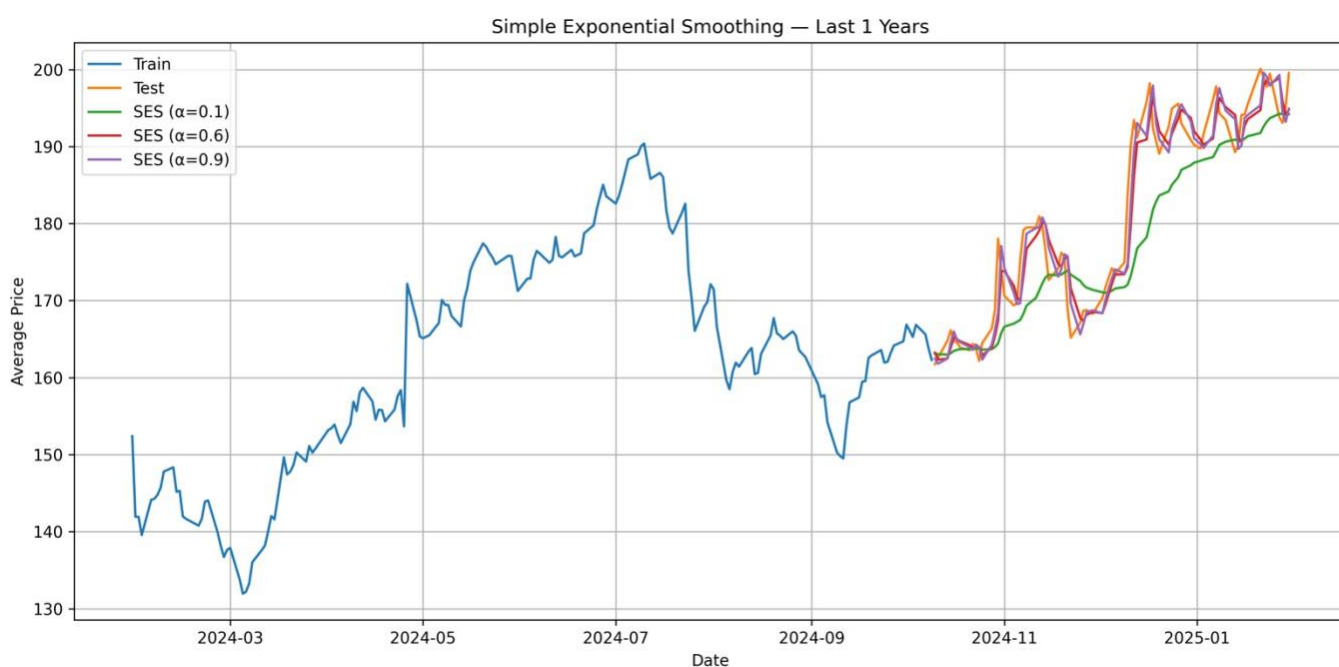


Рис. 4. Реалізація методу простого експоненціального згладжування на річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: складено автором

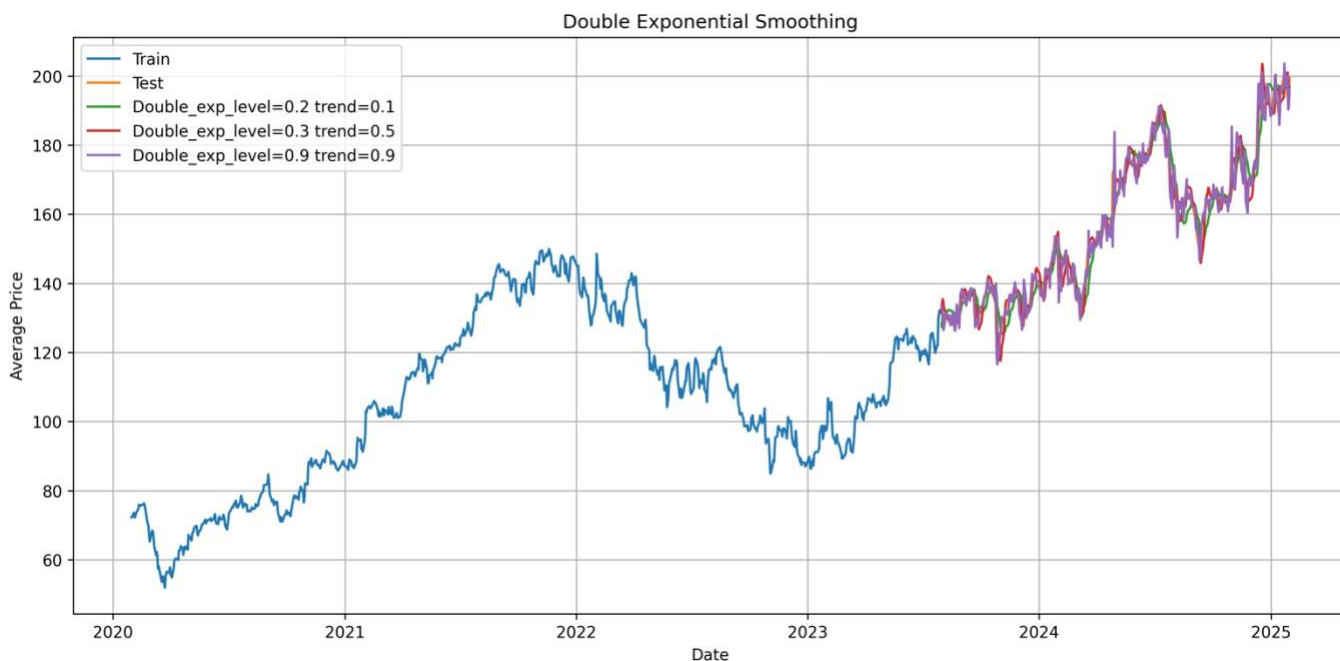


Рис. 5. Реалізація методу подвійного експоненціального згладжування на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: розроблено автором

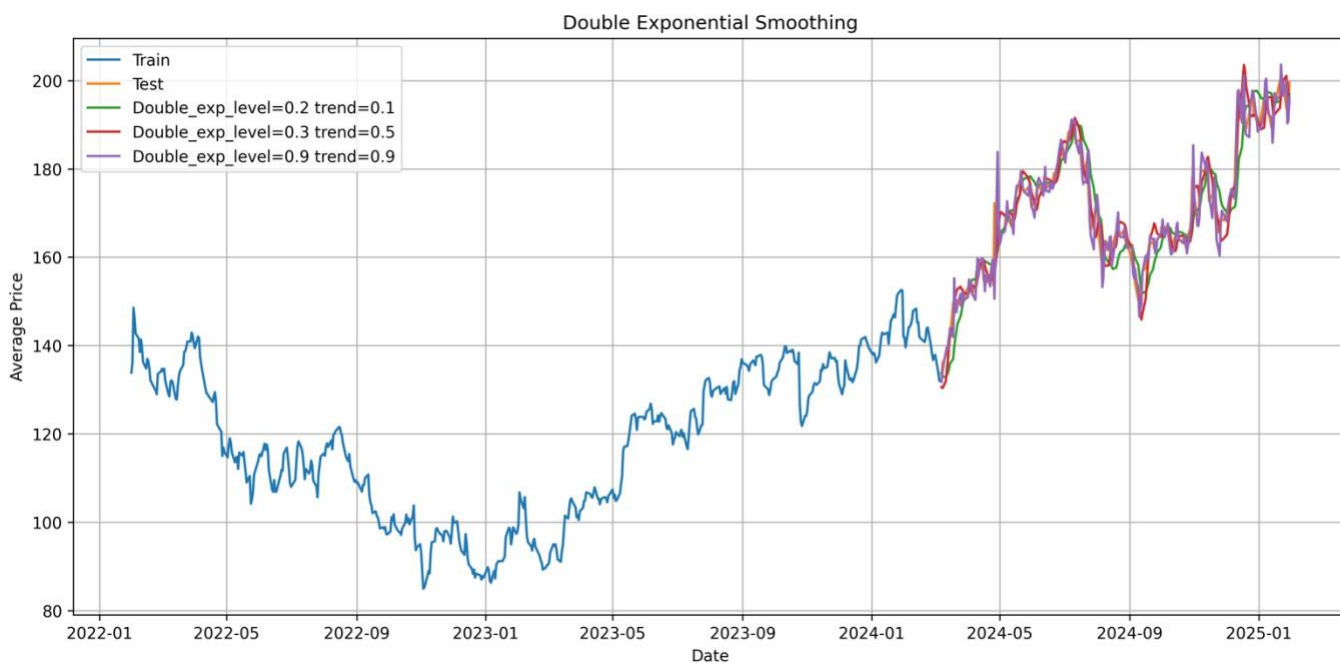


Рис. 6. Реалізація методу подвійного експоненціального згладжування на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: розроблено автором

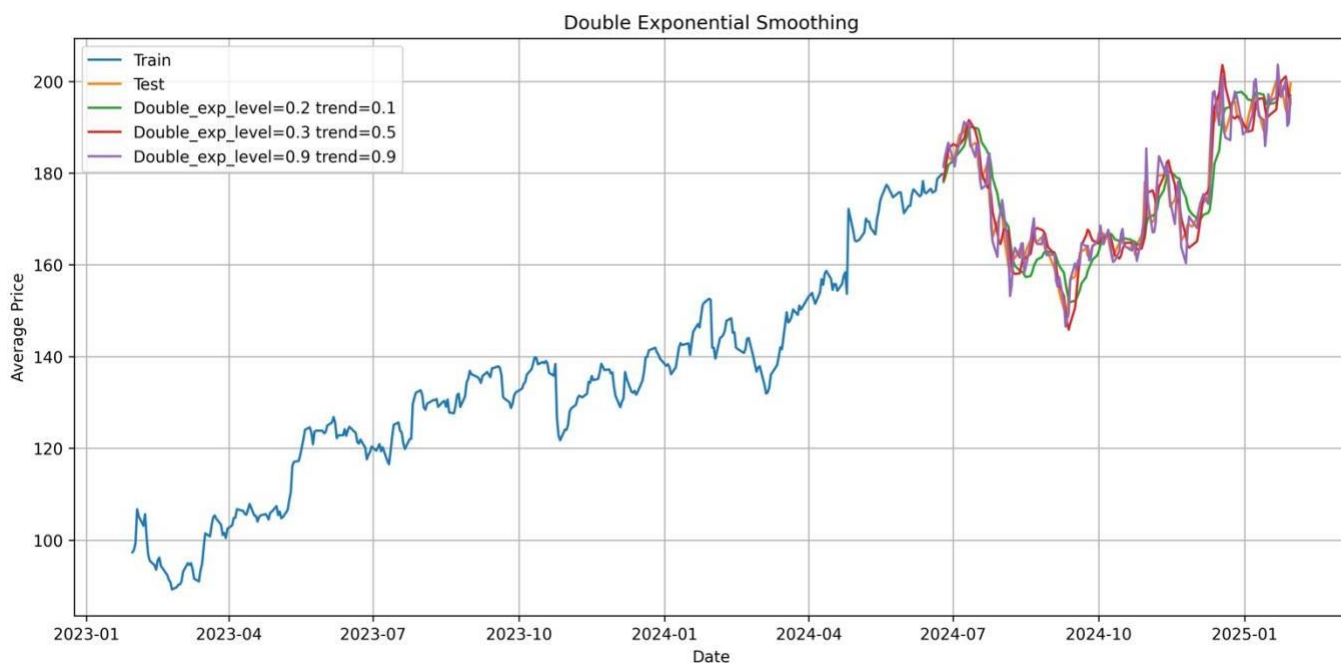


Рис. 7. Реалізація методу подвійного експоненціального згладжування на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: розроблено автором

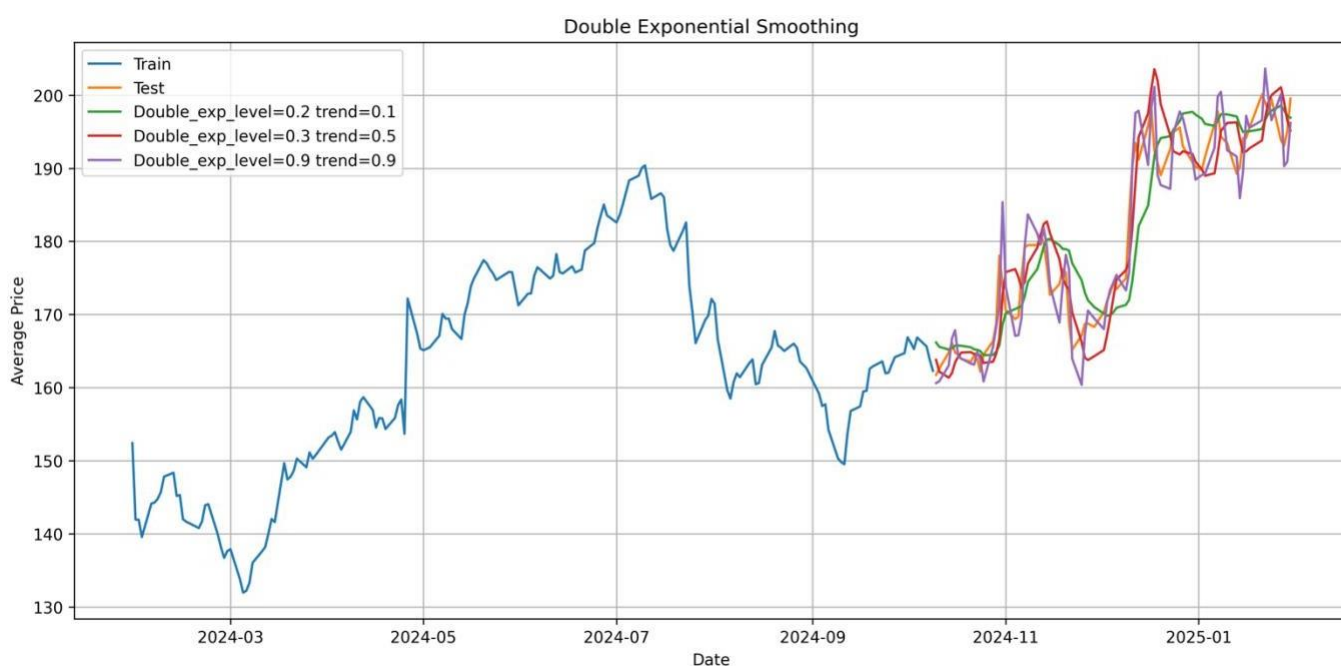


Рис. 8. Реалізація методу подвійного експоненціального згладжування на річну вибірку з середніх щоденних цін на акції компанії Google LLC

Джерело: розроблено автором

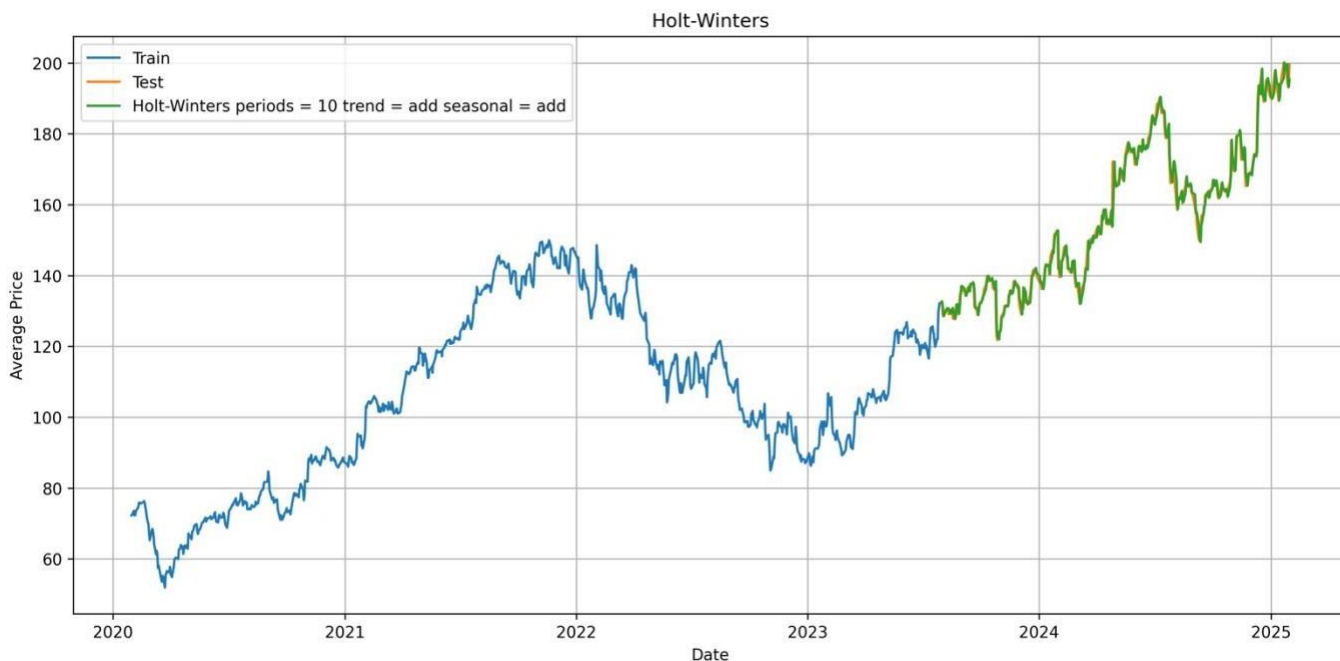


Рис. 9. Реалізація методу Холта-Вінтерса на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

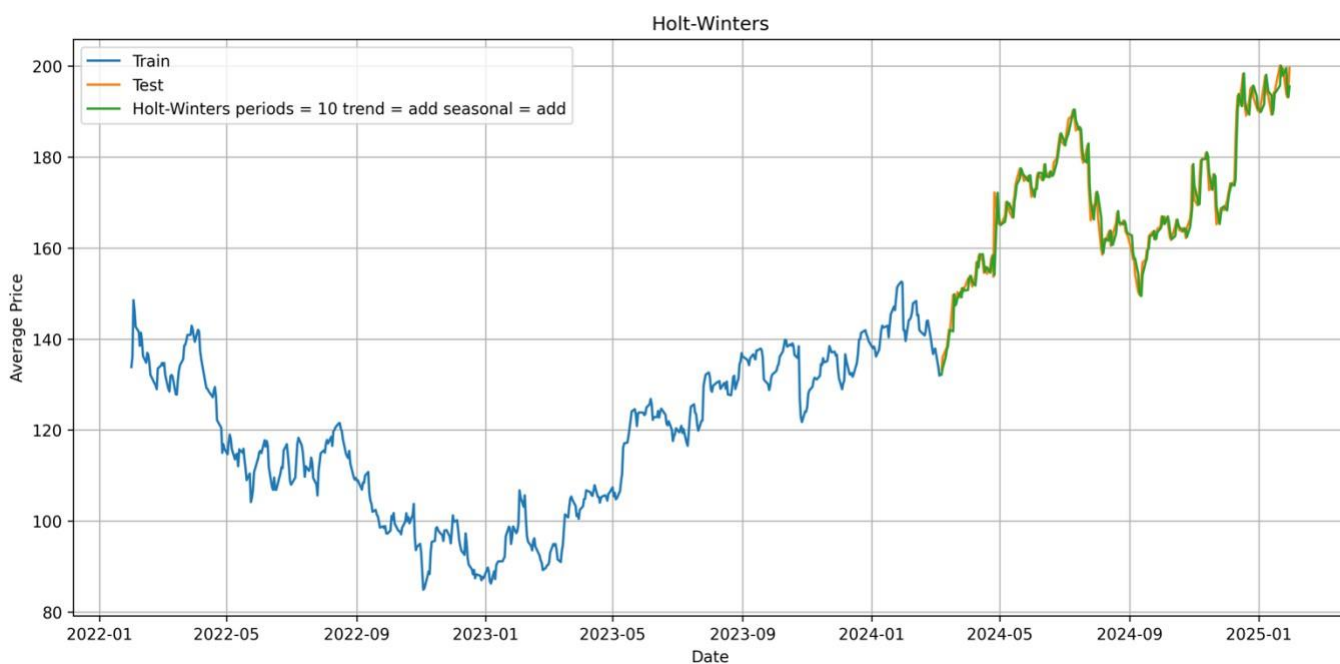


Рис. 10. Реалізація методу Холта-Вінтерса на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

