

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту**

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи і методи штучного інтелекту»

спеціальності 122 «Комп'ютерні науки»

на тему: «Прогнозування показників фінансових ринків

з використанням систем штучного інтелекту»

Виконав:

студент ІV курсу, групи КІ-01

Скоробогатов Станіслав Юрійович _____

Керівник:

доктор техн. наук, професор

Зайченко Олена Юріївна _____

Консультант з економічного розділу:

доктор економ. наук, професор

Шевчук Олена Анатоліївна _____

Консультант з нормоконтролю:

фахівець першої категорії кафедри ШІ,

Кравець Павло Володимирович _____

Рецензент:

доктор техн. наук, професор

Мухін Вадим Євгенович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«31» січня 2024 р.

ЗАВДАННЯ

на дипломну роботу студенту

Скоробогатов Станіслав Юрійович

1. Тема роботи «Прогнозування показників фінансових ринків з використанням систем штучного інтелекту», керівник роботи Зайченко Олена Юріївна, доктор техн. наук, професор, затверджені наказом по університету від «31» травня 2024 р. № 2240
2. Термін подання студентом роботи «10» червня 2024 року.
3. Вихідні дані до роботи: історичні дані цін акцій компанії Tesla
4. Зміст роботи: аналіз предметної області дослідження, підготовка даних до аналізу, аналіз вже створених моделей для прогнозування, побудова прогнозу із використанням наявних моделей, оцінка результатів прогнозування, аналіз недоліків моделей та тестування запропонованої моделі прогнозування, пропозиція методологій для застосування нової моделі.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): аналіз та обробка набору даних, огляд та порівняння архітектури мереж, порівняння створених та запропонованої моделей, огляд результатів та недоліків у прогнозах моделей, опис областей для застосування нової моделі.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук Олена Анатоліївна, професор, д. е. н.		

7. Дата видачі завдання «05» лютого 2024 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою дослідження	12.03.2024-19.03.2024	Виконано
2	Затвердження теми дипломної роботи	20.03.2024-22.03.2024	Виконано
3	Підготовка першого розділу	23.03.2024-31.03.2024	Виконано
4	Підготовка другого розділу	01.04.2024-15.04.2024	Виконано
5	Розробка програмного продукту	16.04.2024-02.05.2024	Виконано
6	Аналіз результатів роботи, представлення нової моделі	03.05.2024-21.05.2024	Виконано
7	Підготовка економічної частини	22.05.2024-31.05.2024	Виконано
8	Підготовка презентації доповіді	01.06.2024-05.06.2024	Виконано
9	Оформлення дипломної роботи	06.06.2024-09.06.2024	Виконано

Студент

Станіслав СКОРОБОГАТОВ

Керівник

Олена ЗАЙЧЕНКО

РЕФЕРАТ

Дипломна робота: 130 с., 27 рис., 8 табл., 35 посилань, 1 додаток.

ФІНАНСОВІ РИНКИ, ФІНАНСОВІ ПОКАЗНИКИ, ЧАСОВІ РЯДИ, НЕЙРОННІ МЕРЕЖІ, ПРОГНОЗ, МЕРЕЖА ТРАНСФОРМЕР, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ.

У ході дослідження застосовано та проаналізовано результати із використанням наявних моделей для прогнозування показників фінансових ринків. Для тестування моделей обрано часовий ряд із даними акцій компанії Tesla.

Об'єкт дослідження – тенденції розвитку показників фінансових ринків.

Предмет дослідження – методологія прогнозування показників, застосування штучного інтелекту при побудові прогнозних моделей для визначення показників фінансових ринків на основі відкритих даних.

Мета роботи – аналіз методів прогнозування, розроблення надійної моделі прогнозування шляхом вивчення потенціалу систем штучного інтелекту (ШІ) у сфері прогнозування фінансових ринків та подальшому прогнозуванні показників фінансових ринків з використанням систем штучного інтелекту.

У результаті дослідження запропоновано модель прогнозування наступних значень часового ряду із використанням текстових даних на додаток до попередніх значень показників ринків. У запропонованій моделі використано переднавчену архітектуру трансформера BERT на додаток до архітектури, побудованої на основі блоків LSTM. Використання текстових даних покращило прогнозування показників у порівнянні із наявними моделями.

ABSTRACT

Master's thesis: 130 p., 27 figures, 8 tables, 35 references, 1 appendix.

FINANCIAL MARKETS, FINANCIAL INDICATORS, TIME SERIES, NEURAL NETWORKS, FORECAST, TRANSFORMER NETWORK, RECURRENT NEURAL NETWORKS, DEEP LEARNING.

In the course of the study, the results are applied and analysed using the existing models for forecasting financial market indicators. To test the models, a time series with data on Tesla shares was chosen.

The object of the study is trends in the development of financial market indicators.

The subject of research is the methodology of forecasting indicators, the use of artificial intelligence in building forecasting models to determine financial market indicators based on public data.

The purpose of the work is to analyse forecasting methods, develop a reliable forecasting model by studying the potential of artificial intelligence (AI) systems in the field of financial market forecasting and further forecasting financial market indicators using AI systems.

As a result of the study, a model for predicting the next values of a time series using textual data in addition to the previous values of market indicators is proposed. The proposed model uses the pre-trained BERT transformer architecture in addition to the architecture built on the basis of LSTM blocks. The use of textual data improved the forecasting of indicators compared to existing models.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 Основи функціонування фінансових ринків.....	12
1.1 Сутність та учасники фінансового ринку	12
1.2 Основні поняття та ключові слова.....	15
1.3 Характеристика товарів на фінансовому ринку	17
1.4 Актуальність предметної області.....	20
1.5 Постановка задачі дослідження	23
Висновки до розділу 1.....	26
РОЗДІЛ 2 Аналіз методів прогнозування фінансових ринків.....	27
2.1 Побудова інтелектуальних систем для прогнозування фінансових ринків.....	27
2.2 Дослідження існуючих інтелектуальних систем.....	31
2.2.1 Моделі ковзного середнього	33
2.2.2 Авторегресійні моделі.....	36
2.2.3 Багатошаровий перцептрон	38
2.2.4 Рекурентна нейронна мережа.....	42
2.2.5 Моделі довгої короткочасної пам'яті (LSTM).....	46
2.3 Критерії оцінки результатів та інтерпретація прогнозів	50
Висновки до розділу 2.....	53
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДЕЛІ	55
3.1 Вибір методології для прогнозування фінансових ринків	55
3.2 Робота із даними.....	58
3.2.1 Обґрунтування вибору даних.....	59
3.2.2 Підготовка даних для дослідження	60
3.3 Аналіз інтелектуальних систем для прогнозування показників	66
3.3.1 Порівняльний аналіз результатів використання наявних методик	67
3.3.2 Запропонована модель прогнозування показників фінансових ринків на основі методів штучного інтелекту	76

	7
3.4 Аналіз отриманих результатів.....	80
Висновки до розділу 3.....	81
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	83
4.1 Постановка задачі проектування	84
4.2 Обґрунтування функцій програмного продукту	84
4.3 Обґрунтування системи параметрів програмного продукту.....	87
4.4 Аналіз експертного оцінювання параметрів.....	90
4.5 Аналіз рівня якості варіантів реалізації функцій	94
4.6 Економічний аналіз варіантів розробки програмного продукту	96
4.7 Вибір кращого варіанту програмного продукту техніко-економічного рівня	102
Висновки до розділу 4.....	103
ВИСНОВКИ	104
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	110

ВСТУП

Сучаний розвиток суспільства характеризується великою кількістю технологічних досягнень та економічних змін, тому можливість прогнозування фінансових показників із високою точністю стає вкрай важливою справою, що може помітно вплинути на фінанси інвесторів і компаній. Складна взаємодія економічних змінних, геополітичних подій та очікувань інвесторів робить традиційні методи прогнозування не достатньо ефективними для створення достатньо точних прогнозів на основі динаміки змін сучасних фінансових ринків. У результаті виникає нагальна потреба в інноваційних підходах, здатних враховувати значну кількість факторів для створення прогнозів в складних фінансових системах з високою точністю та гнучкістю.

Тривалий час фінансові ринки були предметом уваги як науковців, так і практиків, виступаючи життєво важливими індикаторами економічного стану та визначальними рушіями економічного розвитку. Точне прогнозування фінансових індикаторів відіграє ключову роль у процесах прийняття рішень інвесторами, політиками та фінансовими установами. Саме тому пошук моделей прогнозування, здатних надавати високу точність прогнозу, привертає увагу фахівців із різних сфер.

Застосування штучного інтелекту є потужним інструментом, який надає можливості для виявлення закономірностей, тенденцій та прогнозування ринкових змін із високою точністю. Використовуючи великі обсяги даних, що генеруються у фінансових системах, AI-системи здатні виявляти приховані кореляції, ідентифікувати закономірності, що виникають, і прогнозувати ринкові показники з рівнем складності, що перевершує традиційні методи. Завдяки цьому прогноз із використанням моделей на основі штучного інтелекту наразі використовується не тільки представниками фінансових установ, але також бізнесами для алгоритмічної торгівлі, при якій навіть невеликі зміни у точності прогнозування дозволяють уникнути величезних втрат.

Мета дослідження полягає в аналізі методів прогнозування, розробленні надійної моделі прогнозування шляхом вивчення потенціалу систем штучного інтелекту (ШІ) у сфері прогнозування фінансових ринків та подальшому прогнозуванні показників фінансових ринків з використанням систем штучного інтелекту.

Об'єкт дослідження: тенденції розвитку показників фінансових ринків.

Предмет дослідження: методологія прогнозування показників, застосування штучного інтелекту при побудові прогнозних моделей для визначення показників фінансових ринків на основі відкритих даних.

Для досягнення поставленої мети визначено такі **завдання:**

- дослідити теоретичні основи функціонування фінансових ринків;
- проаналізувати методичні підходи до прогнозування показників з використанням існуючих інтелектуальних систем;
- обґрунтувати вибір методології для прогнозування фінансових ринків;
- відібрати дані для прогнозування, здійснити їх підготовку для дослідження;
- здійснити порівняльний аналіз результатів використання наявних методик прогнозування;
- побудувати модель прогнозування показників фінансових ринків на основі методів штучного інтелекту та проаналізувати отримані результати.

За результатами впровадження запропонованих заходів відбудеться підвищення точності та надійності прогнозів фінансових ринків шляхом використання можливостей таких методів ШІ, як машинне навчання та обробка природної мови (NLP). Для створення моделі, яка надає змогу отримати високу точність прогнозу, проведено теоретичний аналіз, дослідження вже існуючих систем, щоб з'ясувати потенціал моделей прогнозування у формуванні прогнозних показників фінансових ринків, а також представлено порівняння із прогнозуванням із використанням наявних методів, завдяки чому підтверджено

продуктивність запропонованої архітектури, яка дозволить значно підвищити точність прогнозу показників фінансових ринків, які залежать не тільки від попередніх показників прогнозу, але й від інших зовнішніх подій у світі.

Під час виконання роботи були використані наступні методи дослідження: поєднання історичного та логічного методів при дослідженні сутності та основних характеристик фінансового ринку; методи критичного аналізу при порівнянні методів прогнозування фінансових ринків; узагальнення та систематизації при виборі методології для прогнозування фінансових ринків; навчання та тестування для розробки моделі прогнозування показників фінансових ринків, а також аналізу отриманих результатів; методи функціонально-вартісного аналізу було застосовано для вибору та обґрунтування кращого варіанту програмного продукту.

Наукова новизна результатів дипломної роботи полягає у тому, що була запропонована модель прогнозування показників фінансових ринків на основі методів штучного інтелекту, що враховує не лише кількісні параметри, але й аналіз текстових даних, що збільшує точність прогнозування.

Практичне значення роботи полягає у розробленні програмного продукту для прогнозування показників фінансових ринків на основі методів штучного інтелекту, що може бути використаний в роботі фінансових установ, бізнес-менеджерами, інвесторами та іншими учасниками фінансового ринку, оскільки дозволяє отримати більш точні прогнозні дані.

Результати проведеного дослідження впроваджено у роботу ТОВ «Аваліс» (довідка про впровадження № 4819 від 15 травня 2024 року). Основні положення роботи були представлені на міжнародних конференціях: IV Міжнародна науково-практична конференція “MODERN RESEARCH IN SCIENCE AND EDUCATION” (7-9.12.2023 Чикаго, США), III Міжнародна науково-практична конференція “INNOVATIVE DEVELOPMENT OF SCIENCE, TECHNOLOGY AND EDUCATION” (14-16.12.2023 Ванкувер, Канада) та подана до публікації стаття у фаховому виданні України.

Структура роботи: робота складається зі вступу, чотирьох основних розділів, висновків, списку використаних джерел, додатку.

РОЗДІЛ 1 ОСНОВИ ФУНКЦІОНУВАННЯ ФІНАНСОВИХ РИНКІВ

1.1 Сутність та учасники фінансового ринку

Фінансові ринки виступають основою сучасної економіки, сприяючи руху капіталу між його власниками та позичальниками, дозволяючи цим самим бізнесу розвиватися та впроваджувати інновації. Дані ринки є своєрідною платформою для визначення цін та управління ризиками.

За своєю суттю фінансові ринки виступають посередниками між тими, хто має надлишок коштів (заощадниками), і тими, хто потребує фінансування (позичальниками). Вони надають позичальникам механізм доступу до капіталу, а заощадникам можливість отримувати дохід від своїх інвестицій. Така взаємодія між заощадниками та позичальниками має важливе значення для економічного зростання та розвитку.

Фінансові ринки виконують важливі функції у забезпеченні ефективного функціонування економіки, а саме:

- діють як механізми ціноутворення, визначаючи справедливую вартість фінансових активів через взаємодію попиту та пропозиції;
- сприяють ефективному розподілу капіталу від заощадників до позичальників;
- забезпечують механізми управління ризиками, які дозволяють суб'єктам фінансових ринків зменшувати або передавати ризики, які вони не можуть або не бажають нести;
- забезпечують ліквідність активів, надаючи можливість їх легко продавати та конвертувати у грошові кошти;
- виступають в якості інформаційних центрів, поширюючи ринкові дані, економічні показники та корпоративні новини серед інвесторів та компаній.