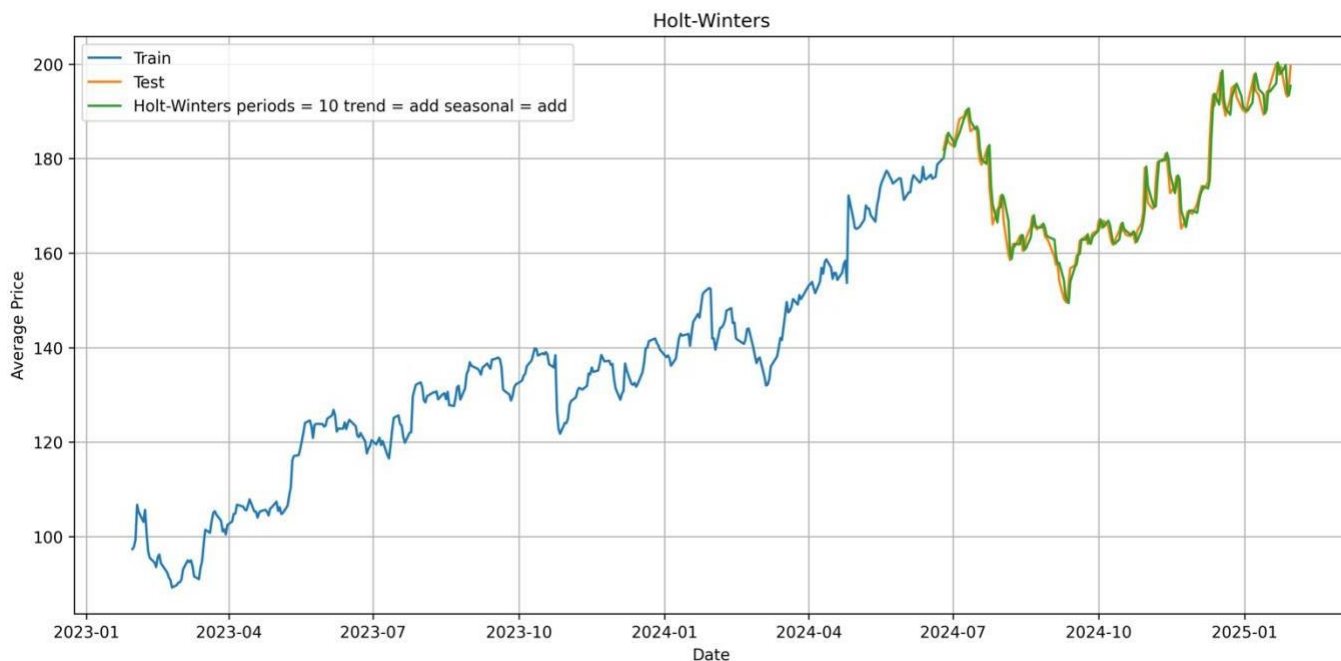


Рис. 11. Реалізація методу Холта-Вінтерса на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

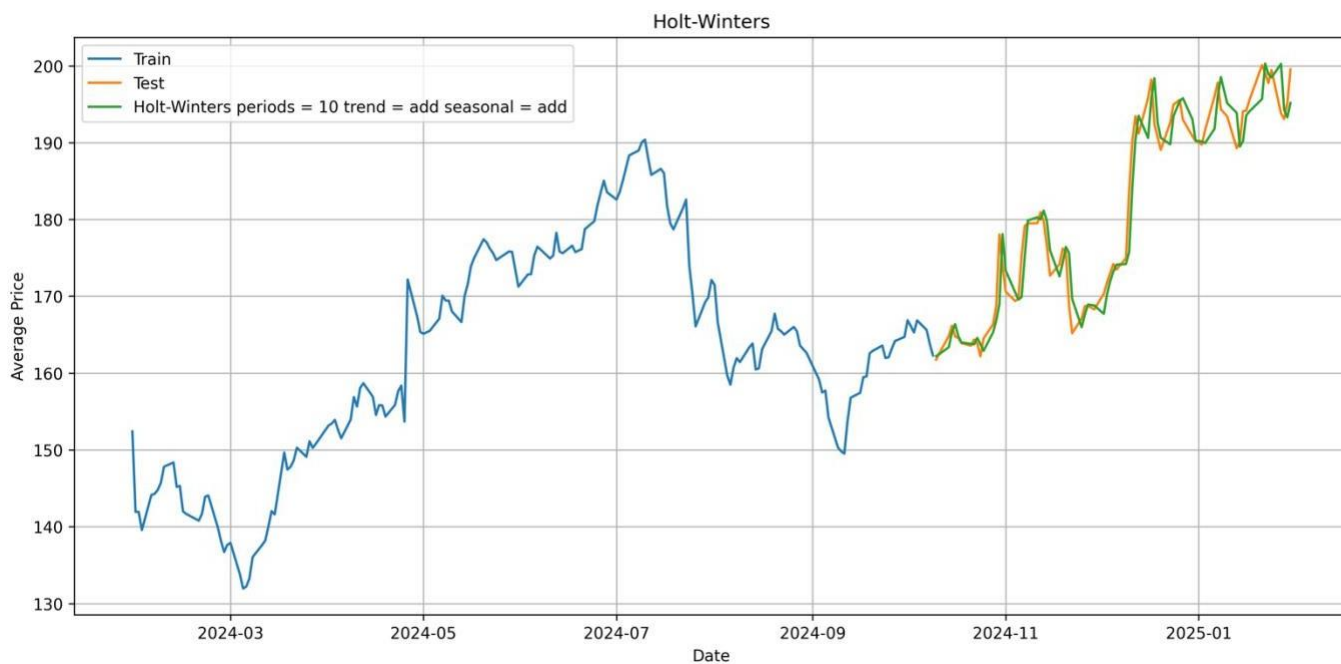


Рис. 12. Реалізація методу Холта-Вінтерса на річну вибірку з середніх щоденних цін на акції компанії Google LLC

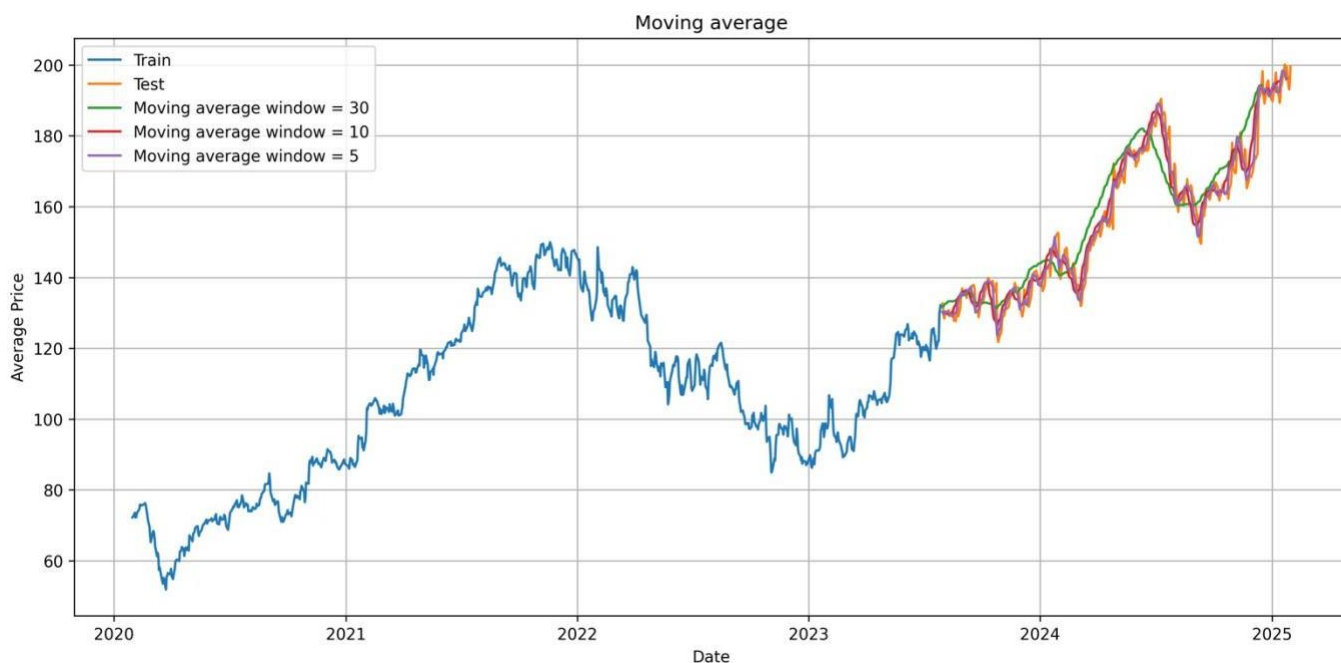


Рис. 13. Реалізація методу ковзного середнього на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

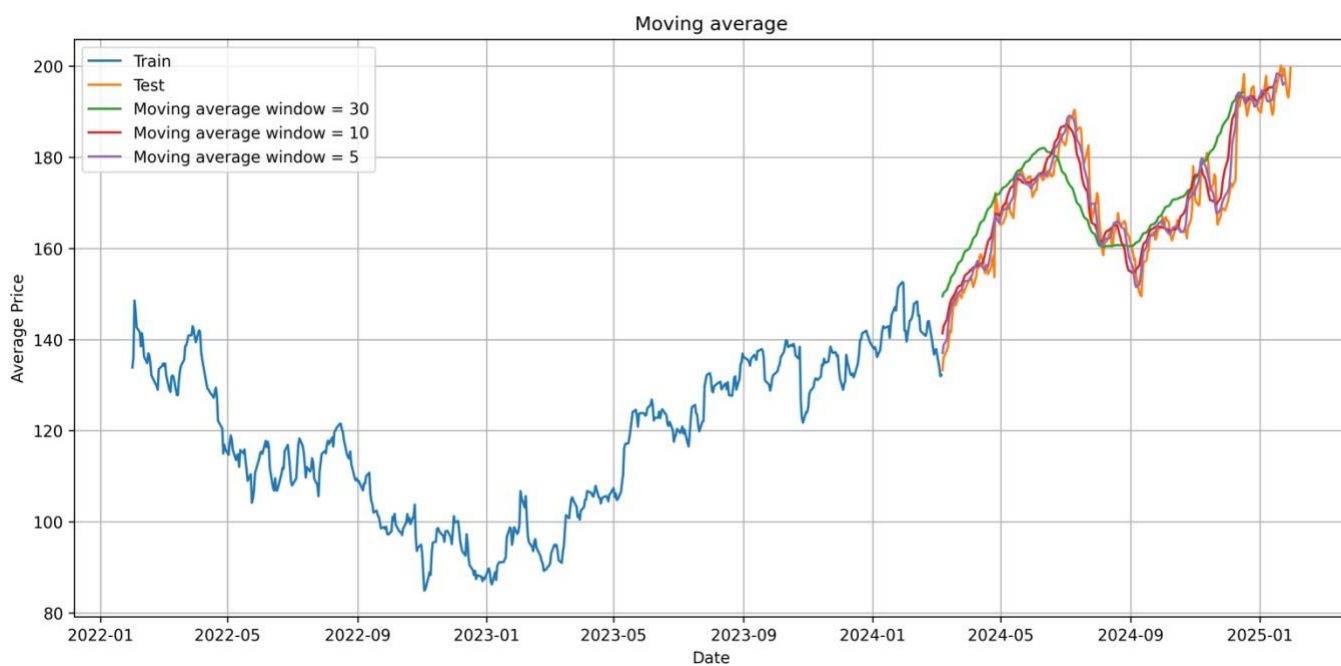


Рис. 14. Реалізація методу ковзного середнього на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

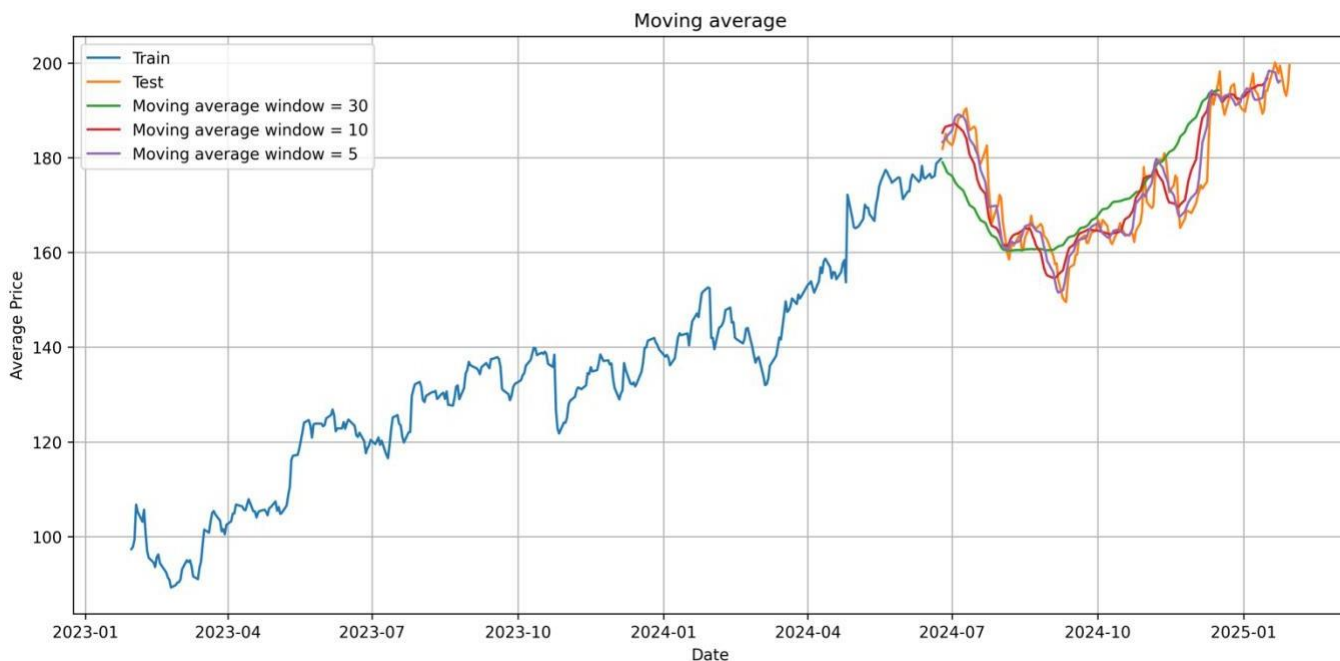


Рис. 15. Реалізація методу ковзного середнього на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

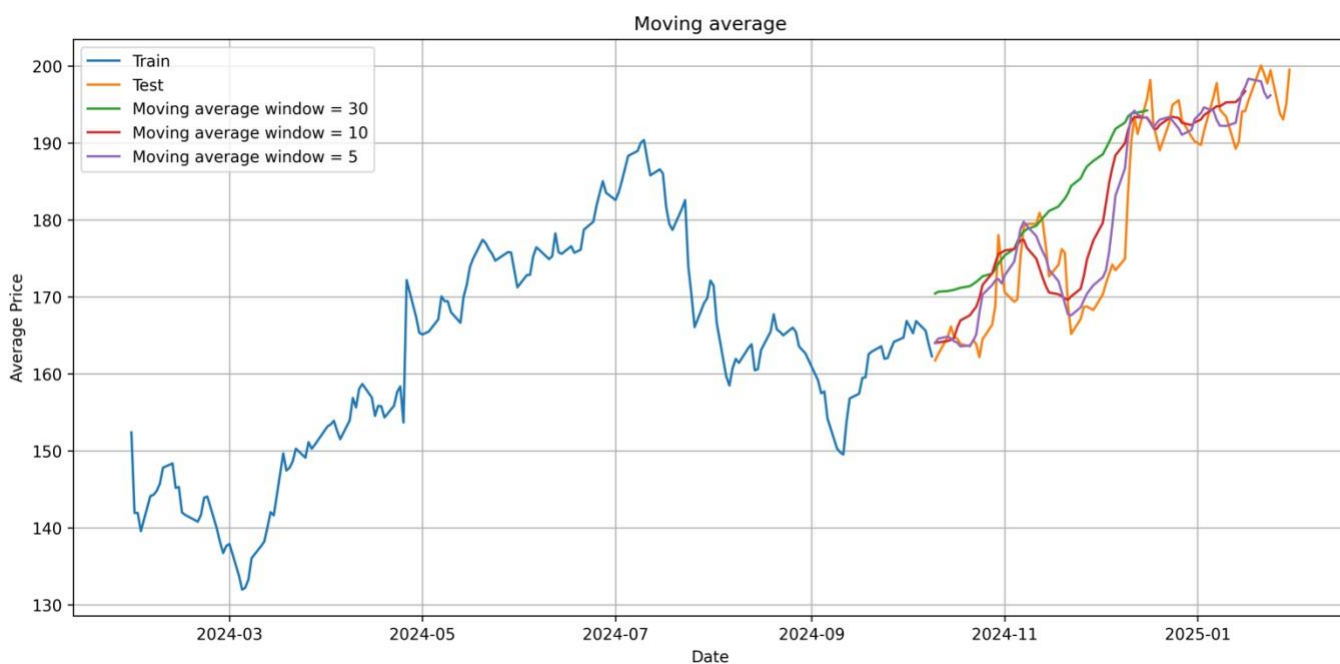


Рис. 16. Реалізація методу ковзного середнього на річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 17. Реалізація методу простої авторегресії  $(2, 0, 0)$  на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 18. Реалізація методу авторегресії  $(2, 0, 1)$  на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 19. Реалізація методу авторегресії (2, 2, 1) на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

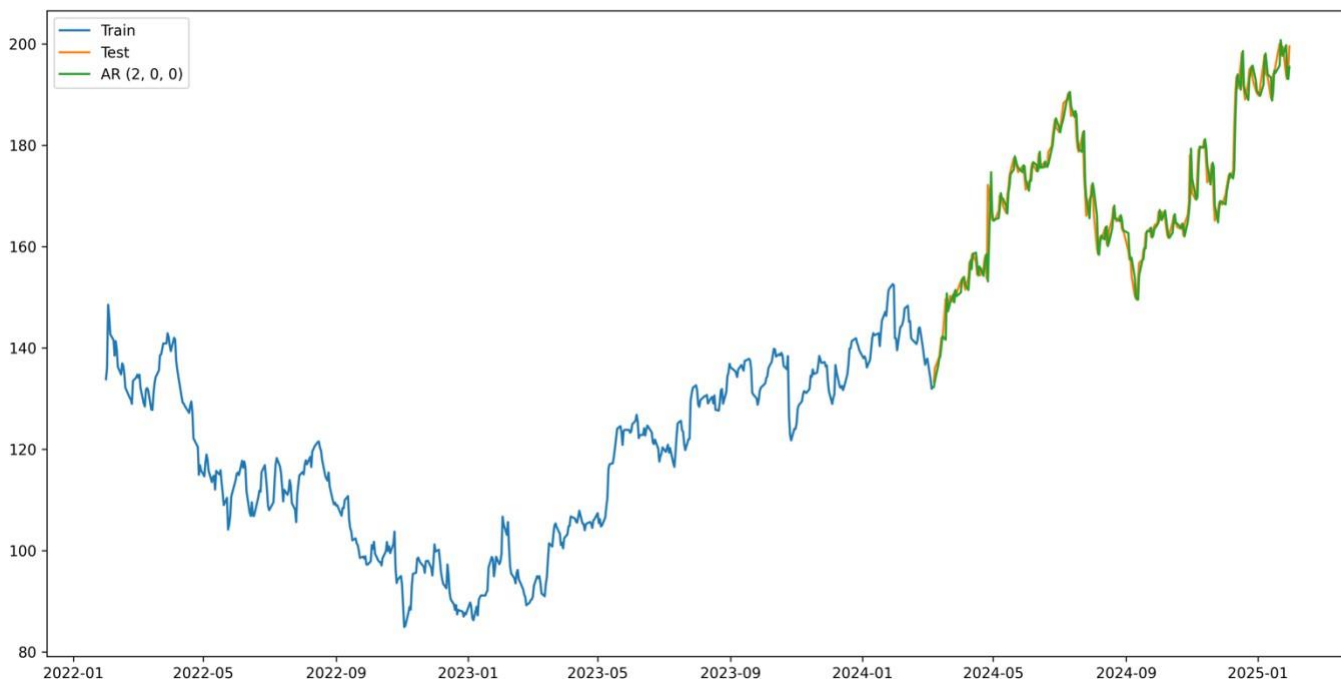


Рис. 20. Реалізація методу простої авторегресії (2, 0, 0) на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