Аналіз наукових джерел інформації [0] дозволив визначити основні типи фінансових ринків:

- фондові ринки, які виступають центрами купівлі-продажу акцій публічних компаній, що представляють частки власності в цих компаніях;
- ринки облігацій, які забезпечують платформу для організацій, таких як уряди та корпорації, для випуску облігацій з метою залучення капіталу, оскільки інвестори можуть купувати та торгувати цими кредитними інструментами;
- ринки деривативів, на яких торгують деривативами, тобто фінансовими контрактами (похідними цінними паперами), вартість яких залежить від результатів діяльності базових активів;
- товарні ринки, що сприяють торгівлі фізичними товарами, такими як сільськогосподарська продукція, метали та енергетичні ресурси; такі ринки надають виробникам і споживачам можливості для визначення цін та захисту від ризиків, забезпечуючи безперебійний потік цих важливих товарів;
- валютні ринки, що поєднують покупців і продавців валют, надаючи можливість обмінювати одну валюту на іншу;
- ринки криптовалюти, які з'явилися на хвилі зростання біткойна та інших криптовалют і зараз активно розвиваються; такі ринки сприяють торгівлі цими цифровими активами, надаючи інвесторам платформу для участі в криптовалютній екосистемі [2].

Основними суб'єктами (учасниками) фінансових ринків є інвестори, фінансові установи, регулятори та посередники.

Інвестори надають компаніям, урядам та іншим суб'єктам капітал, необхідний для фінансування їхньої діяльності та інвестицій, вони є основою фінансових ринків. Інвестори – це фізичні особи, пенсійні фонди та страхові компанії, підприємства та організації, уряди та міжнародні організації. Вони використовують різноманітні фактори, такі як схильність до ризику, очікування доходу та стан ринку, щоб зробити припущення щодо вигідності купівлі, продажу або збереження фінансових активів [3].

Фінансові установи є ключовими гравцями на фінансових ринках, виступаючи посередниками між позичальниками та кредиторами і надаючи основні послуги, такі як кредитування, платіжні системи та інвестиційні консультації. До основних фінансових установ належать комерційні банки, інвестиційні банки, компанії з управління активами та страхові компанії. Завдяки різноманітним фінансовим продуктам і послугам вони сприяють накопиченню заощаджень, спрямовують кошти на продуктивне використання та допомагають управляти фінансовими ризиками [4].

Регулятори здійснюють нагляд за діяльністю учасників ринку та забезпечують дотримання правил і норм для забезпечення справедливих і прозорих ринків, виступаючи в ролі охоронців фінансової стабільності та доброчесності. Вони встановлюють регуляторні стандарти, здійснюють нагляд за ринком та впроваджують політику, спрямовану на підтримку ефективності ринку та захист інтересів інвесторів, включаючи центральні банки, комісії з цінних паперів та фінансові регулятори. Їхні дії мають вирішальне значення для забезпечення стабільності та стійкості фінансової системи [5].

Посередники на фінансовому ринку сприяють безперебійному функціонуванню ринкових операцій та підвищують ліквідність, виступаючи посередниками між покупцями та продавцями на фінансових ринках. Прикладом посередників можуть бути брокерські фірми, біржі та маркет-мейкери. Вони надають платформи для торгівлі цінними паперами, зводять покупців і продавців та забезпечують виявлення цін, а також відіграють важливу роль у зниженні трансакційних витрат, підвищенні ліквідності ринку та покращенні його ефективності [6].

Різнманітний характер фінансових ринків відображає багатогранність фінансової системи та її роль у підтримці економічної активності в різних секторах та географічних регіонах. Саме тому розуміння структури та функцій даних ринків є важливим для інвесторів, бізнесу та політиків, які прагнуть зорієнтуватися в складнощах фінансового світу. Учасники фінансового ринку відіграють різні, але взаємопов'язані ролі у формуванні динаміки фінансових

ринків та у впливі на результати економіки. Інвестори надають капітал, фінансові установи опосередковують рух коштів, регулятори забезпечують цілісність ринку, а посередники сприяють здійсненню ринкових операцій.

1.2 Основні поняття та ключові слова

Досліджуючи питання прогнозування розвитку фінансових ринків важливим є визначення однозначного тлумачення певних категорій, які будуть задіяні при подальшій розробці моделі.

Відповідно до Закону України «Про інвестиційну діяльність», інвестори – це суб'єкти інвестиційної діяльності, які приймають рішення про вкладення власних, позичкових або залучених майнових та інтелектуальних цінностей в об'єкти інвестування [7]. Для подальшого дослідження будемо розуміти поняття «інвестор» як фізичну або юридичну особу, яка вкладає капітал з метою отримання прибутку. Інвестори купують певні активи: акції, облигації, товари та нерухомість для участі у фінансових ринках. На їхні інвестиційні рішення впливають такі фактори, як толерантність до ризику, очікування доходу, ринкові умови та інвестиційні цілі.

Фондові ринки – це платформи, на яких відбувається купівля та продаж акцій, що перебувають у вільному обігу. Ці ринки надають інвесторам ліквідність для торгівлі акціями публічних компаній. Найбільші фондові біржі сприяють торгівлі та відіграють важливу роль у визначенні цін та формуванні капіталу, такі як Нью-Йоркська фондова біржа (NYSE) [8], NASDAQ [9] та Лондонська фондова біржа (LSE) [10].

Компанія – це правовий суб'єкт, створений з метою ведення господарської діяльності, виробництва товарів або надання послуг. Компанії можуть бути публічними або приватними. Власність може бути розподілена між акціонерами або належати одній особі чи групі осіб. Для того, щоб оцінити інвестиційний

потенціал компанії, важливо розуміти її бізнес-модель, фінансові показники, управлінську команду, конкурентне середовище та перспективи зростання.

Акції – це частки власності в компанії, також відомі як share (частка) або equity (акція). Коли інвестори купують акції, вони набувають пропорційну частку власності в компанії-емітенті і мають право на потенційні дивіденди та приріст капіталу. Акції торгуються на фондових біржах і можуть коливатися в ціні залежно від таких факторів, як прибутки компанії, ринкові настрої, економічні умови та галузеві тенденції.

Ринкові індикатори – це показники або індекси, які дають уявлення про загальну ефективність і настрої на фінансових ринках. До загальних ринкових індикаторів належать фондові індекси, такі як S&P 500 та Dow Jones Industrial Average [11], індекси волатильності, такі як VIX, дохідність облігацій, ціни на сировинні товари та економічні показники, такі як зростання ВВП та інфляція. Зазначені показники є своєрідними індикаторами стану ринку. За ними уважно стежать інвестори, аналітики та політики.

Біржові інвестиційні фонди (exchange-traded fund ETF) – це інвестиційні фонди, якими торгують на фондовій біржі та володіють такими активами, як акції, облігації, сировинні товари або валюти [12]. Управління ETF здійснюється пасивно, щоб відстежувати результати діяльності певного індексу або класу активів. Однак деякі ETF можуть бути активно керованими.

Управління ризиками – це процес виявлення, оцінки та зменшення ризиків, пов'язаних з інвестиційною діяльністю [13]. З метою управління ризиками та захисту свого інвестиційного капіталу інвестори використовують різні методи управління ризиками, такі як диверсифікація, страхування, розподіл активів та оптимізація портфеля. Ефективне управління ризиками вимагає розуміння факторів ризику, таких як ринковий ризик, кредитний ризик, ризик ліквідності та операційний ризик.