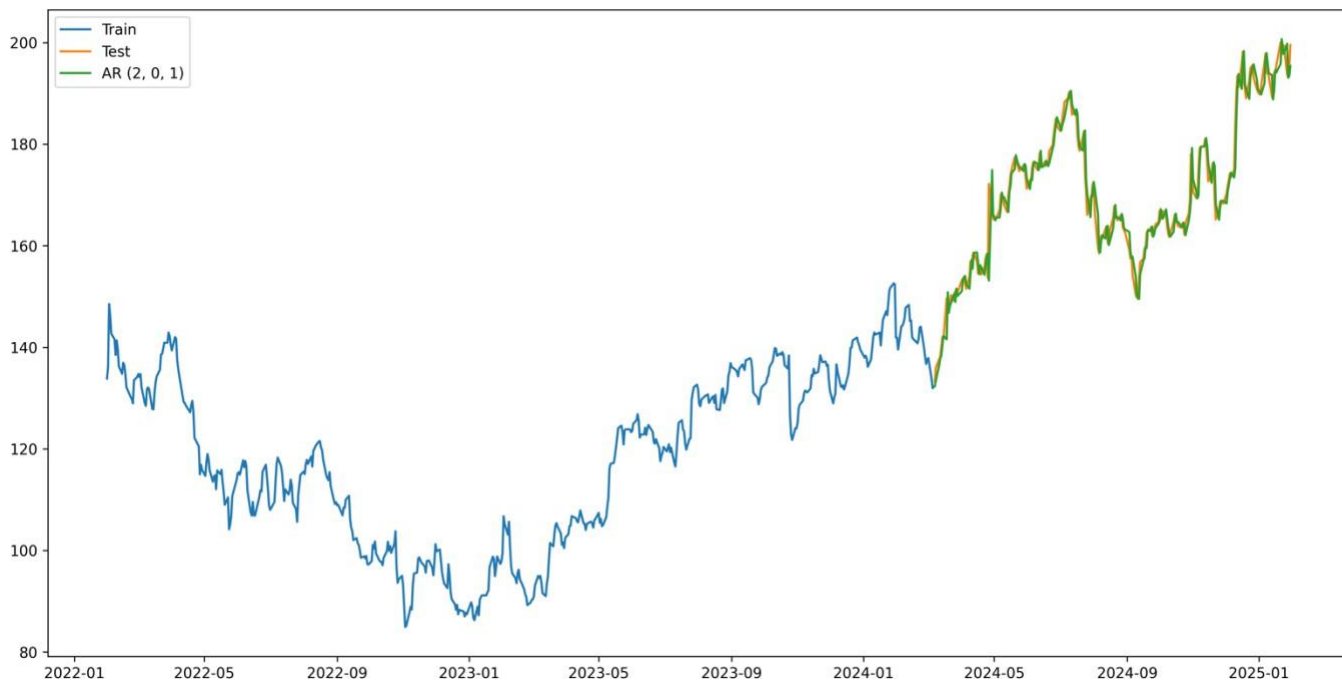


Рис. 21. Реалізація методу простої авторегресії (2, 0, 1) на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

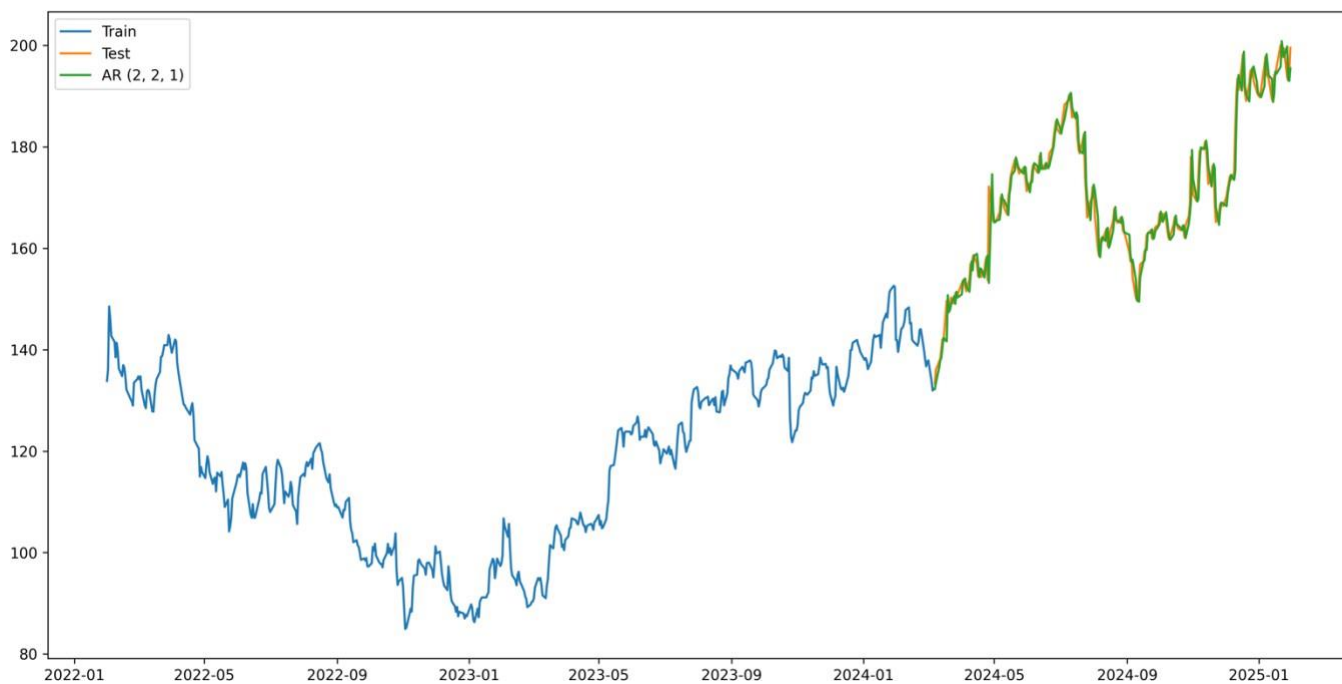


Рис. 22. Реалізація методу простої авторегресії (2, 2, 1) на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 23. Реалізація методу простої авторегресії  $(2, 0, 0)$  на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

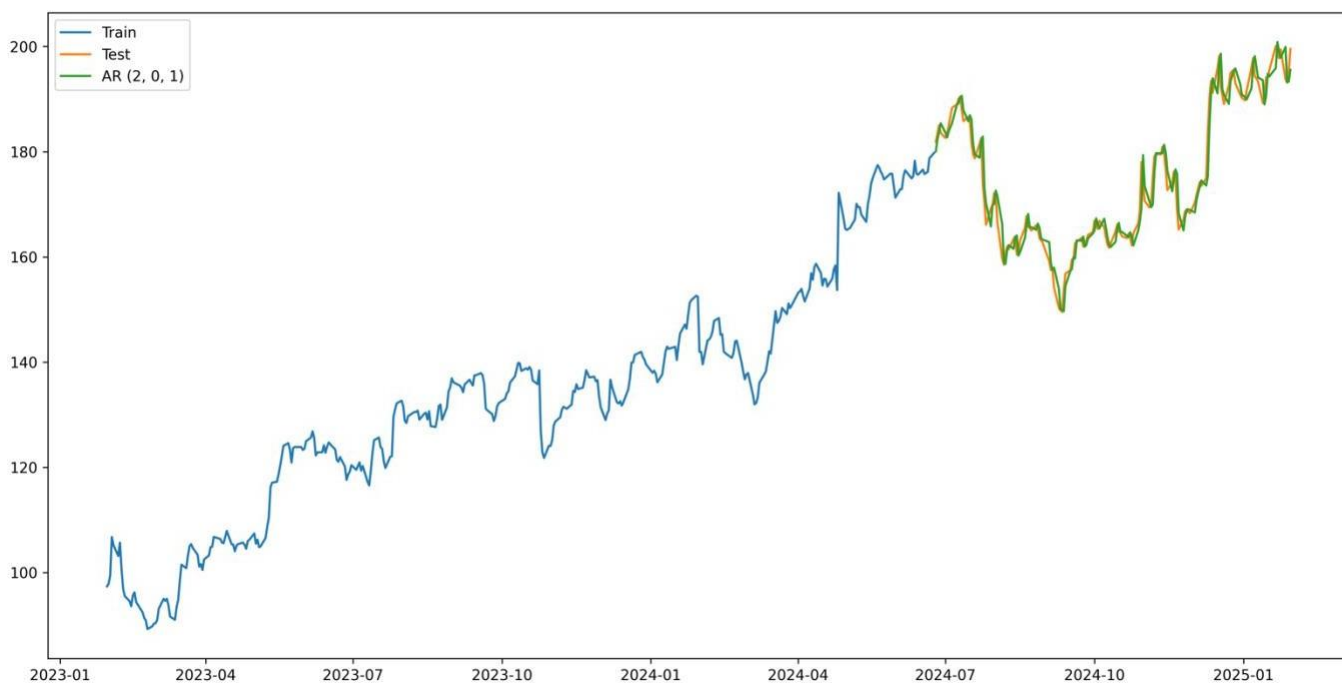


Рис. 24. Реалізація методу простої авторегресії  $(2, 0, 1)$  на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

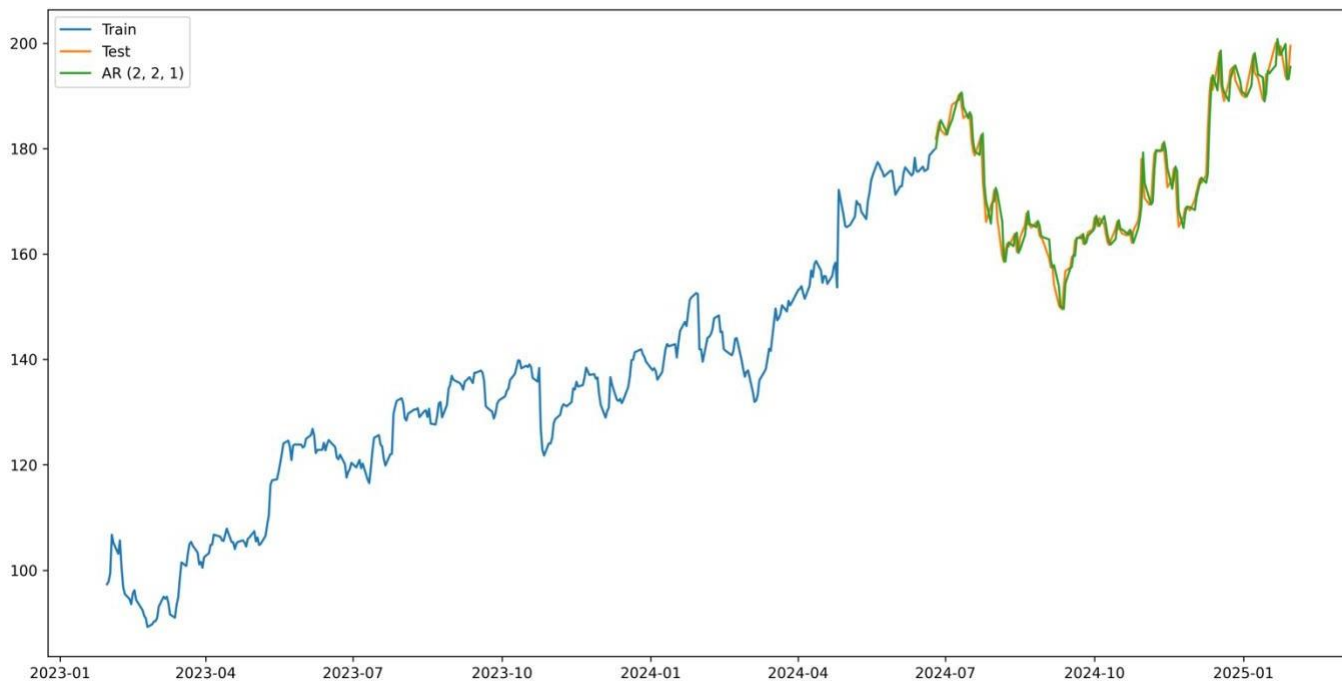


Рис. 25. Реалізація методу простої авторегресії  $(2, 2, 1)$  на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC

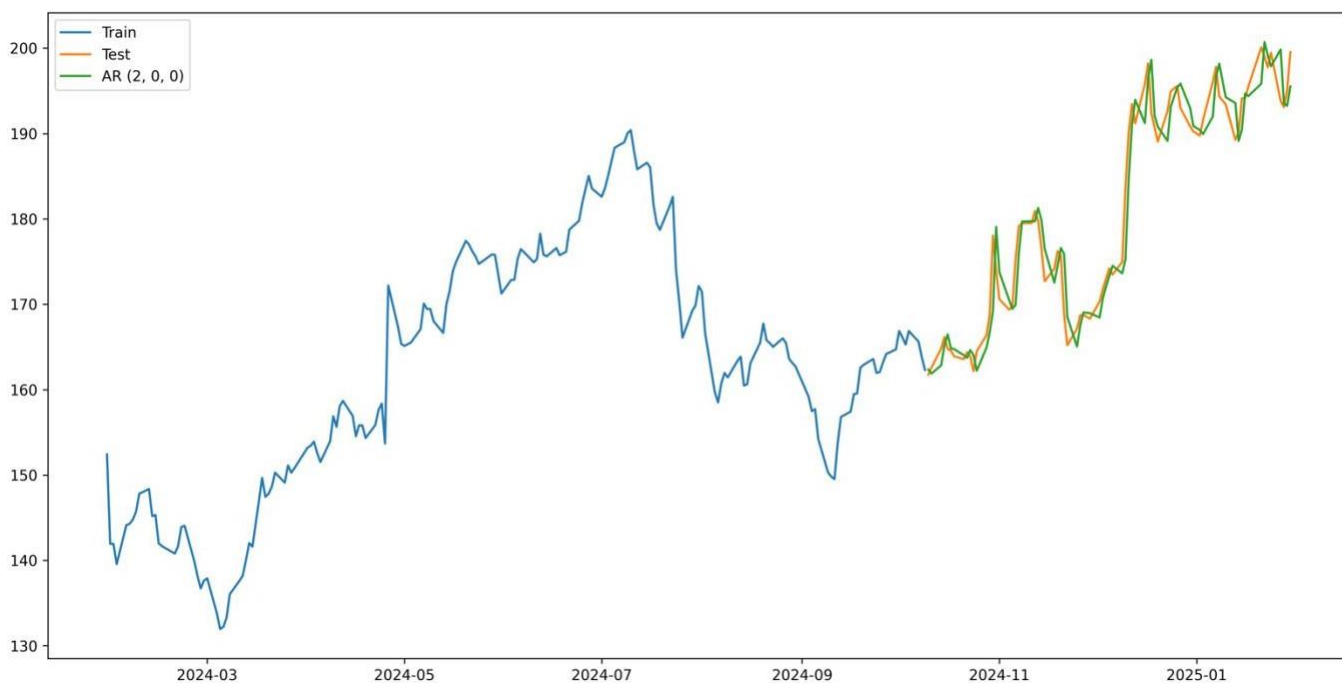


Рис. 26. Реалізація методу простої авторегресії  $(2, 0, 0)$  на річну вибірку з середніх щоденних цін на акції компанії Google LLC

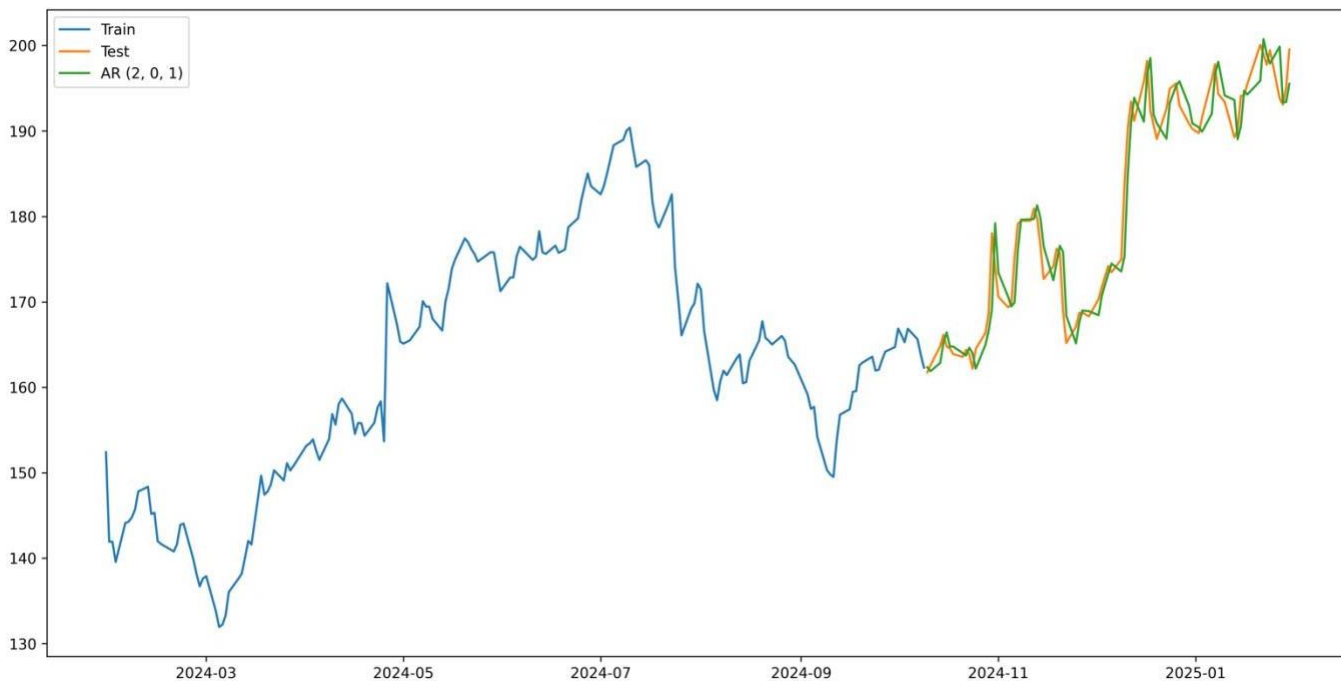


Рис. 27. Реалізація методу простої авторегресії (2, 0, 1) на річну вибірку з середніх щоденних цін на акції компанії Google LLC

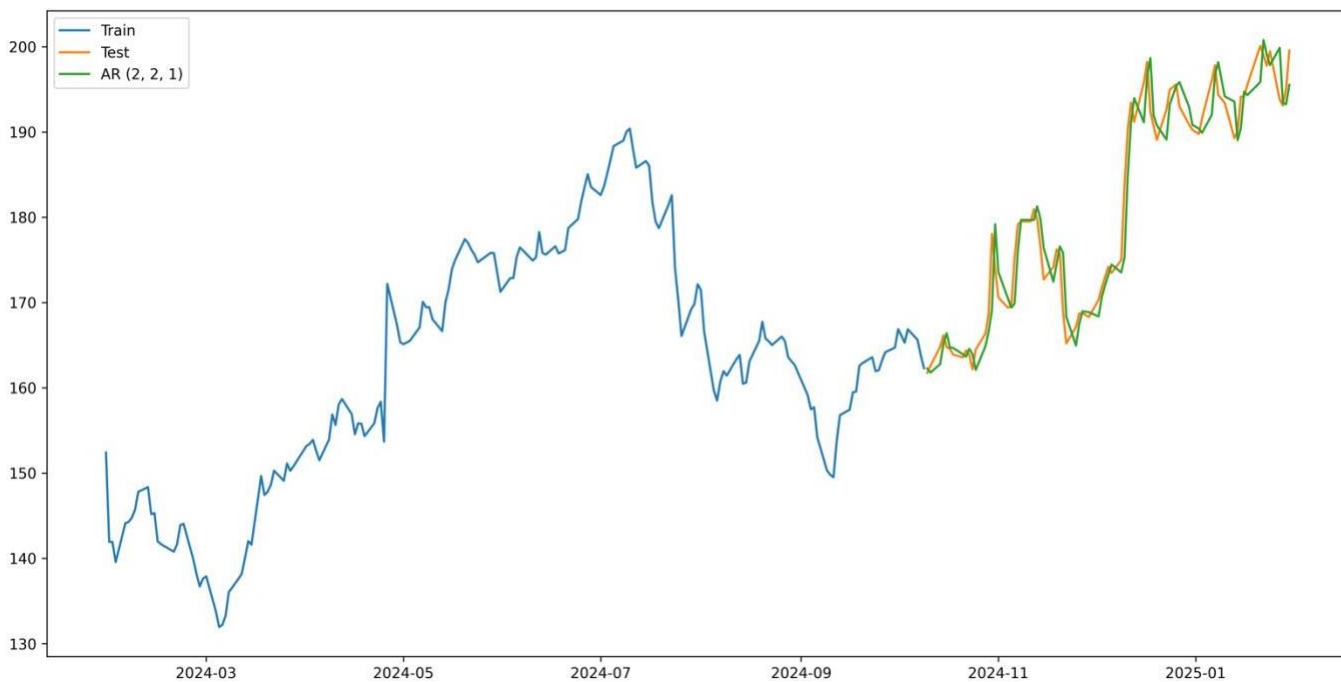


Рис. 28. Реалізація методу простої авторегресії (2, 2, 1) на річну вибірку з середніх щоденних цін на акції компанії Google LLC

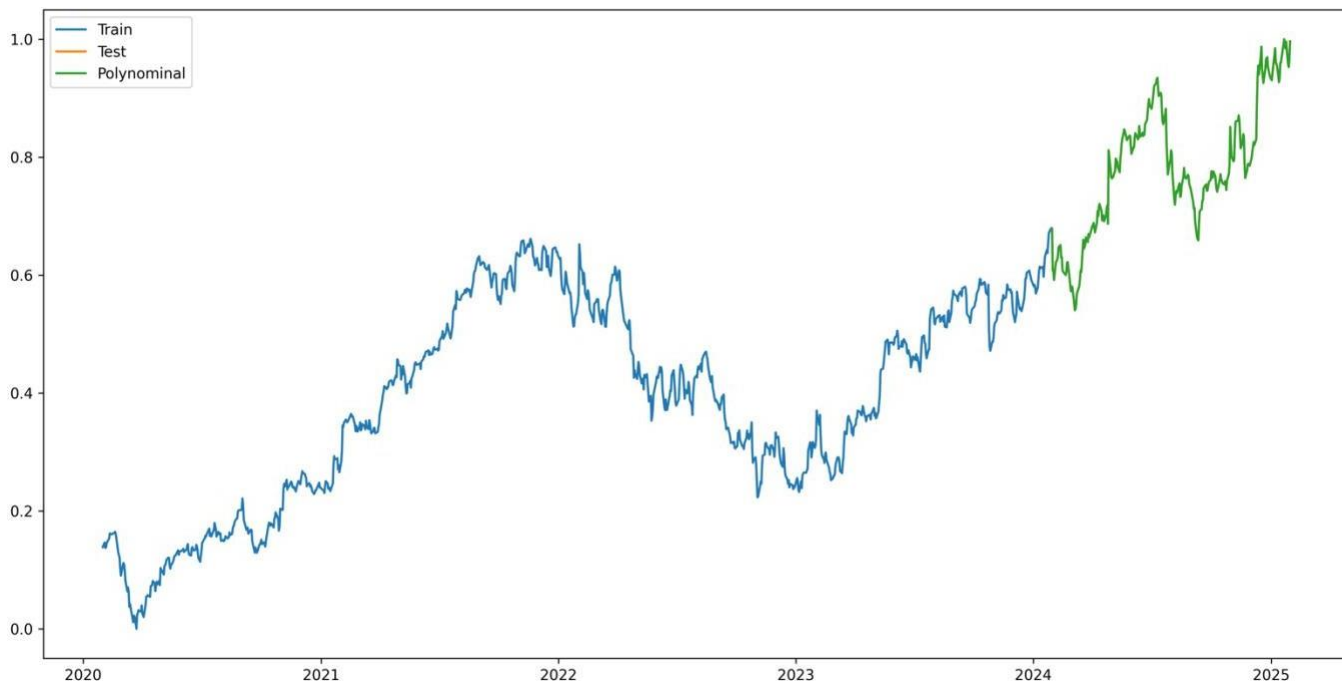


Рис. 29 Реалізація методу простої лінійної регресії на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

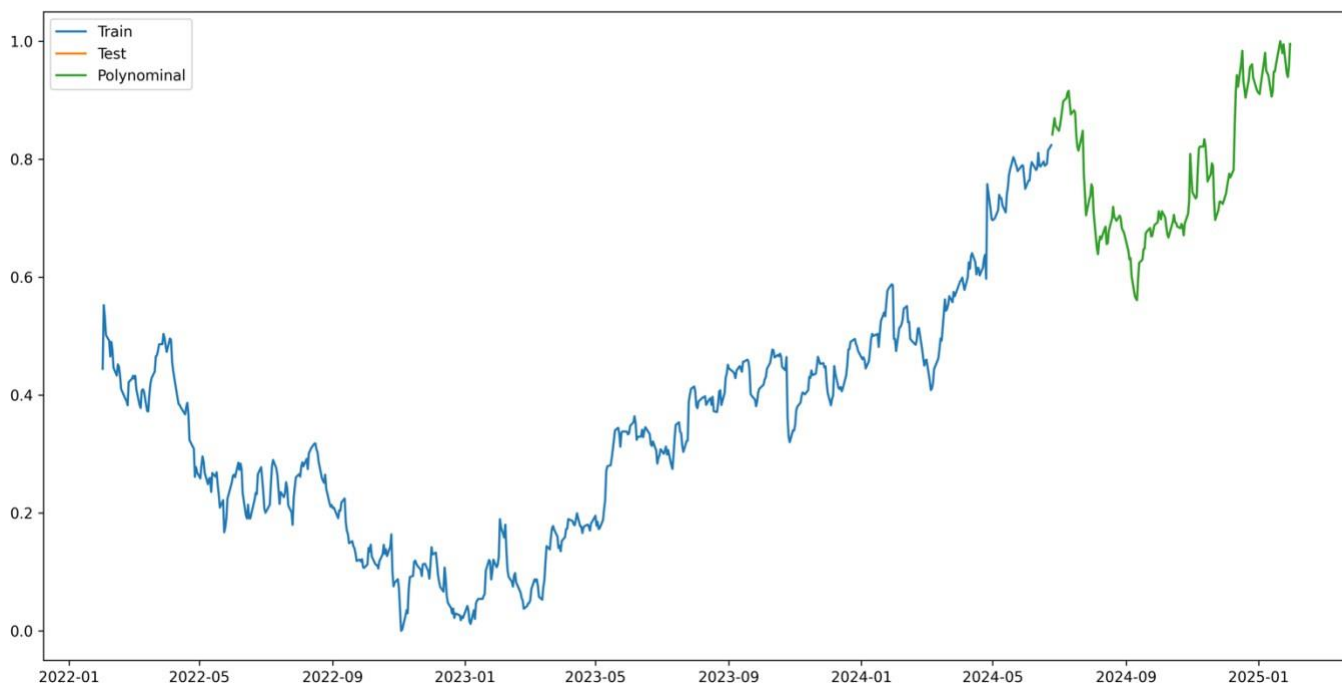


Рис. 30. Реалізація методу простої лінійної регресії на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

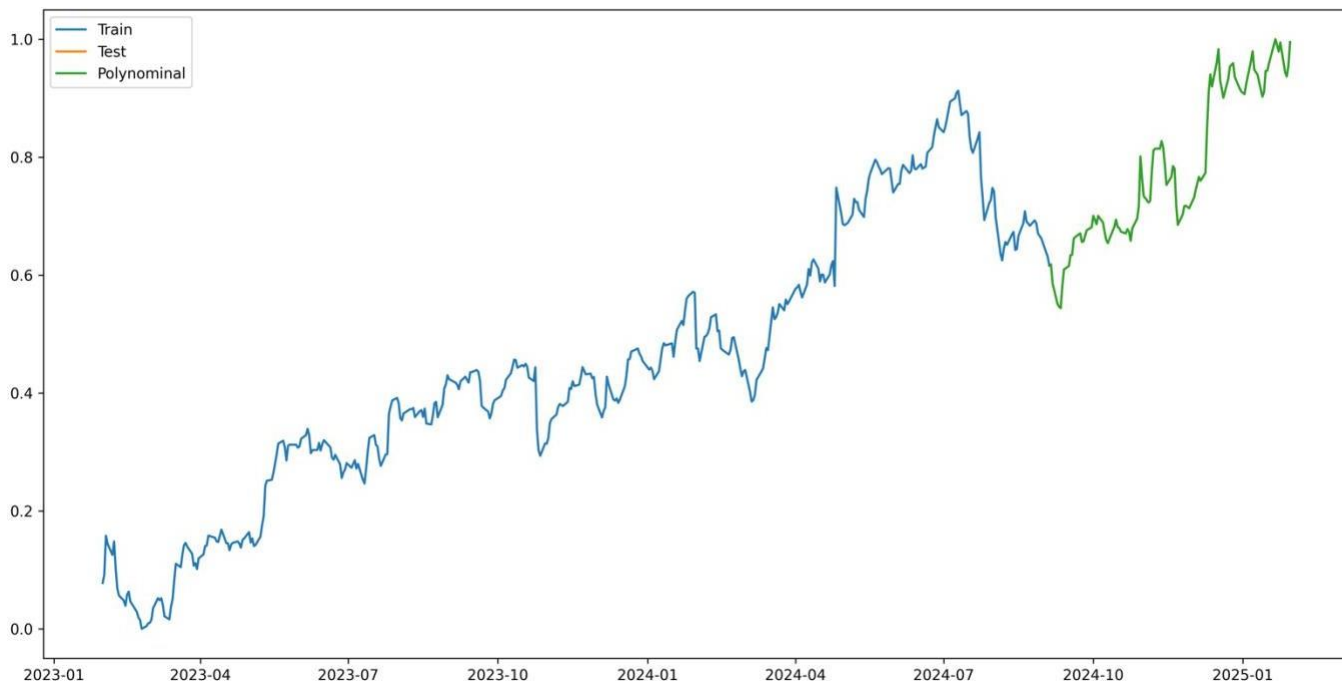


Рис. 31. Реалізація методу простої лінійної регресії на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 32. Реалізація методу простої лінійної регресії на річну вибірку з середніх щоденних цін на акції компанії Google LLC

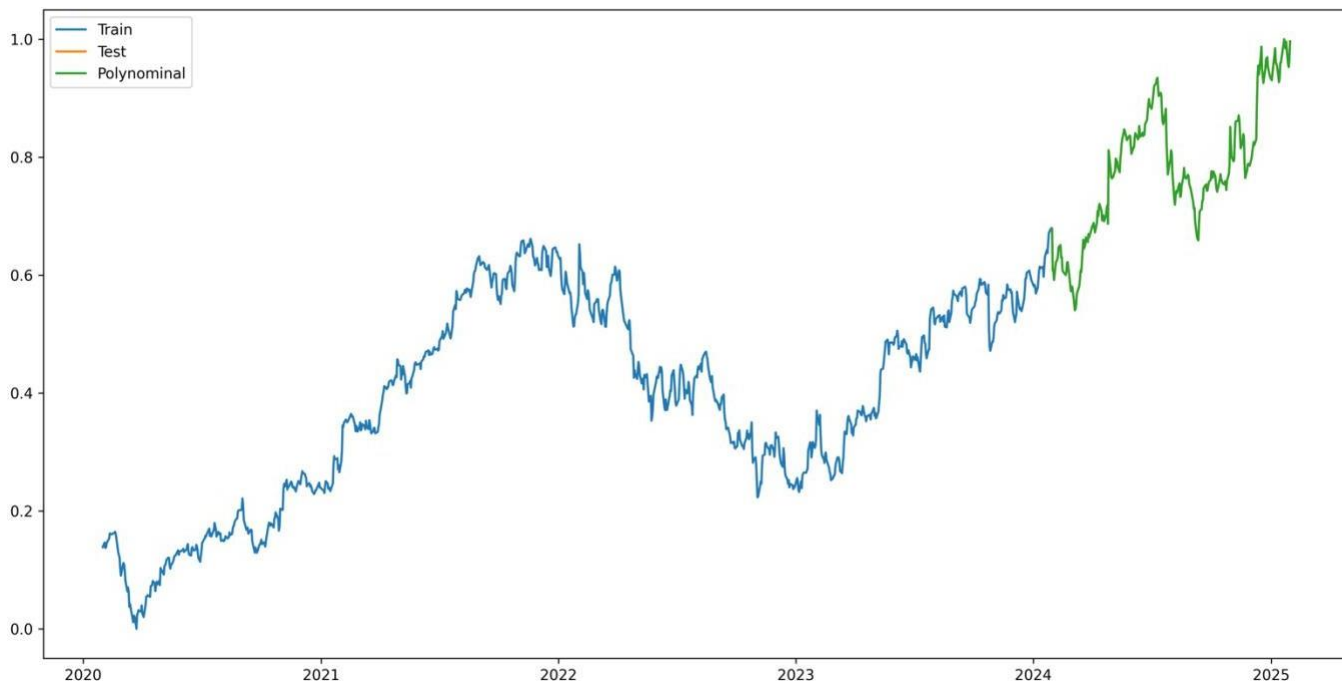


Рис. 33. Реалізація методу множинної регресії на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

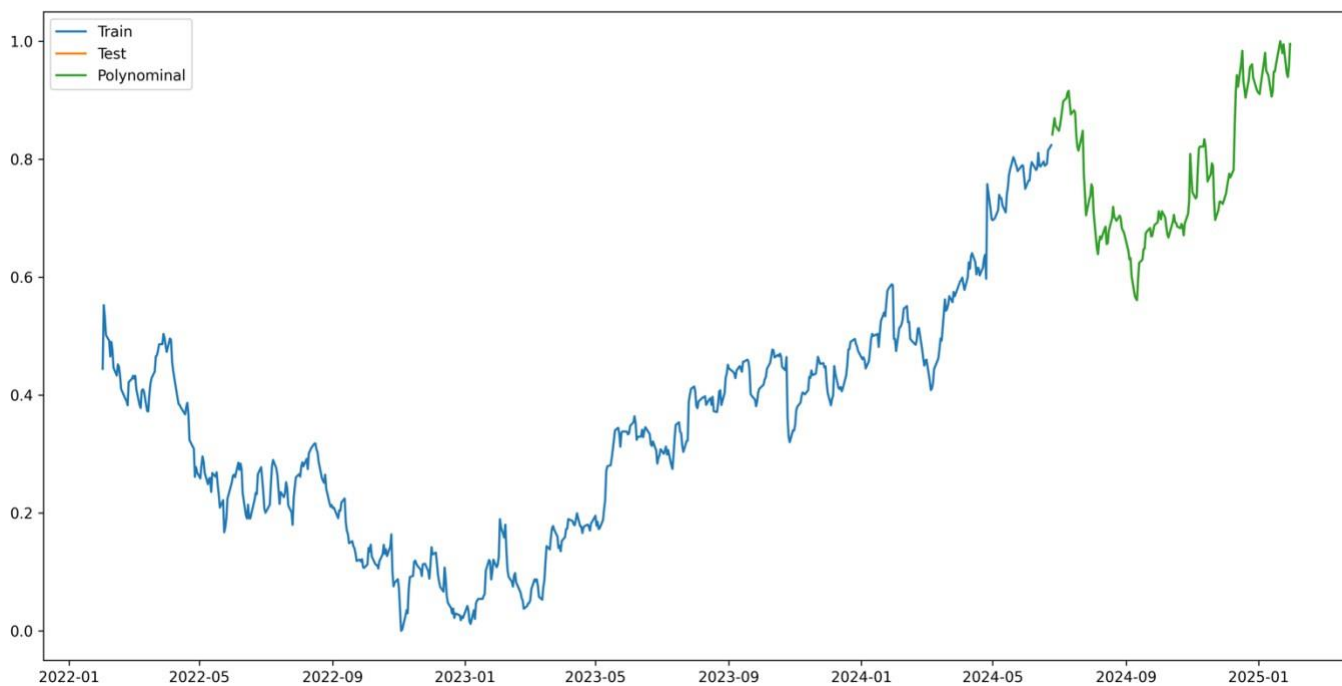


Рис. 34. Реалізація методу множинної регресії на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

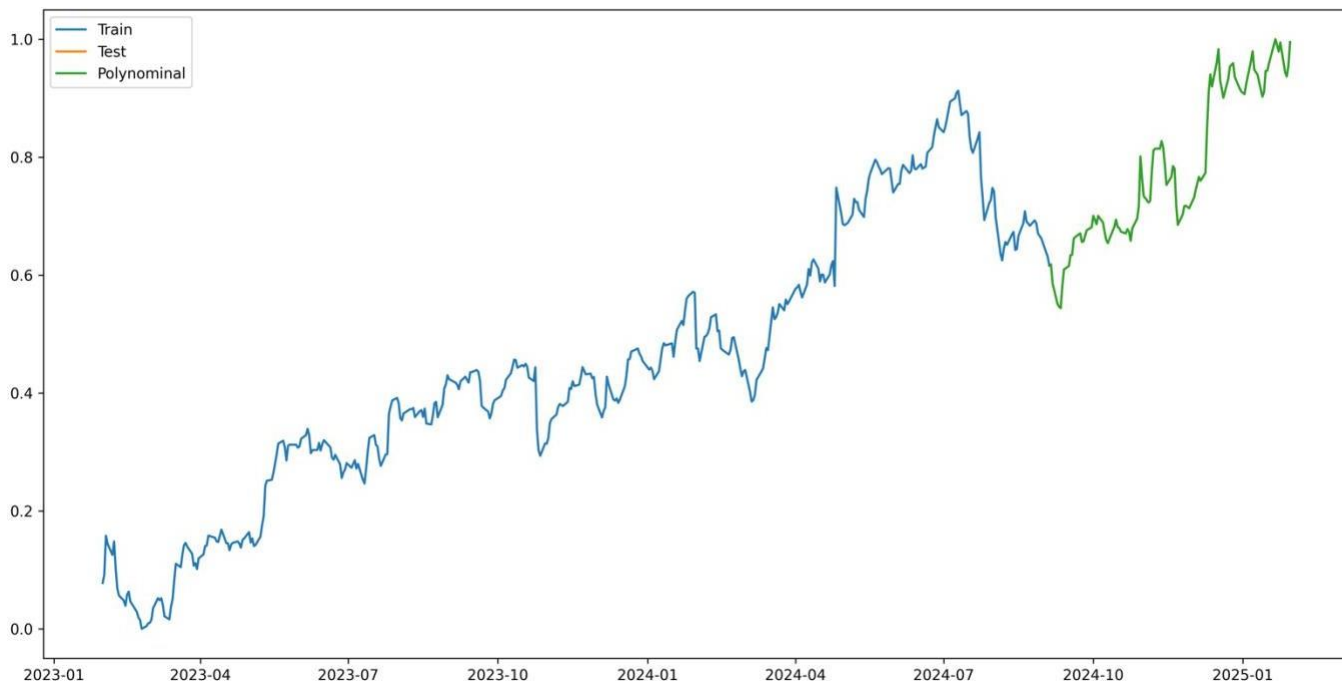


Рис. 35. Реалізація методу множинної регресії на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 36. Реалізація методу множинної регресії на річну вибірку з середніх щоденних цін на акції компанії Google LLC