1.3 Характеристика товарів на фінансовому ринку

Характеристики товарів на фінансовому ринку охоплюють широкий спектр факторів, які впливають на їхню поведінку, ціноутворення та можливість торгівлі. Кожна характеристика відіграє важливу роль у формуванні ринкової динаміки та прогнозуванні результатів – від ліквідності та волатильності до профілів ризику та прибутковості. Товари на фінансовому ринку включають в себе різні фінансові інструменти та активи, якими торгують інституційні інвестори. Фінансовий ринок пропонує широкий спектр інструментів, які інвестори можуть купувати та продавати для досягнення своїх інвестиційних цілей та управління ризиками. Ці фінансові інструменти відрізняються за складністю, профілем ризику та базовими активами. Для того, щоб прогнозувати показники фінансового ринку за допомогою систем штучного інтелекту, важливо розуміти різні типи інструментів, якими торгують на фінансовому ринку.

Акції, також відомі як *stocks*, представляють собою частки власності в компаніях, акції яких розміщені на біржі [14]. Коли інвестори купують акції, вони стають власниками компанії-емітента. Вони мають право на потенційні дивіденди та приріст капіталу. Акції торгуються на таких біржах, як Нью-Йоркська фондова біржа (NYSE) та NASDAQ. Ціни на них коливаються залежно від результатів діяльності компанії, ринкових настроїв та економічних факторів.

Облігації являють собою боргові інструменти, які випускаються урядами, місцевими органами влади або компаніями з метою залучення капіталу [14]. Коли інвестор купує облігацію, він позичає гроші компанії-емітенту в обмін на періодичну виплату відсотків (купона) та погашення основної суми боргу в кінці терміну погашення. Облігації відрізняються за кредитною якістю, терміном погашення та типом емітента і можуть мати фіксовану або плаваючу відсоткову ставку. Найпоширенішими видами облігацій є державні, муніципальні, корпоративні та конвертовані облігації.

Ф'ючерсні контракти – це стандартизовані угоди на купівлю або продаж певного активу (наприклад, товарів, валют, фондових індексів) за заздалегідь визначеною ціною (ф'ючерсною ціною) на майбутню дату (дату поставки) [14]. Ф'ючерсні контракти торгуються на організованих біржах, таких як Чиказька товарна біржа (CME). Вони пропонують інвесторам можливість зменшити ризик, спекулювати на зміні цін та отримати доступ до різних класів активів. Ф'ючерсні контракти – це інструменти з кредитним портфелем, які дозволяють інвесторам контролювати великі обсяги активів з відносно невеликою початковою націнкою.

Опціонні контракти дають інвестору право, але не зобов'язання, купити (опціон на купівлю) або продати (опціон на продаж) певний актив за заздалегідь визначеною ціною (ціна виконання) протягом певного періоду часу (дата закінчення) [14]. Опціони пропонують інвесторам гнучкість в управлінні ризиками, отримання доходу і спекуляції на зміні цін, і торгуються на організованих біржах, таких як Чиказька біржа опціонів (CBOE). Ціна опціонного контракту базується на таких факторах, як ціна базового цінного паперу, волатильність, час до закінчення терміну дії та відсоткові ставки.

Біржові фонди (ETF) – це інвестиційні фонди, які володіють диверсифікованим портфелем активів, таких як акції, облігації, сировинні товари або валюти. Вони торгуються на біржах, як окремі акції. ETF пропонують інвесторам доступ до різних класів активів, секторів та інвестиційних стратегій. Вони мають низьку вартість, прозорість та ліквідність. ETF можна пасивно управляти, щоб відстежувати динаміку певного індексу. Вони також можуть активно управлятися для досягнення конкретних інвестиційних цілей.

Для ефективного прогнозування показників фінансового ринку за допомогою інтелектуальних систем машинного навчання необхідно чітко розуміти характеристики цих товарів.

Термін «ліквідність» означає простоту, з якою певний фінансовий інструмент може бути куплений або проданий без значного впливу на його ціну [15]. Високоліквідні активи, такі як основні міжнародні валюти, державні облігації та

акції з великою капіталізацією, можуть легко продаватися і купуватися з невеликими різницями між цінами попиту і пропозиції та великими обсягами торгів. І навпаки, неліквідні активи, такі як акції з малою капіталізацією, корпоративні облігації та певні деривативи, можуть мати обмежену ринкову пропозицію і вищі транзакційні витрати. Ліквідність активу впливає на точність моделей прогнозування, оскільки активи з вищою ліквідністю, як правило, демонструють більш стабільні цінові коливання і кращу доступність даних.

Волатильність – це показник ступеня цінових коливань або мінливості вартості фінансових активів з плином часу [16]. Активи з високою волатильністю зазнають більших цінових коливань, тоді як активи з низькою волатильністю демонструють більш стабільну динаміку цін. На волатильність може впливати низка факторів. До них відносяться ринкові настрої, економічні події, геополітичні ризики та поведінка інвесторів. Для того, щоб точно передбачити майбутній рух цін і оцінити рівень ризику на фінансовому ринку, моделі прогнозування повинні враховувати динаміку волатильності.

Ризик і дохідність – це фундаментальні фінансові поняття, які описують компроміс між потенційними прибутками і збитками, пов'язаними з інвестиціями. Різні фінансові інструменти, залежно від таких факторів, як категорія активів, їхня кредитна репутація, строк погашення та ринкова ситуація, пропонують різні рівні ризику та дохідності. Для того, щоб оцінити інвестиційні можливості та оптимізувати ефективність портфеля, моделі прогнозування повинні враховувати дохідність, скориговану на ризик, оскільки інвестори вимагають більшої дохідності за прийняття вищого рівня ризику.

Кореляція – це показник ступеня статистичного зв'язку між доходами двох або більше фінансових активів. Позитивна кореляція вказує на те, що активи рухаються в одному напрямку, тоді як негативна кореляція свідчить про протилежні рухи. Диверсифікація має на меті зменшити ризик портфеля та підвищити дохідність, скориговану на ризик, шляхом інвестування в комбінацію активів з низькою або від'ємною кореляцією. Для побудови диверсифікованих

портфелів і зменшення ризику концентрації, моделі прогнозування повинні враховувати кореляційні моделі між активами.

Ефективність ринку – це ступінь, до якого ціни на фінансові активи відображають всю наявну інформацію вчасно і неупереджено. Ефективні ринки швидко враховують нову інформацію. Це ускладнює для інвесторів завдання постійно випереджати ринок. Щоб залишатися ефективними у визначенні прибуткових торгових можливостей, моделі прогнозування повинні адаптуватися до мінливих ринкових умов та інформаційних потоків.

У формуванні ринкової динаміки та впливі на результати прогнозування вирішальну роль відіграють характеристики активів фінансового ринку. Фінансовий ринок пропонує широкий спектр інструментів, які інвестори можуть купувати та продавати для досягнення своїх інвестиційних цілей, управління ризиками та диверсифікації своїх портфелів. Основними видами фінансових інструментів, що торгуються на ринку, є акції, облігації, ф'ючерсні контракти, опціонні контракти та біржові фонди. Для того, щоб розробляти ефективні моделі прогнозування та приймати обґрунтовані інвестиційні рішення в динамічному середовищі фінансових ринків, важливо розуміти характеристики та функції цих інструментів.

1.4 Актуальність предметної області

Фінансові ринки являють собою динамічне і складне середовище, де інвестори ухвалюють рішення на основі аналізу безлічі чинників, включно з економічними показниками, політичними подіями, технічними аспектами торгівлі та багато іншого. В умовах швидко мінливої світової економіки необхідно мати інструменти, здатні передбачати можливі напрямки руху фінансових ринків з високою точністю і оперативністю.