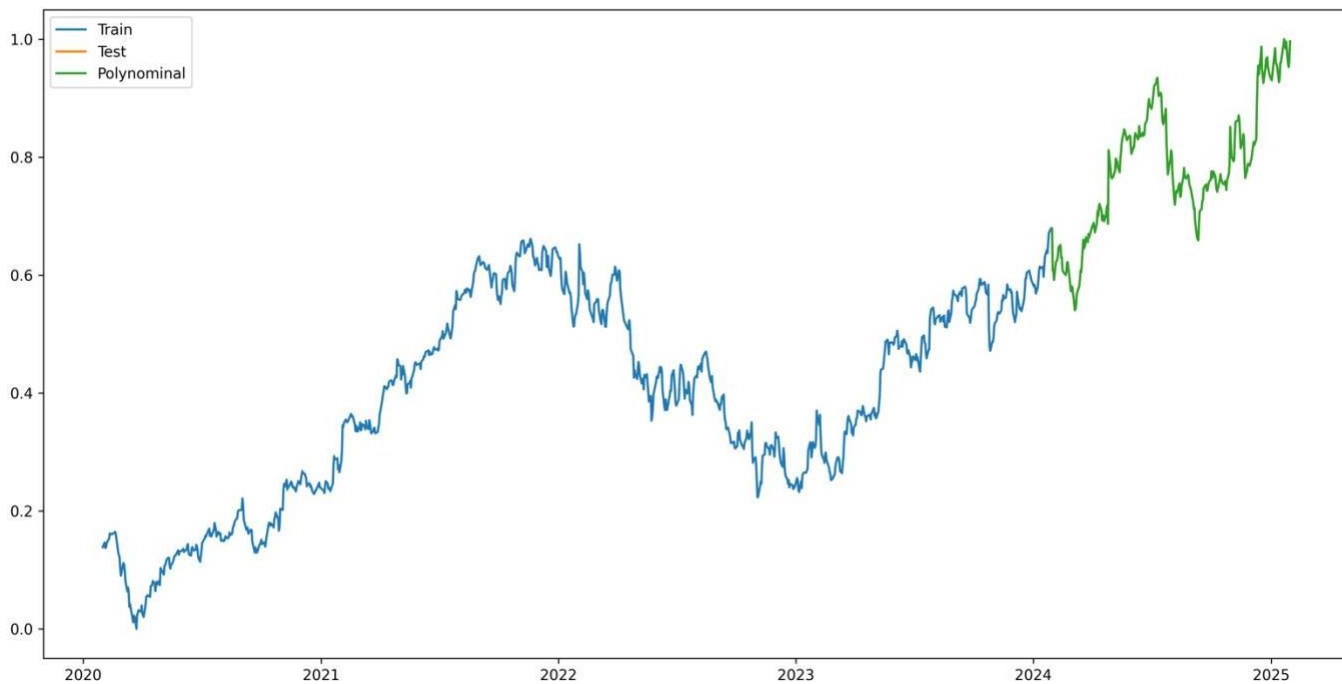


Рис. 37. Реалізація нелінійної регресії на 5-річну вибірку з середніх щоденних цін на акції компанії Google LLC

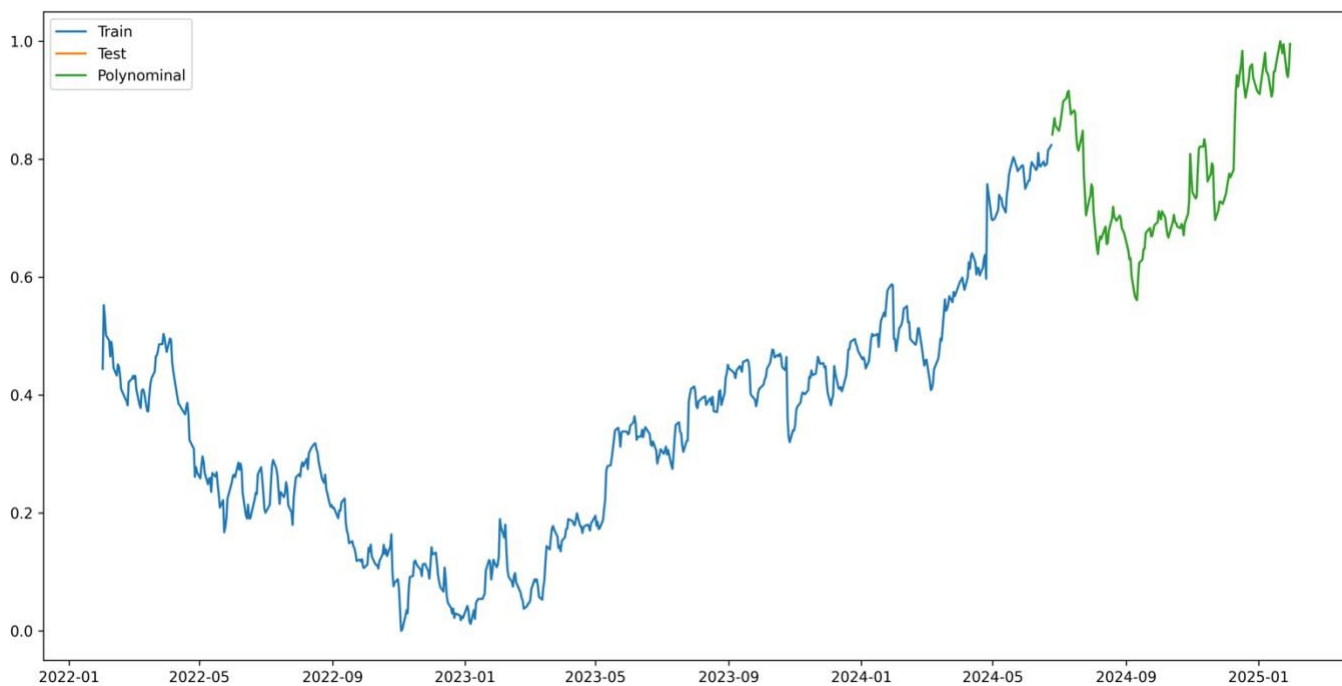


Рис. 38. Реалізація нелінійної регресії на 3-річну вибірку з середніх щоденних цін на акції компанії Google LLC

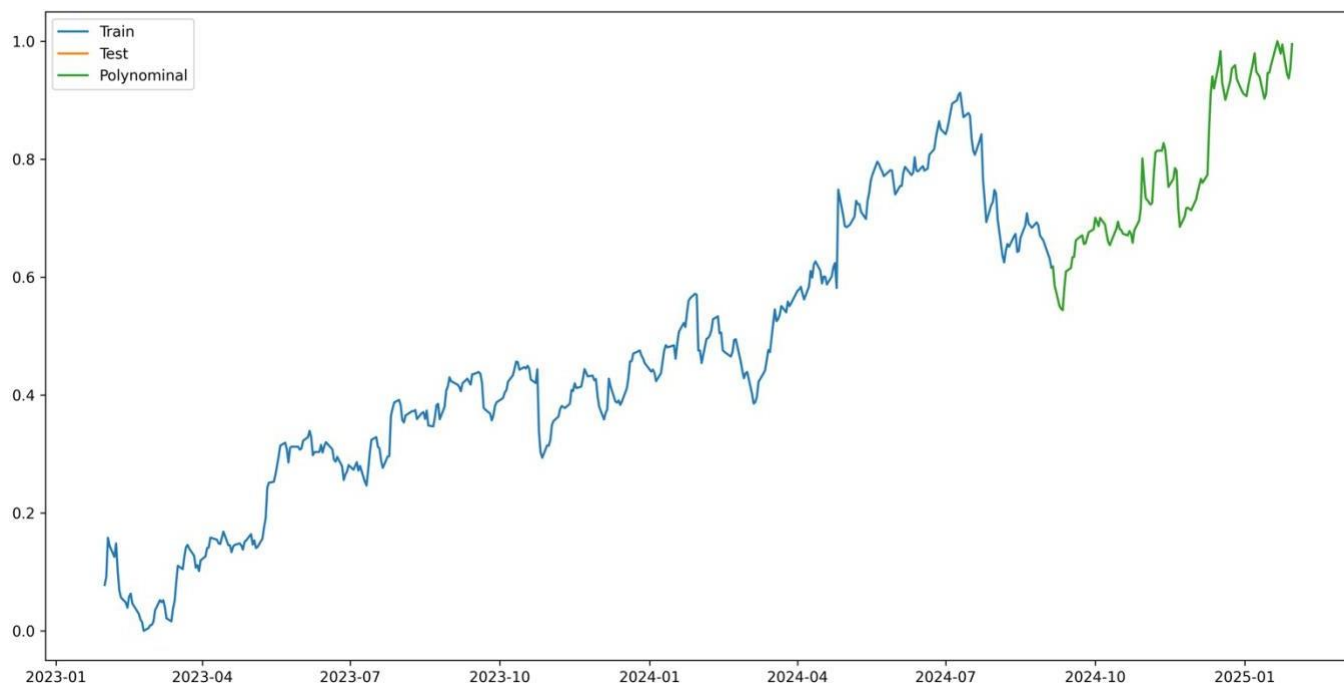


Рис. 39. Реалізація нелінійної регресії на 2-річну вибірку з середніх щоденних цін на акції компанії Google LLC



Рис. 40. Реалізація нелінійної регресії на річну вибірку з середніх щоденних цін на акції компанії Google LLC

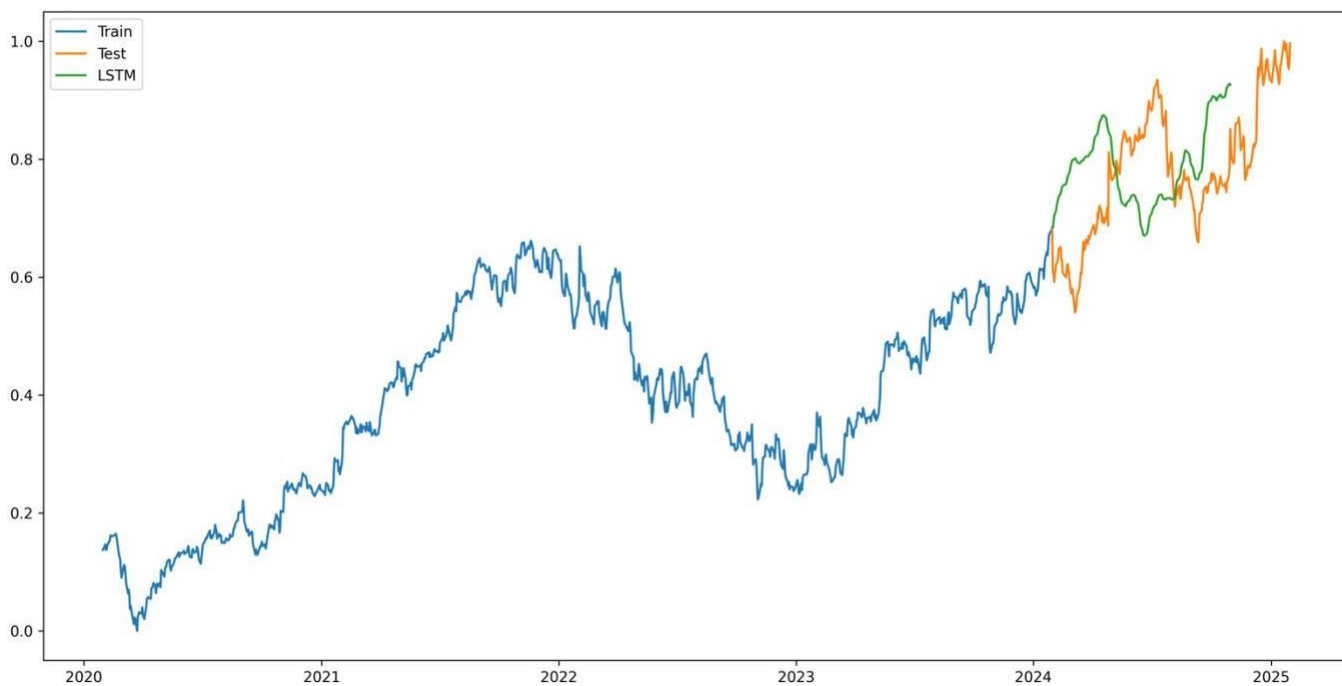


Рис. 41. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки (кількість епох 1)

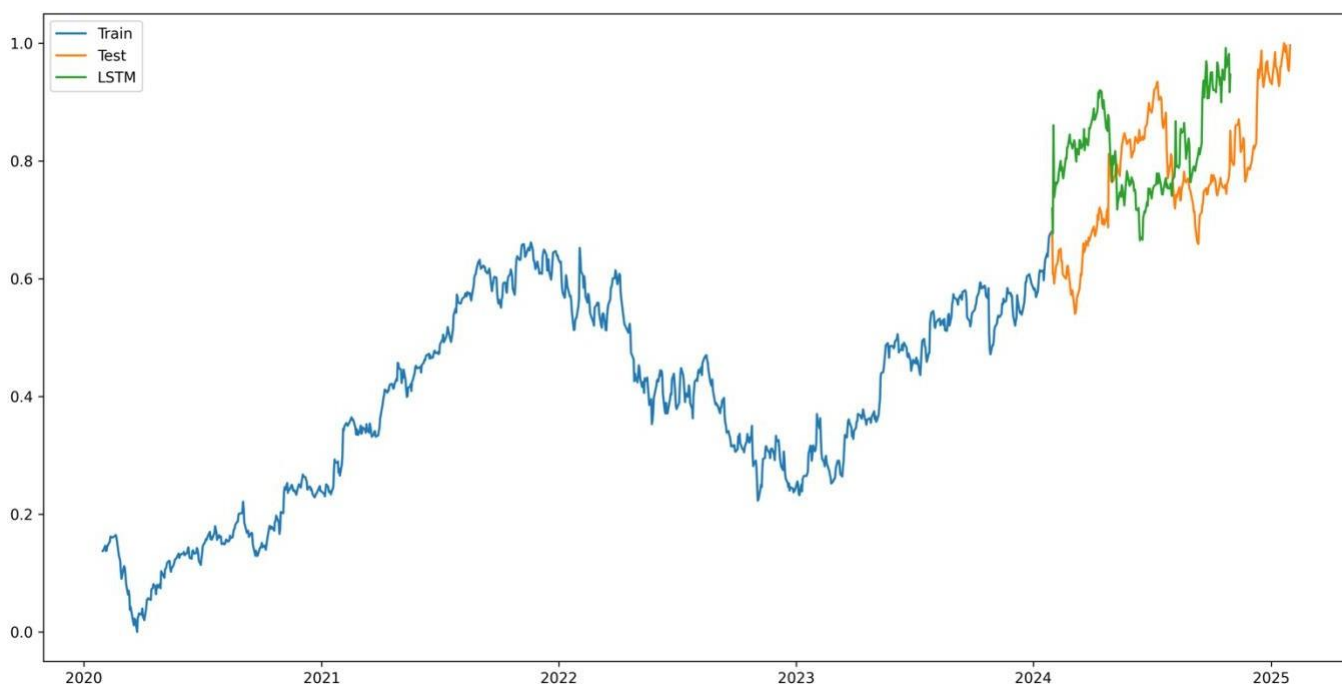


Рис. 42. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки (кількість епох 10)

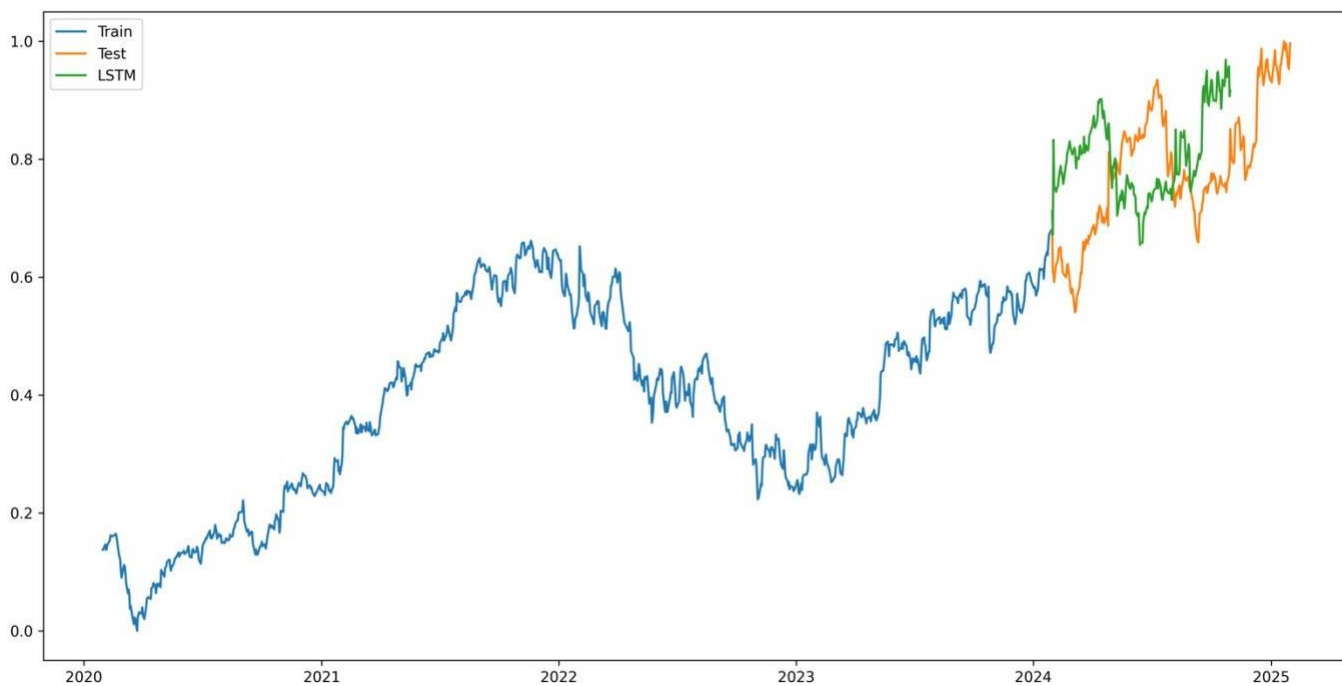


Рис. 43. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки (кількість епох 20)

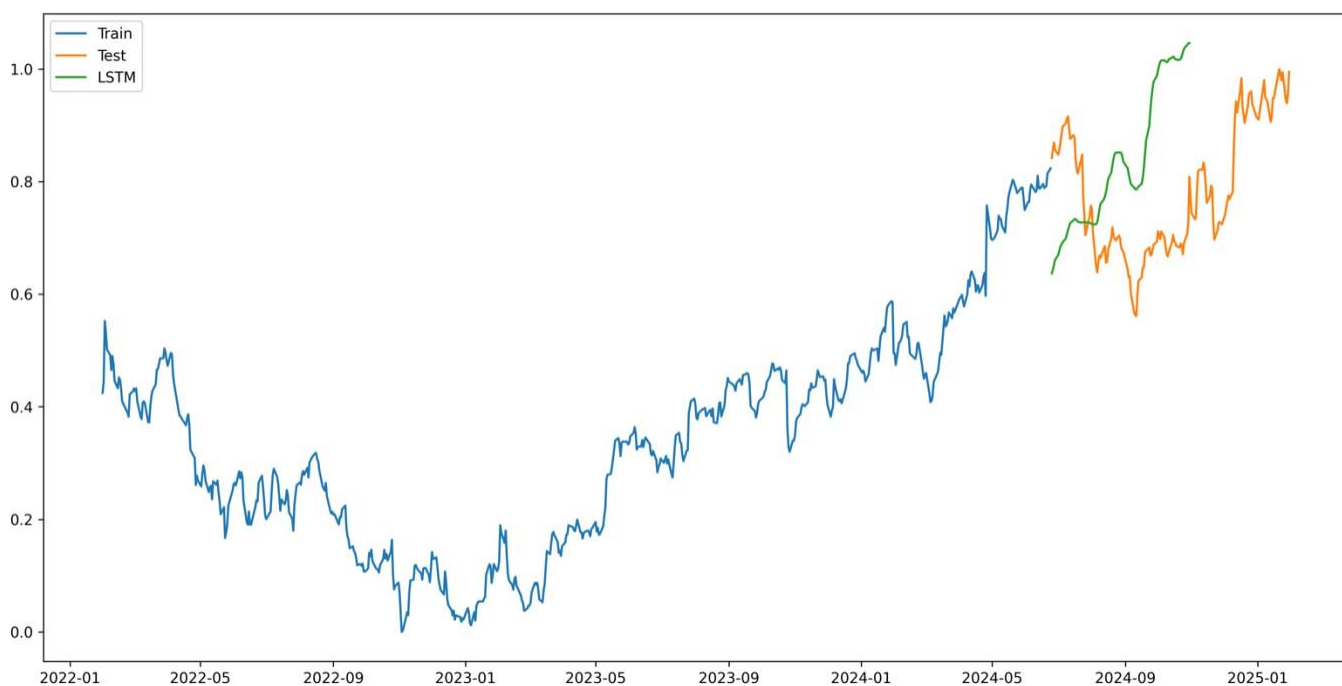


Рис. 44. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки (кількість епох 1)