Для інвесторів, трейдерів і фінансових установ, які приймають обґрунтовані рішення щодо розподілу активів, управління портфелем і зниження ризиків, точне прогнозування показників фінансового ринку має важливе значення. Моделі прогнозування на основі штучного інтелекту дають змогу учасникам ринку виявляти тенденції, передбачати рух ринку та використовувати інвестиційні можливості, застосовуючи передові обчислювальні методи для аналізу величезних обсягів даних і виявлення значущих факторів. Спрямовуючи капітал у ті види економічної діяльності та галузі, які є найбільш перспективними, точне прогнозування підтримує інновації, продуктивність та економічне зростання.

Прогнозування відіграє вирішальну роль в оцінці та зменшенні ризиків, а управління ризиками є важливим аспектом операцій на фінансових ринках. Системи штучного інтелекту здатні аналізувати історичні дані та виявляти закономірності, кореляції та аномалії, які можуть свідчити про потенційні ризики на фінансових ринках. Прогнозуючи такі показники, як волатильність, коливання ставок, відсоткові розкиди та настрої, AI-моделі дають змогу ризик-менеджерам проактивно управляти ризиками та захищати портфелі від несприятливих ринкових явищ.

Ефективність фінансових ринків полегшує розподіл капіталу для його найпродуктивнішого використання, сприяє ціноутворенню та підвищує ліквідність ринку. Моделі прогнозування на основі штучного інтелекту сприяють підвищенню ефективності ринку, надаючи учасникам своєчасну і точну інформацію, зменшуючи інформаційну упередженість і підвищуючи прозорість ринку. Підвищення ефективності фінансових ринків приносить користь інвесторам, забезпечуючи справедливість і стабільність ринкових процесів, а також знижуючи транзакційні витрати.

Алгоритмічна торгівля, також відома як автоматизована або кількісна торгівля, – це метод виконання торгових стратегій на основі заздалегідь визначених правил і критеріїв за допомогою комп'ютерних алгоритмів. Моделі прогнозування на основі штучного інтелекту відіграють вирішальну роль в

алгоритмічній торгівлі, формуючи повідомлення, визначаючи торгові можливості та оптимізуючи торгові стратегії в режимі реального часу. Алгоритмічна торгівля підвищує ліквідність ринку, покращує цінову ефективність і зменшує затримку виконання, що приносить користь як інвесторам, так і цілісності ринку.

Індикатори фінансових ринків, такі як процентні ставки, темпи інфляції та обмінні курси, відіграють вирішальну роль у прийнятті рішень центральними банками та урядами щодо монетарної політики. Точне прогнозування цих показників дає змогу політикам впроваджувати своєчасну та ефективну монетарну та фіскальну політику з метою досягнення макроекономічної стабільності, зокрема цінової стабільності, повної зайнятості та стійкого економічного зростання. Узгоджуючи політичні дії з прогнозованою ринковою кон'юнктурою, політики можуть зменшити ризик економічних дисбалансів та посилити економічну стабільність.

Індикатори фінансового ринку впливають на рівень ділової та споживчої довіри, що, в свою чергу, впливає на рівень витрат, інвестицій та економічної активності. Позитивні прогнози економічного зростання, зростання цін на акції та низькі відсоткові ставки можуть покращити ділові настрої та заохотити інвестиції, стимулюючи споживчі витрати. І навпаки, негативні прогнози економічного спаду, волатильності ринку або інфляційного тиску можуть призвести до зниження рівня довіри, що, в свою чергу, може спричинити обережну поведінку серед бізнесу та споживачів. Точне прогнозування допомагає зменшити невизначеність і створити сприятливе середовище для розширення бізнесу, створення робочих місць та економічного процвітання.

Важливість прогнозування показників фінансового ринку за допомогою систем штучного інтелекту (ШІ) очевидна в сучасному динамічному фінансовому ландшафті. Моделі прогнозування на основі штучного інтелекту пропонують інвесторам, трейдерам, фінансовим установам та економіці в цілому численні переваги, зокрема покращення процесу прийняття рішень та управління ризиками, підвищення ефективності ринку, а також рушійну силу для

інновацій. Надаючи особам, відповідальним за прийняття рішень, включаючи політиків, інвесторів, бізнес та споживачів, цінну інформацію про динаміку ринку та майбутні тенденції, точне прогнозування сприяє прийняттю обґрунтованих рішень та ефективному розподілу капіталу, а також сприяє процесу управління ризиками. Очікується, що з розвитком технологій важливість прогнозування для економіки зростатиме, і це матиме значний вплив на траєкторію економічного розвитку та процвітання в найближчі роки.

1.5 Постановка задачі дослідження

Дане дослідження спрямоване на розробку та перевірку ефективності систем штучного інтелекту для прогнозування фінансових ринків з метою досягнення високого рівня точності та надійності отриманих прогнозів. Однією з цілей дослідження є вирішення проблем і невизначеностей, пов'язаних з даними фінансового ринку, таких як шум, пропущені значення і нестационарність. Дослідження вивчатиме методи попередньої обробки даних, методи інженерії особливостей та підходи до регуляризації моделей для підвищення якості даних, зменшення упередженості даних та покращення надійності моделей.

Включення додаткових джерел даних, таких як новини, настрої в соціальних мережах, економічні показники та корпоративні оголошення, поряд з історичною вартістю акцій, може значно підвищити точність прогнозування показників фінансового ринку. Використовуючи різноманітні джерела даних, системи штучного інтелекту можуть вловлювати нюанси ринкової динаміки та визначати відповідні фактори, що впливають на рух цін на акції.

В рамках дослідження буде проведено комплексну оцінку розроблених моделей прогнозування для визначення їхньої ефективності, здатності до узагальнення та надійності за різних ринкових умов та часових періодів. Для перевірки точності та надійності моделей будуть застосовані ретельні процедури

тестування, включаючи перехресну перевірку та тестування на окремому сеті даних.

Визначено план дослідження.

1. Огляд літератури. Провести всебічний огляд існуючої літератури з питань прогнозування фінансових ринків, методів штучного інтелекту та інтеграції новин і зовнішніх джерел даних у прогнозному моделюванні. Визначити відповідні теорії, методології та емпіричні результати для формування дослідницького підходу.
2. Збір та підготовка даних. Зібрати історичні фінансові дані, включаючи ціни на акції, обсяги торгів та інші ринкові показники, з авторитетних джерел, таких як фінансові бази даних і біржі. Для доповнення історичних даних слід збирати новини, пости в соціальних мережах, економічні звіти та корпоративні оголошення. Очистити і попередньо обробити дані, щоб видалити шум, обробити відсутні значення і забезпечити узгодженість.
3. Інженерія ознак. Виконати інженерію ознак для вилучення релевантних ознак із зібраних джерел даних. Розробити методи представлення текстових новинних даних у вигляді числових ознак, таких як оцінка настроїв. Дослідити методи відбору ознак для визначення найбільш інформативних ознак для прогнозування тенденцій цін на акції та їхньої майбутньої вартості.
4. Розробка моделей. Розробити моделі прогнозування на основі штучного інтелекту, які об'єднують історичні ціни на акції з додатковими джерелами даних, зокрема аналізом новинних настроїв та економічними показниками. Для створення надійних моделей прогнозування дослідити різні алгоритми машинного навчання, такі як регресійні моделі, аналіз часових рядів та архітектури глибинного навчання. Налаштувати параметри моделі та оптимізувати гіперпараметри для максимізації точності прогнозування.

5. Метрики оцінювання. Визначити відповідні метрики оцінювання для оцінки ефективності розроблених моделей прогнозування. Враховувати такі показники, як середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE), Precision, Recall та оцінка F1, щоб оцінити точність прогнозування, надійність та здатність до узагальнення. Провести ретельні валідаційні експерименти з використанням методів перехресної валідації та позавибіркового тестування для забезпечення надійності та узгодженості результатів.
6. Інтерпретація результатів. Розробити методи інтерпретації результатів прогнозування та вилучення практичних висновків з прогнозованих значень. Аналіз впливу новинних настроїв, економічних показників та інших зовнішніх факторів на рух цін на акції. Візуалізувати результати та тенденції прогнозування, щоб полегшити розуміння та прийняття рішень зацікавленими сторонами.
7. Порівняльний аналіз. Порівняти ефективність моделей прогнозування на основі ШІ, що враховують новини та зовнішні дані, з традиційними моделями, заснованими виключно на історичних цінах на акції. Провести порівняльний аналіз, щоб оцінити додану вартість інтеграції аналізу новинних настроїв і зовнішніх джерел даних для підвищення точності прогнозування та інтерпретації результатів.

Висновки до розділу 1

В результаті опрацювання літературних джерел визначено, що сприяючи розподілу капіталу, управлінню ризиками та сприяючи економічному зростанню і розвитку, фінансовий ринок слугує важливою основою глобальної економіки. Фінансовий ринок складається з різних учасників, включаючи інвесторів, фінансові установи, регуляторів та посередників, кожен з яких відіграє окрему, але взаємопов'язану роль у формуванні того, як функціонує ринок і як він виконує свої функції.

Встановлено, що до ключових характеристик, які визначають фінансові інструменти та впливають на ефективність ринку, належать ліквідність, волатильність, ризик і дохідність, кореляція та ефективність ринку. Для того, щоб точно та ефективно прогнозувати показники фінансового ринку, важливо розуміти сутнісні характеристики даного ринку.

Проаналізувавши різні алгоритми машинного навчання та метрики оцінювання, визначено переваги та недоліки застосування кожного з них. Ідентифіковано методи для прогнозування тенденцій цін на акції та їхньої майбутньої вартості. Аналізуючи вплив новин, економічних показників та зовнішніх факторів на динаміку цін на акції, подальше дослідження має на меті покращити розуміння та обґрунтування інвестиційних рішень.

Окреслений план дослідження включає комплексний підхід до збору даних, розробки функцій, моделей, метрик оцінки, інтерпретації результатів, перевірки та поширення. Вирішення цих завдань дозволить поглибити знання та практичне застосування прогнозування фінансових ринків, що в кінцевому підсумку принесе користь інвесторам, фінансовим установам, політикам та економіці в цілому.

РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ ФІНАНСОВИХ РИНКІВ

2.1 Побудова інтелектуальних систем для прогнозування фінансових ринків

Прогнозування часових рядів є важливим аспектом фінансового аналізу та прийняття рішень. Воно передбачає використання історичних даних для прогнозування майбутніх значень часових рядів, що дозволяє інвесторам приймати обґрунтовані інвестиційні рішення, бізнесу – оптимізувати свою діяльність, а політикам – формулювати ефективні стратегії розвитку суспільства.

Перш ніж застосовувати методи прогнозування, важливо попередньо обробити дані часових рядів, щоб видалити будь-який сторонній шум або нерівномірності, які можуть перешкоджати процесу прогнозування. Цей процес, відомий як фільтрація часових рядів, включає деякі методи.

1. Метод ковзної середньої (МА) – це простий і широко використовуваний метод фільтрації даних часових рядів. Він передбачає усереднення сусідніх точок даних в межах заданого розміру вікна, ефективно згладжуючи шум і нерівномірності. МА-фільтр особливо ефективний для видалення високочастотного шуму, зберігаючи при цьому основний тренд і циклічні закономірності. МА-фільтр характеризується розміром вікна, який визначає кількість точок даних, що беруть участь у процесі усереднення.
2. Цифрові фільтри – це більш складний підхід до фільтрації даних часових рядів. Вони використовують математичні рівняння для маніпулювання сигналом, ефективно видаляючи задані частотні компоненти. Цифрові фільтри можна розділити на три основні типи:
 - фільтри низьких частот – пропускають низькочастотні компоненти сигналу, послаблюючи або усуваючи високочастотні компоненти. Вони зазвичай

використовуються для видалення високочастотного шуму, не впливаючи на основний тренд і циклічні патерни;

- високочастотні фільтри – пропускають високочастотні компоненти сигналу, послаблюючи або усуваючи низькочастотні компоненти. Вони корисні для ізоляції раптових змін або стрибків у даних;
 - band-pass фільтри – пропускають певний діапазон частот, послаблюючи або усуваючи інші частоти. Вони часто використовуються для ізоляції певних частотних компонентів, що становлять інтерес.
3. Експоненціальний фільтр – це тип цифрового фільтра, який надає більшу вагу останнім точкам даних, ефективно фіксуючи останні тенденції та динаміку. Він заснований на експоненціальній функції спадання, яка плавно переходить від старіших до новіших точок даних. Експоненціальний фільтр часто використовується для фільтрації в реальному часі, оскільки він може адаптуватися до мінливих тенденцій без значного запізнення.
 4. Поліноміальний фільтр – це тип цифрового фільтра, який підбирає поліноміальну функцію до даних, ефективно видаляючи шум і наближаючи основний тренд. Ступінь полінома визначає складність підгонки, причому поліноми вищого ступеня відображають більш складні закономірності. Поліноміальні фільтри особливо корисні для роботи з нелінійними трендами або складними структурами даних.
 5. Медіанний фільтр – це непараметричний фільтр, який замінює кожну точку даних медіаною її сусідніх точок. Даний фільтр особливо ефективний для видалення викидів та імпульсного шуму, зберігаючи при цьому загальний розподіл даних. Медіанний фільтр часто використовується для обробки зображень і виявлення аномалій [17].

Вибір релевантних ознак має вирішальне значення для ефективного прогнозування часових рядів. Комплексний набір ознак може охопити основну

динаміку даних і підвищити точність прогнозування. Однак, включення занадто великої кількості ознак може призвести до надмірного припасування, коли модель стає занадто складною і не здатною узагальнювати нові дані.

Для визначення найбільш релевантних ознак можна використовувати методи відбору ознак, такі як кореляційний аналіз та вимірювання важливості ознак. Кореляційний аналіз вимірює силу зв'язку між кожною ознакою та цільовою змінною. Показники важливості ознак, такі як Information Gain або SelectKBest, кількісно оцінюють внесок кожної ознаки в передбачуваність моделі. Кореляційний аналіз – це статистичний метод, який вимірює силу і напрямок зв'язку між двома або більше змінними. У прогнозуванні часових рядів кореляційний аналіз може бути використаний для вирішення наступних завдань:

- визначення впливових факторів, тобто визначення змінних, які сильно корелюють з цільовою змінною, що вказує на їх потенційну важливість для прогнозування;
- зменшення простору ознак – усунення сильно корельованих ознак, що зменшує складність моделі та мінімізує надмірне припасування;
- вивчення взаємозв'язків між змінними, тобто розуміння взаємозв'язків між різними змінними в часовому ряді, що дасть змогу зрозуміти основну динаміку [18].