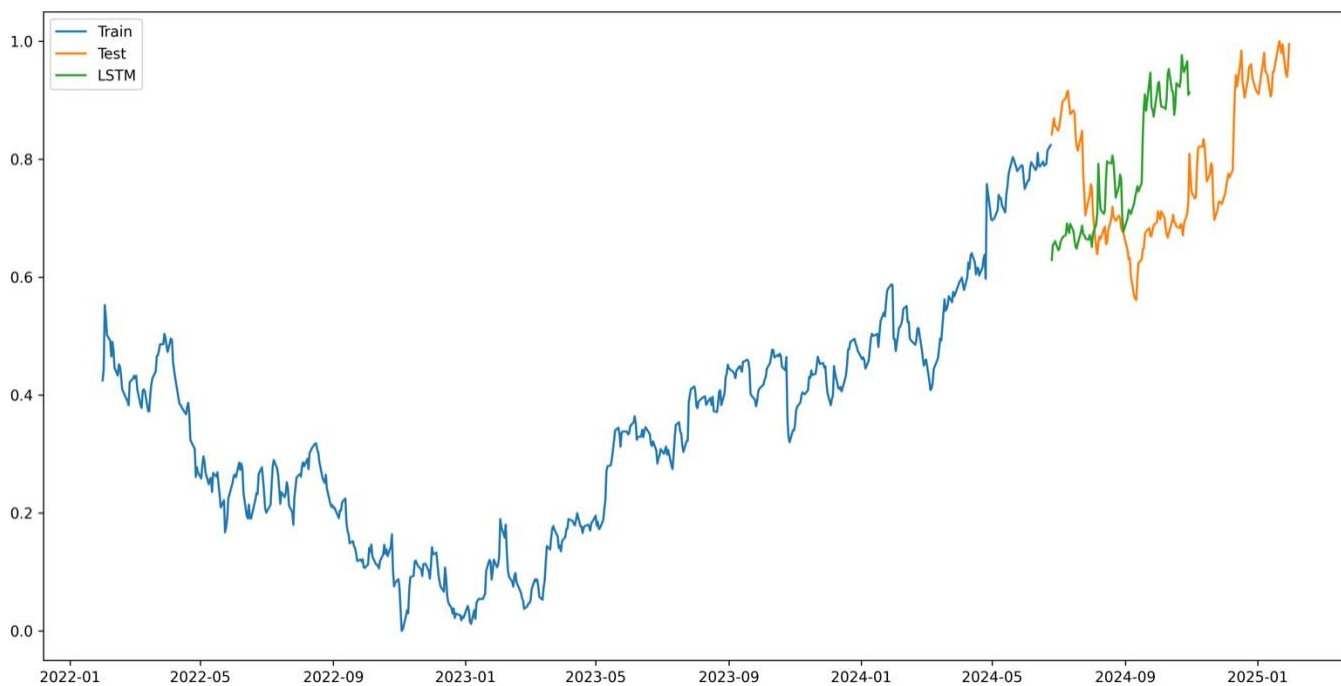


Рис. 45. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки (кількість епох 10)

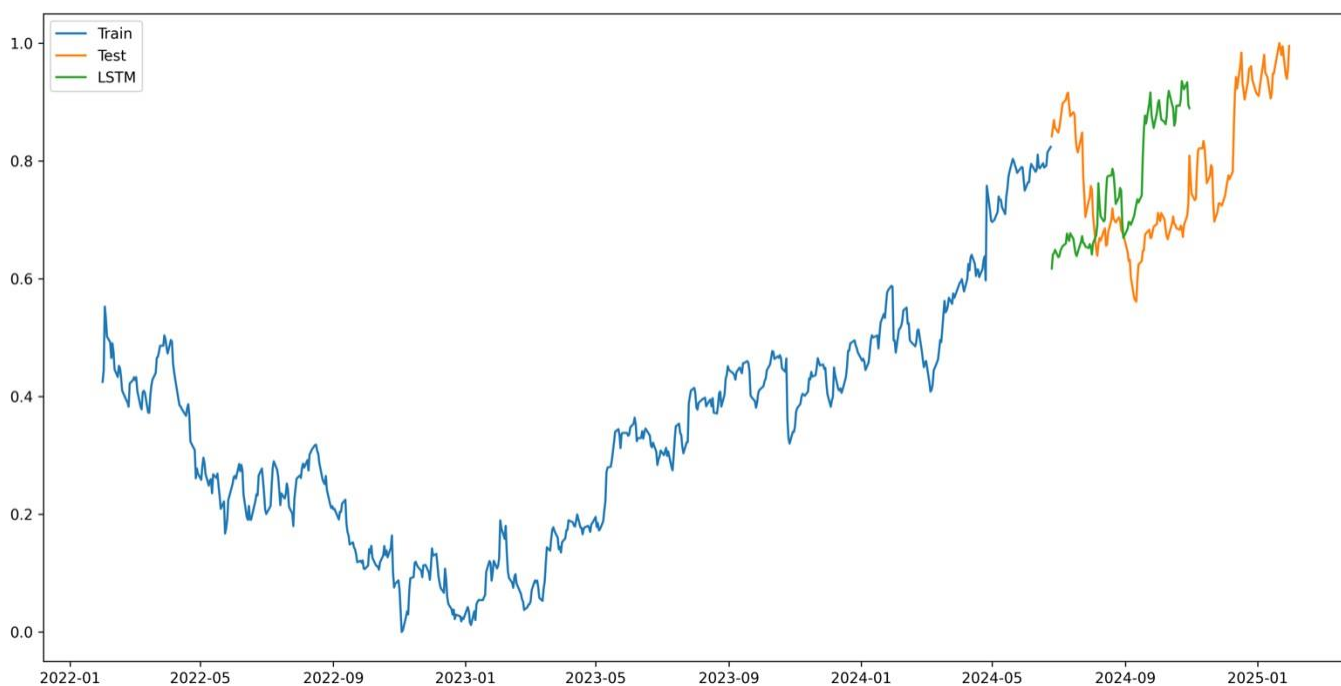


Рис. 46. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки (кількість епох 20)

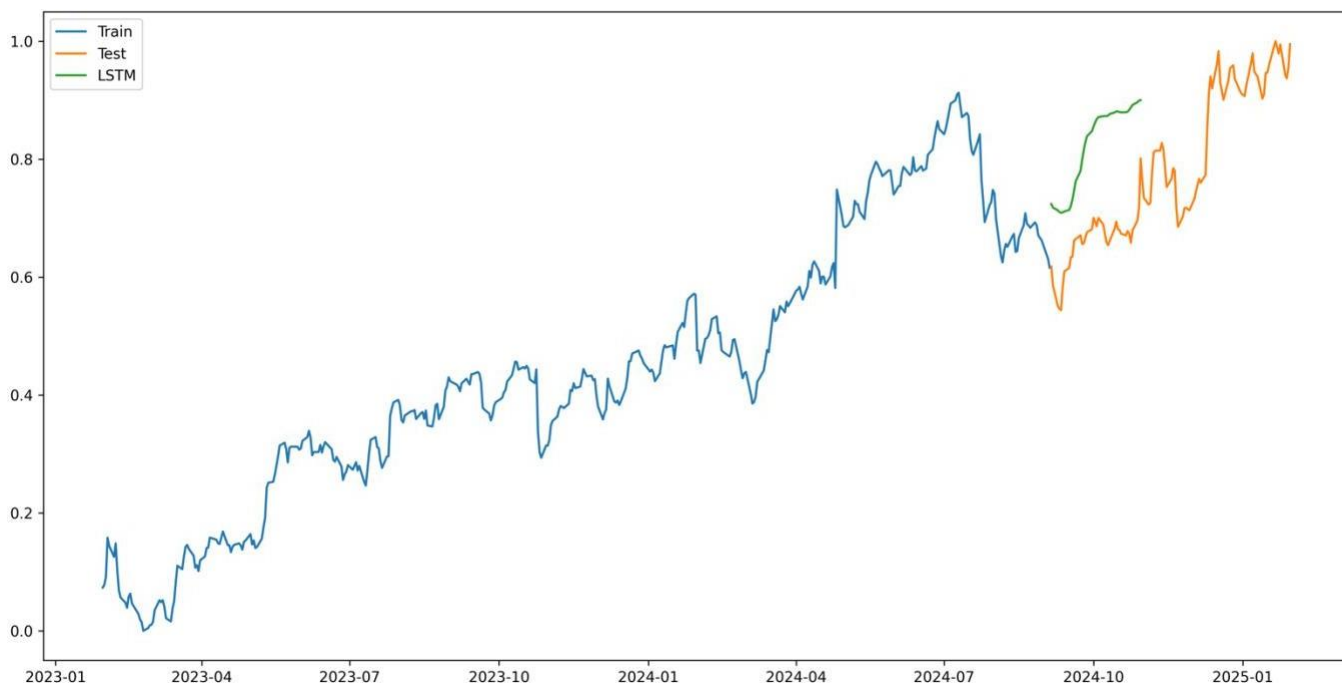


Рис. 47. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки (кількість епох 1)

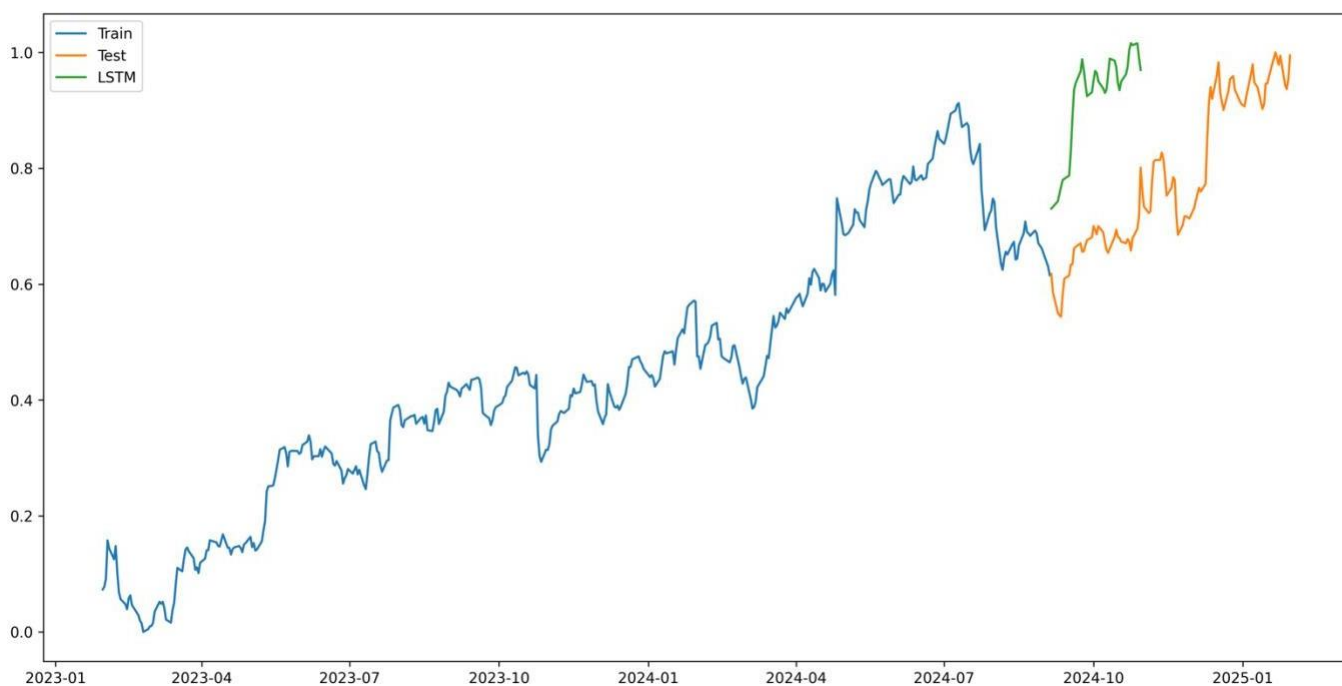


Рис. 48. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки (кількість епох 10)

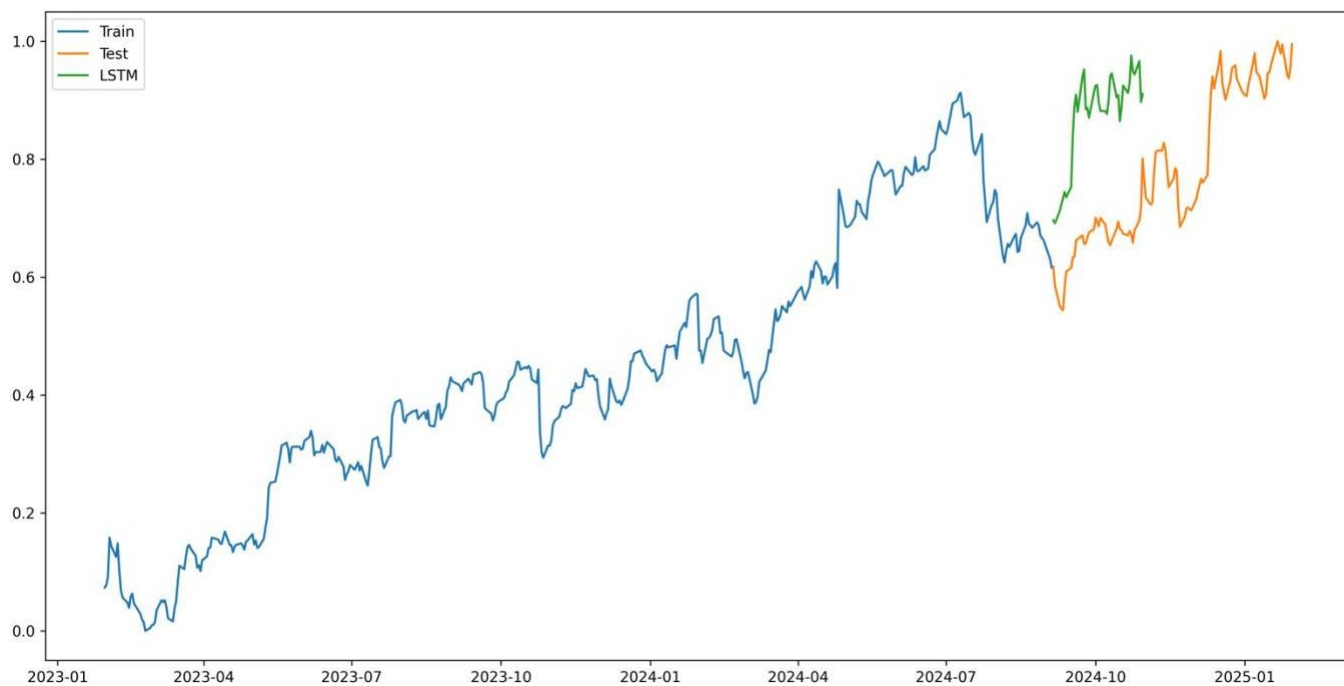


Рис. 49. Реалізація моделі довго-коротко строкової пам'яті для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки (кількість епох 20)

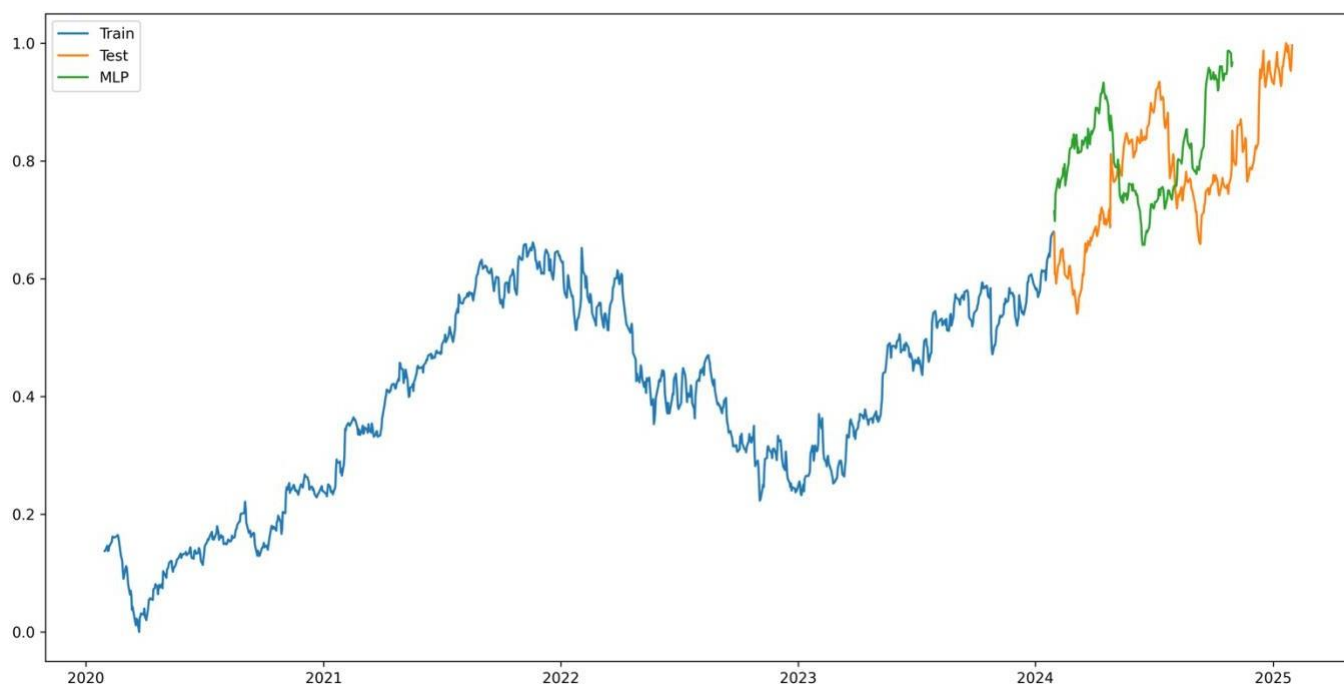


Рис. 50. Реалізація моделі багатошарового парцептрона Румельхарта для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки (кількість епох 50)

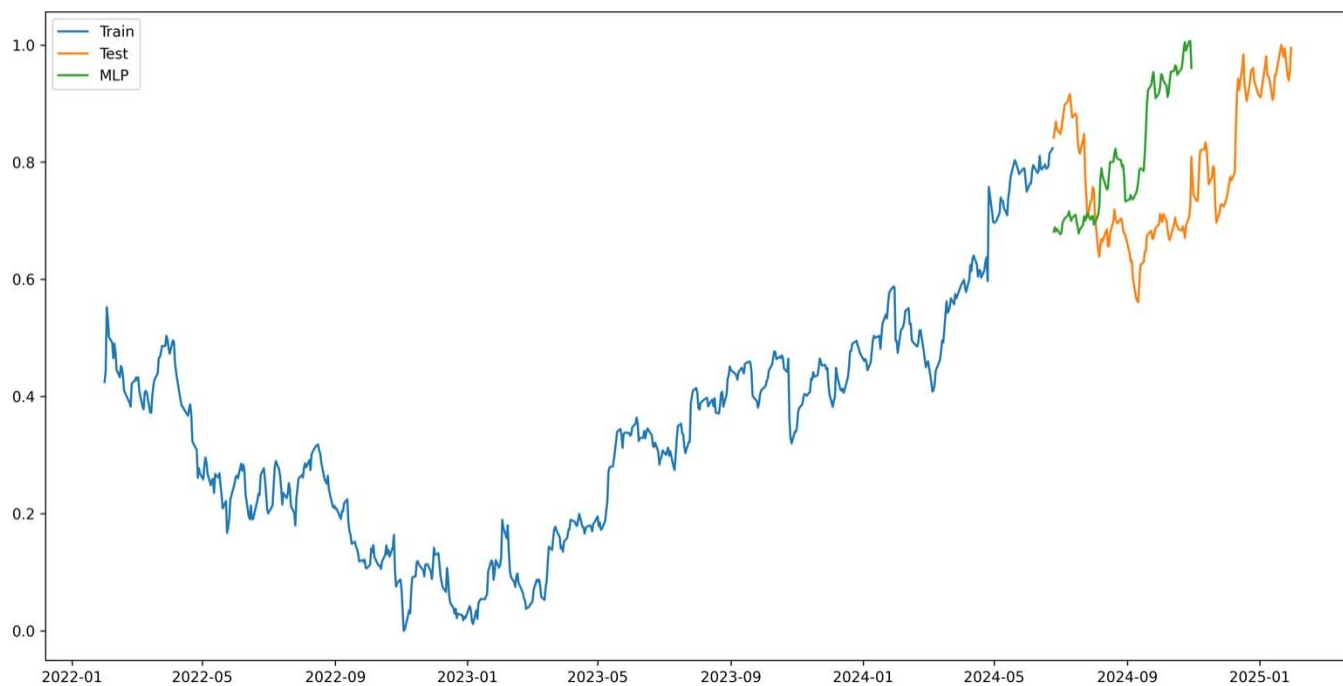


Рис. 51. Реалізація моделі багатошарового парцептрона Румельхарта для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки (кількість епох 50)

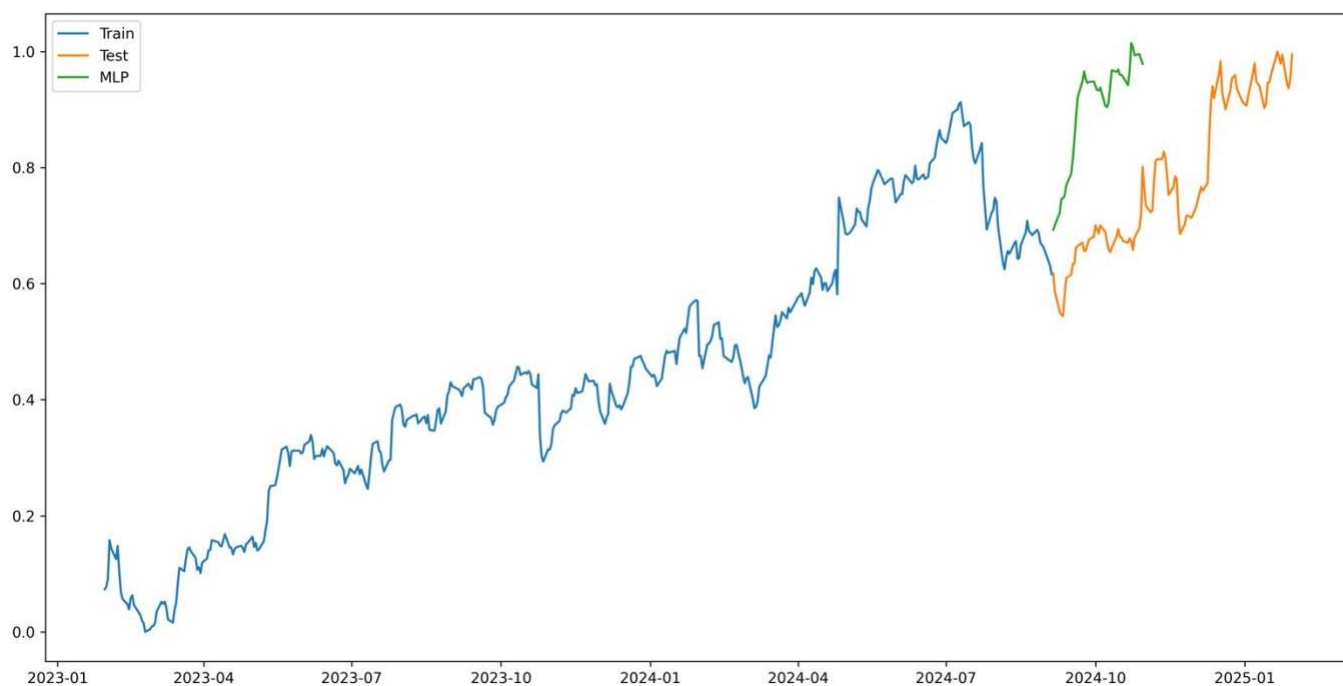


Рис. 52. Реалізація моделі багатошарового парцептрона Румельхарта для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки (кількість епох 50)

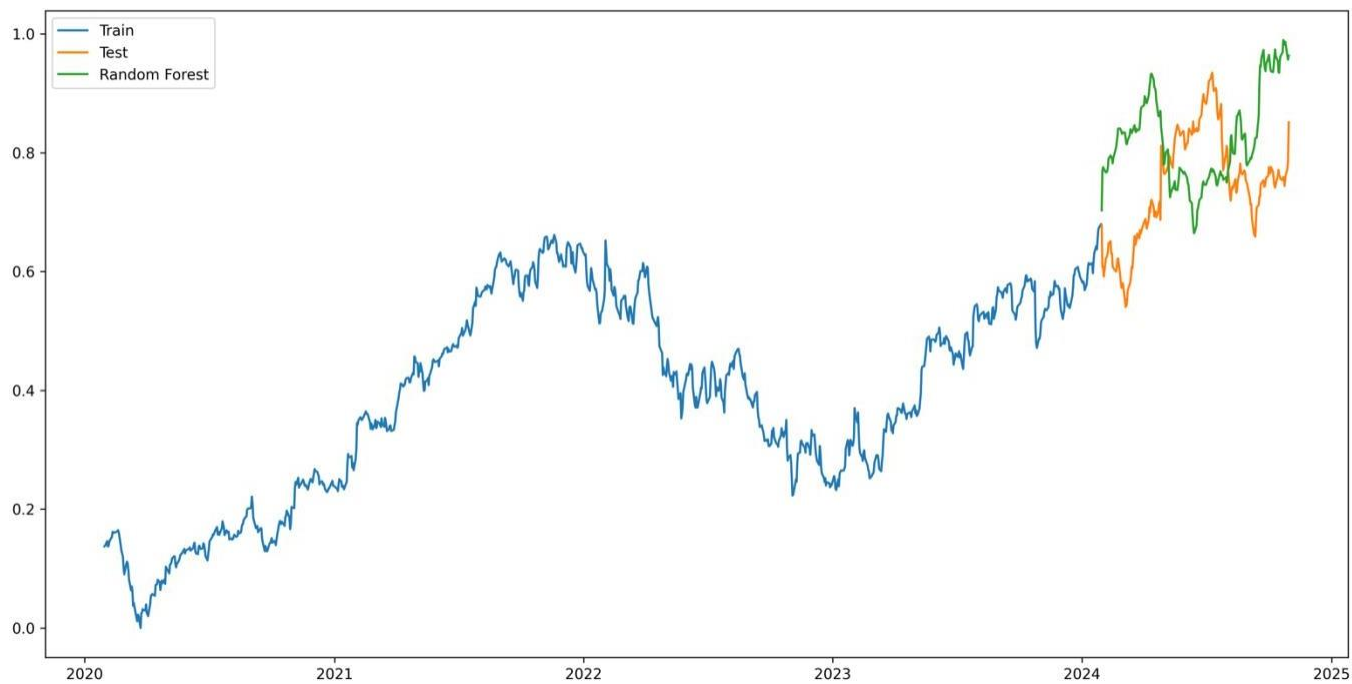


Рис. 53. Реалізація моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки

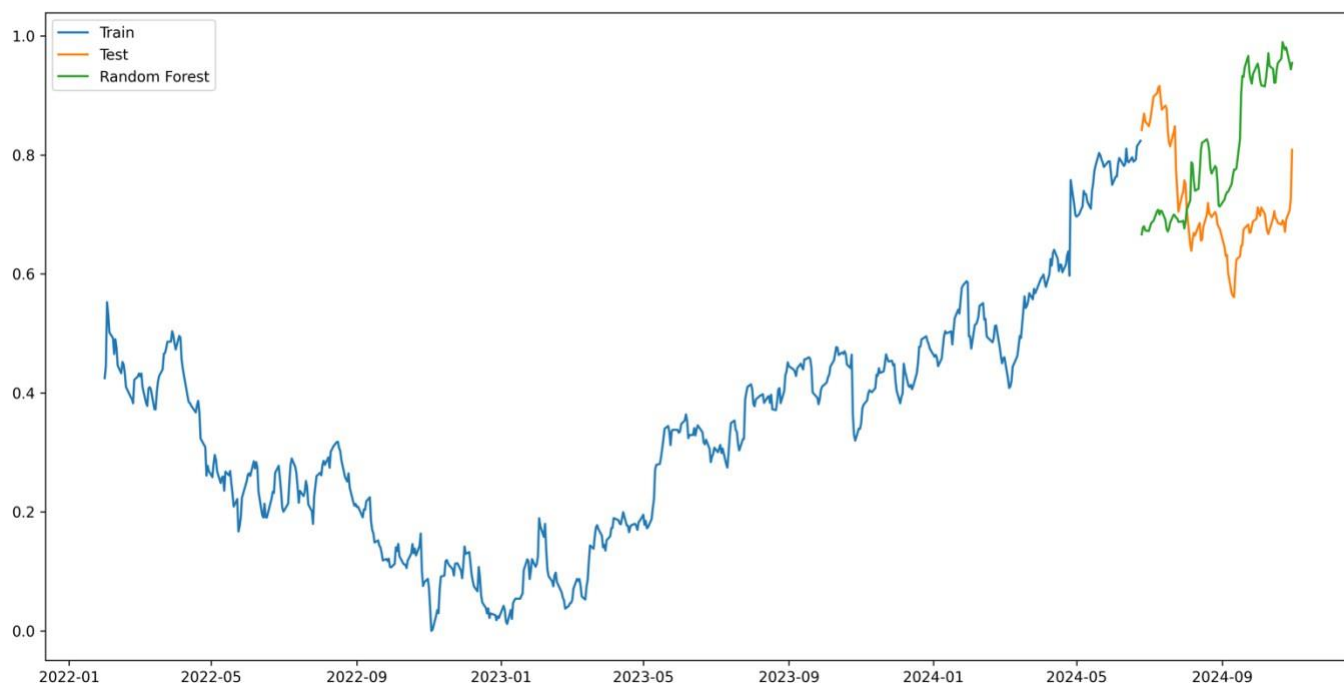


Рис. 54. Реалізація моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки

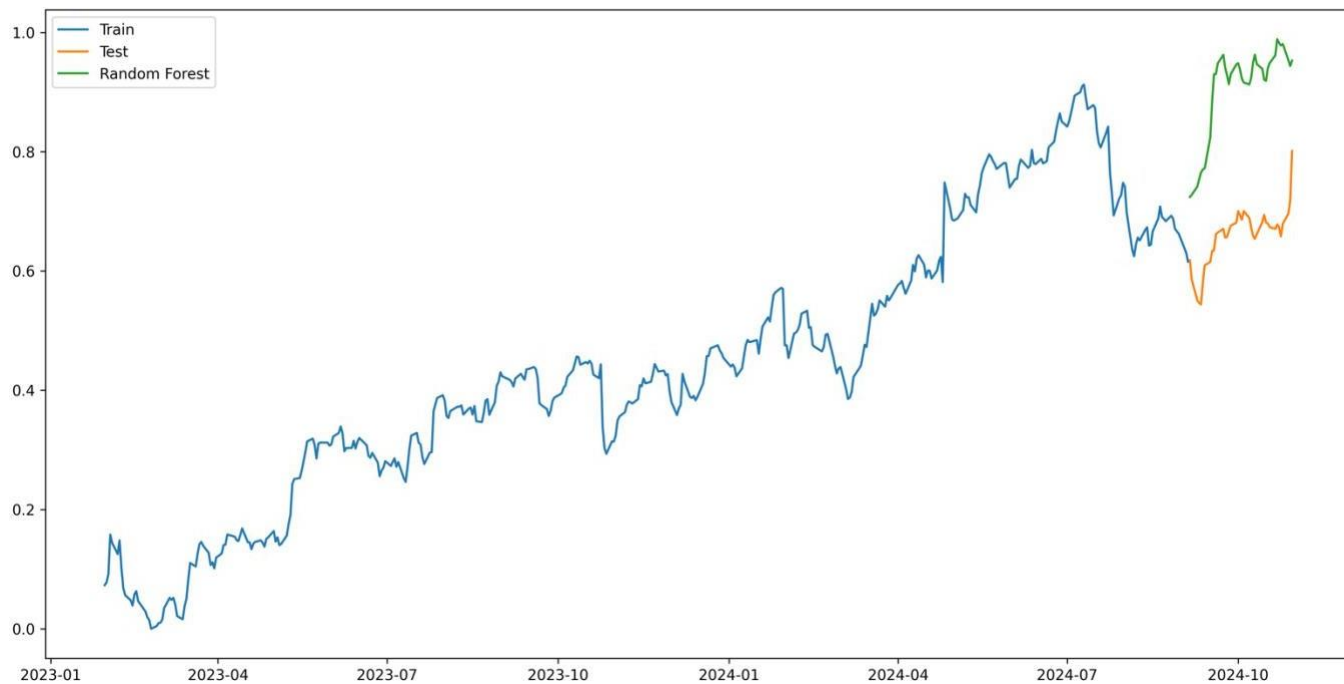


Рис. 55. Реалізація моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки

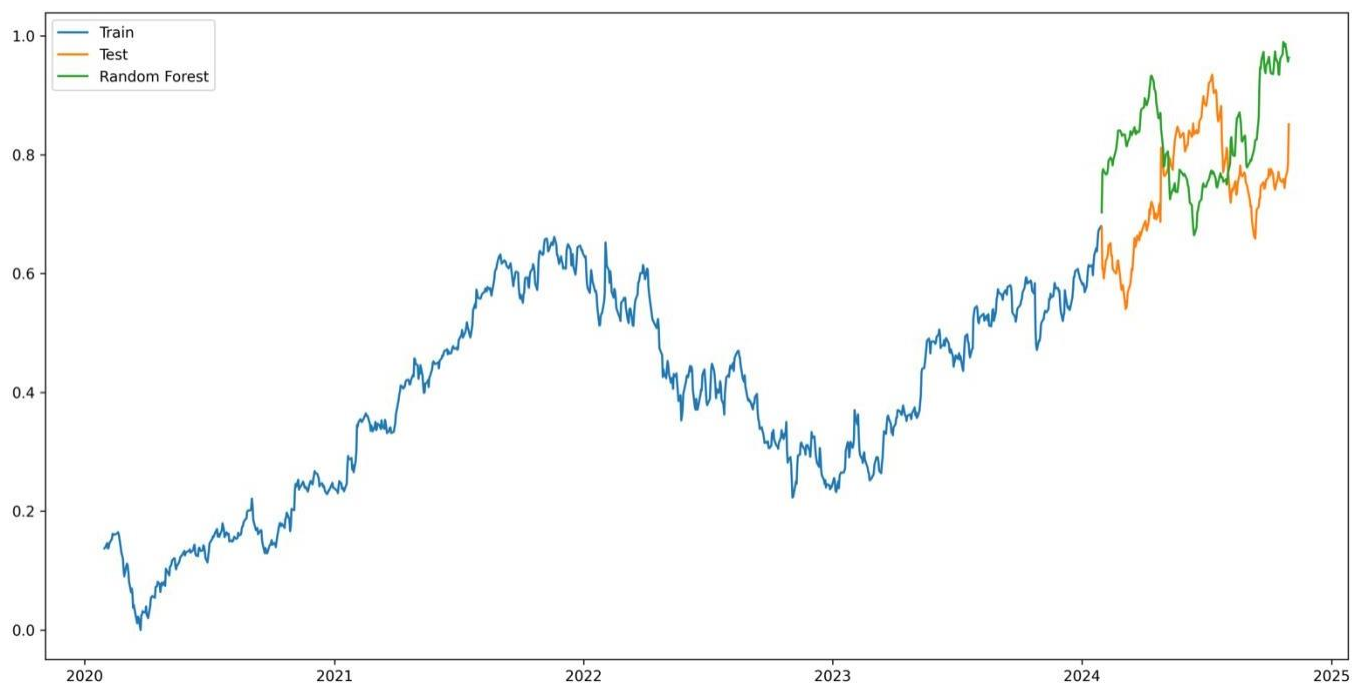


Рис. 56. Реалізація методу рекурсивного усунення ознак на основі моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 5-річної вибірки

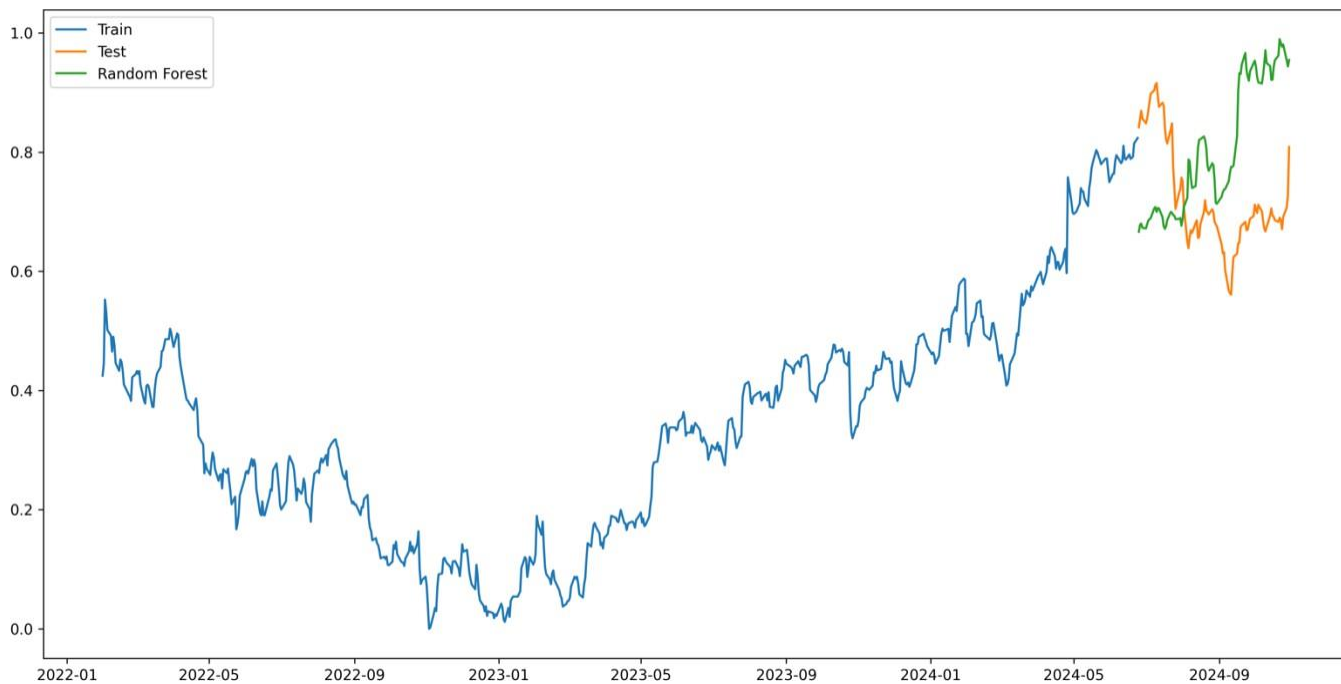


Рис. 57. Реалізація методу рекурсивного усунення ознак на основі моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 3-річної вибірки