Вибір кількості атрибутів є критично важливим аспектом прогнозування часових рядів. Занадто багато атрибутів може призвести до надмірного припасування, тоді як занадто мало атрибутів може не охопити всю складність даних і призвести до неточних прогнозів.

Для визначення оптимальної кількості атрибутів можна використовувати декілька методів:

- статистичні методи, такі як інформаційний критерій акаїке (aic) та байєсівський інформаційний критерій (bic), оцінюють складність моделі та надають перевагу простішим моделям з меншою кількістю атрибутів;

- перехресна перевірка, що передбачає поділ даних на навчальні та тестові набори та оцінку продуктивності моделі на тестовому наборі для різної кількості атрибутів;
- експертне судження – експертиза предметної області може бути використана для вибору атрибутів на основі знань предметної області та розуміння основних взаємозв'язків між змінними.

Відсутність даних може суттєво вплинути на точність прогнозування часових рядів. Для вирішення цієї проблеми можна використати декілька методів:

- обчислення середнього значення або медіани: замінити відсутні значення середнім значенням або медіаною відповідної змінної;
- інтерполяція: оцінка відсутніх значень на основі значень сусідніх точок даних;
- обчислення на основі моделей, тобто слід використовувати статистичні моделі або моделі машинного навчання для прогнозування пропущених значень на основі загального розподілу даних.

Стаціонарність означає властивість часового ряду, коли його статистичні властивості залишаються незмінними з часом. Нестационарні часові ряди демонструють тенденції, цикли або інші закономірності, які змінюються з часом, що ускладнює їх прогнозування за допомогою стандартних методів аналізу часових рядів.

Для перетворення нестационарних часових рядів у стаціонарні можна використати декілька методів, а саме:

- диференціювання – віднімання послідовних значень часового ряду;
- перетворення Бокса-Кокса – застосування логарифмічного або степеневого перетворення для стабілізації дисперсії;
- сезонне коригування – видалення сезонних компонентів за допомогою таких методів, як детрендування та десеоналізація [19].

Нормалізація передбачає масштабування даних до загального діапазону, зазвичай від 0 до 1, щоб гарантувати, що всі змінні мають порівнянну величину та розподіл. Дана стандартизація особливо важлива при використанні алгоритмів машинного навчання, оскільки вона допомагає запобігти домінуванню змінних з більшими значеннями.

Методи нормалізації включають в себе наступні:

- мінімально-максимальна нормалізація, де масштабування значень до діапазону $[0, 1]$ на основі мінімального та максимального значень даних;
- нормалізація за Z -рахунком, тобто віднімання середнього значення та ділення на стандартне відхилення кожної змінної.

Нормалізація полегшує порівняння різних часових рядів і спрощує навчання моделей машинного навчання [20].

Вибір відповідної моделі прогнозування залежить від кількох факторів, серед яких можна виділити наступні:

- характеристики даних: характер даних часового ряду, наприклад, його стаціонарність, тренд і сезонність;
- горизонт прогнозування: довжина горизонту прогнозування, оскільки різні моделі можуть працювати краще для короткострокових або довгострокових прогнозів;
- бажана точність: рівень точності, необхідний для прогнозів, оскільки більш складні моделі можуть пропонувати вищу точність, але також можуть бути дорогими або обчислювально складними;
- експертиза предметної області: знання предметної області про дані та завдання прогнозування можуть допомогти у виборі найбільш підходящої моделі.

2.2 Дослідження існуючих інтелектуальних систем

Для прогнозування часових рядів можна використовувати широкий спектр моделей, кожна з яких має свої сильні та слабкі сторони. Вибір моделі залежить від характеристик даних, горизонту прогнозування та бажаного рівня точності.

Моделі ковзної середньої є простими і широко використовуваними методами для прогнозування даних часових рядів. Вони базуються на припущенні, що на майбутні значення впливають нещодавні минулі значення. Моделі ковзної середньої ефективні для виявлення короткострокових тенденцій та закономірностей, але вони можуть бути не дуже ефективними для більш тривалих горизонтів прогнозування.

Авторегресійні моделі – це ще один клас широко використовуваних методів прогнозування часових рядів. Вони припускають, що майбутні значення визначаються лінійною комбінацією минулих значень. Авторегресійні моделі ефективно відображають лінійні тенденції та закономірності, але вони можуть бути непридатними для нелінійних даних.

Моделі ARIMA поєднують в собі моделі AR та MA з різницею для обробки нестационарних даних. Вони вважаються універсальними моделями прогнозування, які можуть охопити широкий спектр тенденцій та закономірностей.

Рекурентні нейронні мережі (RNN) – це клас нейронних мереж, спеціально розроблених для обробки послідовних даних, таких як часові ряди. RNN чудово вловлюють довгострокові залежності та тенденції, що робить їх добре придатними для прогнозування на довші терміни.

LSTM-мережі – це тип RNN, який є особливо ефективним для обробки довгострокових залежностей у часових рядах даних. Вони включають комірки пам'яті, які дозволяють їм запам'ятовувати минулу інформацію протягом тривалих періодів, що дозволяє їм вловлювати складні закономірності та робити точні довгострокові прогнози.

На додаток до вищезгаданих моделей, для прогнозування часових рядів використовується кілька інших методів, зокрема:

- машини опорних векторів (SVM) – алгоритм керованого навчання, який можна використовувати для прогнозування часових рядів, використовуючи функції ядра для відображення нелінійних зв'язків;
- дерева рішень – деревоподібні моделі, які можна використовувати для прогнозування, розбиваючи дані на підмножини на основі правил прийняття рішень;
- методи ансамблю – це методи, які поєднують декілька моделей прогнозування для підвищення загальної точності прогнозування [21].

2.2.1 Моделі ковзного середнього

Моделі ковзного середнього, які часто скорочено називають МА-моделями, – це клас моделей часових рядів, які зазвичай використовуються для прогнозування майбутніх значень на основі минулих спостережень. Ці моделі відносно прості, але ефективні для виявлення короткострокових тенденцій і згладжування випадкових коливань у даних часових рядів.

До ключових особливостей моделей ковзного середнього можна віднести наступні:

- вікно усереднення: моделі ковзного середнього працюють шляхом усереднення значень певної кількості минулих спостережень, відомих як вікно усереднення або порядок запізнення. розмір вікна усереднення визначає ступінь згладжування, що застосовується до даних часового ряду;
- постійне середнє: моделі ковзного середнього припускають постійне середнє в часі, що означає, що середнє значення часового ряду залишається стабільним протягом усього періоду спостереження;

- фільтрація шуму: усереднюючи минулі спостереження, моделі ковзного середнього відфільтровують випадковий шум і короткострокові коливання в даних часового ряду.

Існують різні типи моделей ковзних середніх:

- просте ковзне середнє (SMA) – найпростіша форма моделі ковзного середнього, де кожне прогнозоване значення розраховується як середнє значення останніх спостережень в межах вікна усереднення (усі спостереження у вікні мають однакову вагу);
- зважене ковзне середнє (WMA) – у таких моделях кожному спостереженню у вікні усереднення присвоюється різна вага (як правило, останнім спостереженням надається більша вага, що відображає їхню більшу релевантність для прогнозування майбутніх значень);
- експоненціальне ковзне середнє (EMA) – дані моделі присвоюють минулим спостереженням ваги, що експоненціально зменшуються, а останнім спостереженням – більшу вагу (моделі EMA швидше реагують на останні зміни в даних часового ряду порівняно з моделями SMA або WMA [22]).