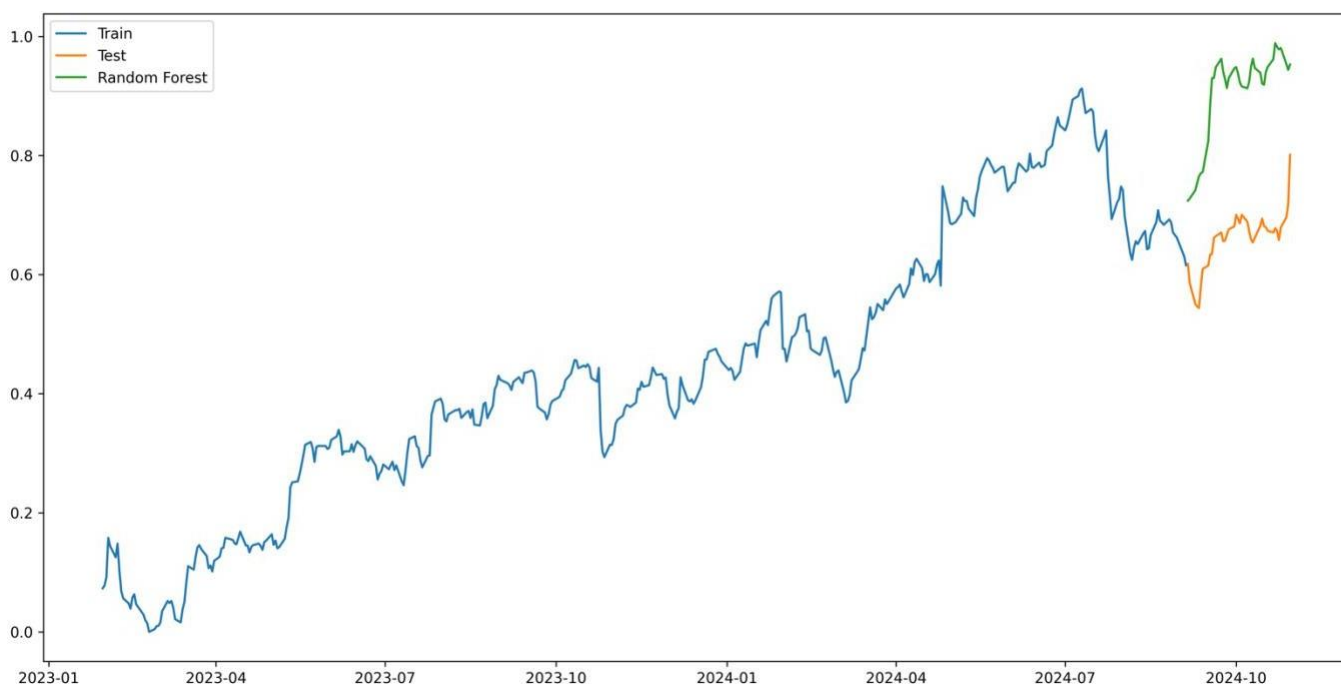


Рис. 58. Реалізація методу рекурсивного усунення ознак на основі моделі випадкового лісу для моделювання майбутньої вартості акцій компанії Google LLC на основі 2-річної вибірки

## ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ ДЛЯ РЕАЛІЗАЦІЇ МОДЕЛЕЙ

```

from set_data import DataProcessing
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, mean_squared_log_error, \
    mean_absolute_percentage_error
from pmdarima.arima import auto_arima
import statsmodels.api as sm
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing, SimpleExpSmoothing, Holt
from keras.models import Sequential
from keras.layers import Dense, LSTM
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
import numpy as np

class Predictions:

    def __init__(self, dataset):
        self.dataset = dataset
        self.train, self.test = DataProcessing().split_data(dataset)

    def forecast_errors(self, test_data, real_data):
        mse = mean_squared_error(test_data, real_data)
        mae = mean_absolute_error(test_data, real_data)
        r2 = r2_score(test_data, real_data)
        # mslge = mean_squared_log_error(test_data, real_data)
        mape = mean_absolute_percentage_error(test_data, real_data)
        metrics = {
            'Metric': ['MSE', 'MAE', 'R2', 'MAPE'],
            'Value': [mse, mae, r2, mape]
        }
        return ('\nMSE = ', mse, '\nMAE = ', mae,
            '\nR2 = ', r2, '\nMAPE = ', mape, '\n'), pd.DataFrame(metrics)

    def arima_parameters(self):
        y = self.dataset.average[:-1]

```

```

for el in range(1, 3):
    stepwise_model = auto_arma(y, start_p=3, start_q=1,
                              max_p=4, max_q=5, m=12,
                              start_P=0, seasonal=False,
                              d=el, D=0, trace=True,
                              error_action='ignore',
                              suppress_warnings=True,
                              stepwise=True)
    return stepwise_model.aic()

def set_arma(self, order=(1, 1, 1)):
    model = sm.tsa.statespace.SARIMAX(self.dataset.average,
                                       order=tuple(order),
                                       seasonal_order=(0, 0, 0, 12),
                                       enforce_stationarity=False,
                                       enforce_invertibility=False)

    results_ar = model.fit()
    pred_ar = results_ar.get_prediction(start=len(self.train), dynamic=False)
    pred_ci = pred_ar.conf_int()
    plt.figure(figsize=(16, 8))
    ax = self.train.average.plot(label='train', color='green')
    ax = self.test.average.plot(label='test', color='blue')
    pred_ar.predicted_mean.plot(ax=ax, label=f'AR {order}', alpha=0.7, color='red')

    ax.set_xlabel('Time check')
    ax.set_ylabel('Share price')
    plt.legend()
    plt.show()
    y_forecasted = pred_ar.predicted_mean
    plt.figure(figsize=(16, 8))
    print(self.forecast_errors(self.test.average, y_forecasted))
    return None

def moving_average(self, window=[30, 10, 5]):
    plt.plot(self.train['average'], label='Train')
    plt.plot(self.test['average'], label='Test')
    for w in window:
        y_hat = ((self.dataset['average'].rolling(w).mean()).iloc[:-1])
        plt.plot(y_hat[:len(self.test)], label='Moving average window = ' + str(w))
        print(self.forecast_errors(self.test.average, y_hat[:len(self.test)]))

```

```
plt.legend(loc='best')
plt.show()
return None
```

```
def simple_smoothing(self, parameters=[0.1, 0.6, 0.9]):
    plt.plot(self.train['average'], label='Train')
    plt.plot(self.test['average'], label='Test')
    for p in parameters:
        fit1 = SimpleExpSmoothing(self.dataset['average']).fit(smoothing_level=p, optimized=False)
        y_hat = fit1.fittedvalues
        plt.plot(y_hat[len(self.train):], label='Simple exp smoothing level=' + str(p))
        print(self.forecast_errors(self.test.average, y_hat[len(self.train):]))
    plt.legend(loc='best')
    plt.show()
    return None
```

```
def double_smoothing(self, parameters=[[0.2, 0.1], [0.3, 0.5], [0.9, 0.9]]):
    plt.plot(self.train['average'], label='Train')
    plt.plot(self.test['average'], label='Test')
    for p, s in parameters:
        fit1 = Holt(self.dataset['average']).fit(smoothing_level=p, smoothing_trend=s)
        y_hat = fit1.fittedvalues
        plt.plot(y_hat[len(self.train):], label='Double_exp_level=' + str(p) + 'trend=' + str(s))
        print(self.forecast_errors(self.test.average, y_hat[len(self.train):]))
    plt.legend(loc='best')
    plt.show()
    return None
```

```
def holt_winters(self, parameters=[[10, 'add', 'add']]):
    plt.plot(self.train['average'], label='Train')
    plt.plot(self.test['average'], label='Test')
    for p, tr, seas in parameters:
        fit1 = ExponentialSmoothing(self.dataset['average'], seasonal_periods=p, trend=tr, seasonal=seas).fit()
        y_hat = fit1.fittedvalues
        plt.plot(y_hat[len(self.train):], label=f"Holt-Winters periods = {p} trend = {tr} seasonal = {seas}")
        print(self.forecast_errors(self.test.average, y_hat[len(self.train):]))
    plt.legend(loc='best')
    plt.show()
    return None
```

```

def linear_regression(self, data):
    data['average'] = (data['High'] + data['Low']) / 2
    x1 = data.average.values
    x2 = data.Open.values
    x3 = data.High.values
    x4 = data.Low.values
    x5 = data.Volume.values
    min_max_scaler = preprocessing.MinMaxScaler()
    x1_scaled = min_max_scaler.fit_transform(x1.reshape(-1, 1))
    x2_scaled = min_max_scaler.fit_transform(x2.reshape(-1, 1))
    x3_scaled = min_max_scaler.fit_transform(x3.reshape(-1, 1))
    x4_scaled = min_max_scaler.fit_transform(x4.reshape(-1, 1))
    x5_scaled = min_max_scaler.fit_transform(x5.reshape(-1, 1))
    data['average'] = x1_scaled
    data['Open'] = x2_scaled
    data['High'] = x3_scaled
    data['Low'] = x4_scaled
    data['Volume'] = x5_scaled
    data['previous_average'] = data['average'].shift(1)
    data['Price Change'] = data['average'] - data['previous_average']
    data = data.dropna()
    X = data[['previous_average', 'Price Change', 'Volume', 'Low', 'High', 'Open']]
    y = data[['average']]

    poly = PolynomialFeatures(degree=2)
    X_poly = poly.fit_transform(X)

    X_train, X_test, y_train, y_test = train_test_split(X_poly, y, test_size=0.2, random_state=42, shuffle=False)
    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(Predictions(data).forecast_errors(y_test, y_pred))
    plt.plot(y_train.index, y_train, label='Train')
    plt.plot(y_test.index, y_test, label='Test')
    plt.plot(y_test.index, y_pred, label='Polynomial')
    plt.legend()
    plt.show()
    return None

```

```

def LSTM_model(self, data):
    data['average'] = (data['High'] + data['Low']) / 2
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaled_data = scaler.fit_transform(data[['average']])

    def create_dataset(data, time_step=1):
        X, Y = [], []
        for i in range(len(data) - time_step - 1):
            a = data[i:(i + time_step), 0]
            X.append(a)
            Y.append(data[i + time_step, 0])
        return np.array(X), np.array(Y)

    time_step = 60
    X, Y = create_dataset(scaled_data, time_step)
    X = X.reshape(X.shape[0], X.shape[1], 1)

    model = Sequential()
    model.add(LSTM(50, return_sequences=True, input_shape=(time_step, 1)))
    model.add(LSTM(50, return_sequences=False))
    model.add(Dense(25))
    model.add(Dense(1))

    model.compile(optimizer='adam', loss='mean_squared_error')
    train_size = int(len(scaled_data) * 0.8)
    test_size = len(scaled_data) - train_size
    train_data, test_data = scaled_data[0:train_size, :], scaled_data[train_size:len(scaled_data), :]
    dates = data.index
    train_dates, test_dates = dates[0:train_size], dates[train_size:len(scaled_data)]
    X_train, Y_train = create_dataset(train_data, time_step)
    X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)

    model.fit(X_train, Y_train, batch_size=1, epochs=1)
    X_test, Y_test = create_dataset(test_data, time_step)
    X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

    predictions = model.predict(X_test)

    plt.plot(train_dates, train_data, label='Train')
    plt.plot(test_dates, test_data, label='Test')

```

```

plt.plot(test_dates[:len(predictions)], predictions, label='LSTM')
plt.legend(loc='best')
plt.show()
print(Predictions(data).forecast_errors(test_data[:len(predictions)], predictions))
return None

```

```
def MLP_model(self, data):
```

```

    data['average'] = (data['High'] + data['Low']) / 2
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaled_data = scaler.fit_transform(data[['average']])

```

```
def create_dataset(data, time_step=1):
```

```

    X, Y = [], []
    for i in range(len(data) - time_step - 1):
        a = data[i:(i + time_step), 0]
        X.append(a)
        Y.append(data[i + time_step, 0])
    return np.array(X), np.array(Y)

```

```
time_step = 60
```

```
X, Y = create_dataset(scaled_data, time_step)
```

```
X = X.reshape(X.shape[0], X.shape[1], 1)
```

```
model = Sequential()
```

```
model.add(Dense(128, input_dim=time_step, activation='relu'))
```

```
model.add(Dense(64, activation='relu'))
```

```
model.add(Dense(1))
```

```
model.compile(loss='mean_squared_error', optimizer='adam')
```

```
train_size = int(len(scaled_data) * 0.8)
```

```
test_size = len(scaled_data) - train_size
```

```
train_data, test_data = scaled_data[0:train_size, :], scaled_data[train_size:len(scaled_data), :]
```

```
dates = data.index
```

```
train_dates, test_dates = dates[0:train_size], dates[train_size:len(scaled_data)]
```

```
X_train, Y_train = create_dataset(train_data, time_step)
```

```
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
```

```
X_test, Y_test = create_dataset(test_data, time_step)
```

```
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
```

```

model.fit(X_train, Y_train, epochs=50, batch_size=32, validation_data=(X_test, Y_test), verbose=1)
predictions = model.predict(X_test)

```

```

plt.plot(train_dates, train_data, label='Train')
plt.plot(test_dates[:len(predictions)], test_data[:len(predictions)], label='Test')
plt.plot(test_dates[:len(predictions)], predictions, label='MLP')
plt.legend(loc='best')
plt.show()
print(Predictions(data).forecast_errors(test_data[:len(predictions)], predictions))
return None

```

```

def random_forest(self, data):
    data['average'] = (data['High'] + data['Low']) / 2
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaled_data = scaler.fit_transform(data[['average']])

```

```

def create_dataset(data, time_step=1):
    X, Y = [], []
    for i in range(len(data) - time_step - 1):
        a = data[i:(i + time_step), 0]
        X.append(a)
        Y.append(data[i + time_step, 0])
    return np.array(X), np.array(Y)

```

```

time_step = 60
X, Y = create_dataset(scaled_data, time_step)
X = X.reshape(X.shape[0], X.shape[1], 1)
train_size = int(len(scaled_data) * 0.8)
test_size = len(scaled_data) - train_size
train_data, test_data = scaled_data[0:train_size, :], scaled_data[train_size:len(scaled_data), :]
dates = data.index
train_dates, test_dates = dates[0:train_size], dates[train_size:len(scaled_data)]
X_train, Y_train = create_dataset(train_data, time_step)

X_test, Y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], -1)
X_test = X_test.reshape(X_test.shape[0], -1)
X = X.reshape(X.shape[0], -1)

```

```

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X, Y)
predictions = model.predict(X_test)

plt.plot(train_dates, train_data, label='Train')
plt.plot(test_dates[:len(predictions)], test_data[:len(predictions)], label='Test')
plt.plot(test_dates[:len(predictions)], predictions, label='Random Forest')
plt.legend(loc='best')
plt.show()
print(Predictions(data).forecast_errors(test_data[:len(predictions)], predictions))

```

## ДОДАТОК В. ЛІСТИНГ ПРОГРАМИ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ЧАСОВОГО РЯДУ

```

import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
import yfinance as yf
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose

class DataProcessing:

    def get_from_computer(self, data_path):
        stock_data = pd.read_csv(str(data_path))
        stock_data = stock_data.iloc[::-1]
        return stock_data

    def get_online(self, ticker, start_date):
        end_date = '2023-09-09'
        stock_data = yf.download(ticker, start=start_date, end=end_date)
        return stock_data

```

```

def split_data(self, stock_data):
    plt.close()
    try:
        stock_data['average'] = 0.5 * (stock_data['High'] + stock_data['Low'])
    except:
        stock_data['average'] = 0.5 * (stock_data['High'].map(lambda x: float(x[1:])) + stock_data['Low'].map(lambda x:
float(x[1:]))
    x = stock_data.average.values
    min_max_scaler = preprocessing.MinMaxScaler()
    x_scaled = min_max_scaler.fit_transform(x.reshape(-1, 1))
    stock_data.average = x_scaled
    n_test = int(0.8*len(stock_data))
    train = stock_data[0:n_test]
    test = stock_data[n_test:]
    return train, test

def data_analysis(self, data):
    adft = adfuller(data['average'], autolag="AIC")
    output_df = pd.DataFrame({"Values": [adft[0], adft[1], adft[2], adft[3], adft[4]['1%'], adft[4]['5%'], adft[4]['10%']] , "Metric": ["Test
Statistics", "p-value", "No. of lags used", "Number of observations used",
                                "critical value (1%)", "critical value (5%)", "critical value (10%)"]})
    autocorr_lag1 = data['average'].autocorr(lag=30)
    autocorr_lag2 = data['average'].autocorr(lag=90)
    autocorr_lag3 = data['average'].autocorr(lag=180)
    autocorr_lag4 = data['average'].autocorr(lag=270)
    decompose = seasonal_decompose(data['average'], model='additive', period=30)
    decompose.plot()
    plt.show()
    return output_df, autocorr_lag1, autocorr_lag2, autocorr_lag3, autocorr_lag4

```

## ЗАВДАННЯ

на кваліфікаційну роботу бакалавра  
студента 4 курсу спеціальності 051 «Економіка»  
освітньої програми «Економічна кібернетика»

Шуби Даниїла Дмитровича

1. Тема роботи: «Моделювання вартості акцій компанії Google за допомогою економетричних моделей та методів машинного навчання»
2. Термін завершення роботи: 02.06.2025 року
3. Попередній захист роботи: 03.06.2025 року
4. Об'єктом дослідження є часові ряди, які містять інформацію щодо щоденного середнього курсу акцій компанії GOOGLE.
5. Предмет дослідження: моделювання середньої щоденної ціни акції компанії GOOGLE за допомогою методів згладжування, економетричних моделей та штучних нейронних мереж, що використовують алгоритми штучного інтелекту.
6. Мета дослідження: порівняти прогностичні можливості досліджуваних методів прогнозування та визначити, який метод є найкращим для моделювання середньої щоденної ціни акції компанії GOOGLE.
7. Основні завдання:
  - 7.1. Дослідження компанії GOOGLE та технології її функціонування
  - 7.2. Аналіз статистичних даних
  - 7.3. Дослідження архітектури методів згладжування, економетричних моделей та методів штучних нейронних мереж.
  - 7.4. Визначення найкращого методу для моделювання середньої щоденної ціни акції компанії GOOGLE

Науковий керівник: д. е. н., професор Ставицький Андрій Володимирович

Андрій СТАВИЦЬКИЙ



Студент: Шуба Даниїл Дмитрович















Даниїл ШУБА



Задтверджено на засіданні кафедри економічної кібернетики

Протокол № 4 від 22.11.2024 р.

## КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№	Етапи кваліфікаційної роботи бакалавра	Терміни виконання	Відмітка керівника про виконання
1	Вибір теми кваліфікаційної роботи бакалавра	01.09.2024 – 04.10.2024	
2	Розробка та затвердження завдання на роботу бакалавра	12.10.2024 – 25.10.2024	
3	Розробка і узгодження з керівником календарного плану виконання кваліфікаційної роботи бакалавра	02.11.2024 – 15.11.2024	
4	Підбір літератури, інформаційних, довідкових та інших матеріалів, ознайомлення з ними та їх систематизація	23.11.2024 – 20.12.2024	
5	Створення змісту кваліфікаційної роботи бакалавра, формулювання актуальності та мети	04.01.2025 – 17.01.2025	
6	Структуризація теоретичного матеріалу та написання першого розділу	25.01.2025 – 14.02.2025	
7	Ознайомлення з методологією прогнозування часових рядів	15.02.2025 – 28.02.2025	
8	Написання другого розділу	01.03.2025 – 21.03.2025	
9	Реалізація практичної частини. Написання третього розділу	22.03.2025 – 31.03.2025	
10	Формулювання висновків	01.04.2025 – 18.04.2025	
11	Оформлення роботи відповідно до методичних правил, редагування списку використаних джерел, додатків	19.04.2025 – 30.04.2025	
12	Перевірка роботи на плагіат	17.05.2025	
13	Подання роботи до попереднього захисту	18.05.2025	
14	Остаточне коригування роботи, отримання рецензії та відгуку наукового керівника	20.05.2025 – 01.06.2025	

Науковий керівник: д. е. н., професор Ставицький Андрій Володимирович

Андрій СТАВИЦЬКИЙ 

Студент: Шуба Даниїл Дмитрович

Даниїл ШУБА 