Переваги моделей ковзного середнього для прогнозування часових рядів полягають у наступному:

- згладжування шуму: моделі ковзного середнього ефективно згладжують випадкові коливання або шум, присутній у даних часових рядів;
- простота та інтуїтивність у застосуванні;
- адаптивність до змін даних: моделі ковзного середнього адаптуються до змін у вихідних даних з плином часу;
- обчислювальна ефективність: такі моделі є обчислювально ефективними, особливо у порівнянні з більш складними моделями часових рядів, такими як моделі авторегресійного інтегрованого ковзного середнього (ARIMA) або алгоритми машинного навчання.

До слабких сторін моделей ковзного середнього для прогнозування часових рядів слідвіднести наступні:

- обмежена прогностична здатність, оскільки моделі ковзного середнього більшою мірою підходять для виявлення короткострокових тенденцій та згладжування шуму в даних часових рядів, проте вони можуть бути не дуже ефективними для прогнозування довгострокових тенденцій, сезонності або складних закономірностей, присутніх у даних;
- брак гнучкості: моделі ковзних середніх мають обмежену гнучкість порівняно з більш складними моделями часових рядів, вибір розміру вікна усереднення визначає ступінь згладжування, що застосовується до даних та самого прогнозу;
- неможливість врахувати нелінійні зв'язки: такі моделі припускають лінійний зв'язок між минулими спостереженнями та майбутніми значеннями, тому вони можуть не врахувати нелінійні закономірності або залежності, присутні в даних, особливо коли основні взаємозв'язки є складними або динамічними;
- чутливість до викидів: моделі ковзного середнього є чутливими до викидів або екстремальних значень у даних часового ряду – викиди можуть непропорційно впливати на обчислені середні значення, що призводить до упереджених прогнозів або неточного відображення основних тенденцій;
- обмежений горизонт прогнозування: такі моделі, як правило, підходять для короткострокових горизонтів прогнозування, тому прогнозування за межами вікна усереднення може призвести до менш надійних прогнозів, оскільки модель може неадекватно відображати довгострокові тенденції або закономірності в даних [23].

Загалом, хоча моделі ковзного середнього є відносно простими у застосуванні, інтерпретованості та адаптивні для задач прогнозування часових

рядів, вони мають обмеження у своїй прогностичній силі, гнучкості та здатності охоплювати складні взаємозв'язки в даних.

2.2.2 Авторегресійні моделі

Авторегресійні (AR) моделі – це клас моделей часових рядів, які зазвичай використовуються для прогнозування майбутніх значень на основі минулих спостережень. Ці моделі припускають, що значення змінної часового ряду в певний момент часу лінійно залежить від її минулих значень, а також від похибки, що представляє випадкові коливання.

Ключові особливості авторегресійних моделей включають в себе наступні:

- залежність від часу: авторегресійні моделі враховують часову залежність або автокореляцію, яка присутня в даних часових рядів. Вони припускають, що значення змінної часового ряду в певний момент часу лінійно залежить від її минулих значень, причому сила залежності визначається параметрами моделі;
- порядок (p): порядок авторегресійної моделі, позначений як «р», представляє кількість спостережень із запізненням, включених до моделі;
- стаціонарність: авторегресійні моделі зазвичай застосовуються до стаціонарних часових рядів, де середнє, дисперсія та структура автоковаріації залишаються постійними з часом;
- оцінка параметрів: параметри авторегресійної моделі, включаючи коефіцієнти, пов'язані з минулими спостереженнями, та дисперсію помилки, оцінюються за допомогою таких методів, як оцінка максимальної правдоподібності (МНК) або оцінка найменших квадратів [24].

Математично авторегресійна модель порядку p (AR(p)) може бути представлена у вигляді:

$$Y_t = c + \phi_1 Y_{\{t-1\}} + \phi_2 Y_{\{t-2\}} + \dots + \phi_p Y_{\{t-p\}} + \varepsilon_t \quad (2.1)$$

де Y_t – значення змінної часового ряду в момент часу t ;

c – постійний член;

$\phi_1, \phi_2, \dots, \phi_p$ – коефіцієнти авторегресії, що представляють лінійну залежність від минулих значень;

$Y_{\{t-1\}}, Y_{\{t-2\}}, \dots, Y_{\{t-p\}}$ – значення із запізненням змінної часового ряду;

ε_t – член похибки, що представляє випадкові зміни або шум.

Переваги авторегресійних моделей для прогнозування часових рядів наступні:

- інтерпретованість: AR – моделі надають результати, які можна інтерпретувати, оскільки коефіцієнти представляють лінійний зв'язок між минулими спостереженнями та майбутніми значеннями;
- авторегресійні моделі відносно прості в реалізації і вимагають мінімального налаштування параметрів порівняно з більш складними моделями часових рядів;
- такі моделі є ефективними для короткострокових горизонтів прогнозування, особливо коли базові дані демонструють стаціонарну поведінку.

Слабкі сторони авторегресійних моделей для прогнозування часових рядів полягають у наступному:

- авторегресійні моделі можуть мати обмежену прогностичну здатність для довгострокових горизонтів прогнозування, особливо коли базові дані демонструють нестаціонарну поведінку або складні патерни; вони мають тенденцію до лінійної екстраполяції минулих тенденцій, що може не враховувати нелінійні або структурні зміни в даних;

- ефективність авторегресійних моделей є чутливою до вибору порядку моделі (кількості спостережень із запізненням, включених до моделі), тому вибір невідповідного порядку моделі може призвести до надмірного або недостатнього підлаштування, що може призвести до упереджених прогнозів або поганої роботи моделі;
- припущення про стаціонарність: авторегресійні моделі припускають, що дані часових рядів є стаціонарними, тобто середнє значення, дисперсія та структура автоковаріації залишаються постійними з часом;
- чутливість до викидів: такі моделі можуть бути чутливими до викидів або екстремальних значень у даних часового ряду, які можуть спотворити оцінку коефіцієнтів авторегресії та призвести до неточних прогнозів; викиди можуть непропорційно впливати на прогнози моделі, особливо якщо вони виникають на ранніх стадіях часового ряду.

Загалом, хоча авторегресійні моделі пропонують простоту, інтерпретованість та ефективність для короткострокового прогнозування, вони мають обмеження в прогнозуванні довгострокових тенденцій та чутливості до порядку моделі, припущень про стаціонарність та викидів.

2.2.3 Багатошаровий перцептрон

Багатошаровий перцептрон (MLP) – це тип штучної нейронної мережі, натхненний структурою людського мозку. Це обчислювальна модель, яка імітує спосіб обробки інформації людським мозком. Вона складається з взаємопов'язаних вузлів, які називаються штучними нейронами, організованих у шари. Інформація проходить через мережу, при цьому кожен нейрон обробляє сигнали і впливає на інші. Це нейронна мережа з прямим поширенням, тобто

інформація рухається в одному напрямку, від вхідного шару через один або кілька прихованих шарів до вихідного шару, без будь-яких циклів або петель.

Ключові компоненти та характеристики багатошарового перцептрона (рис. 2.1):

- вхідний шар, що складається з нейронів, які отримують вхідні дані або ознаки;
- приховані шари, які є проміжними шарами між вхідним і вихідним шарами. кожен прихований шар містить один або більше нейронів, які виконують обчислення над вхідними даними, використовуючи зважені зв'язки та функції активації, приховані шари дозволяють мережі вивчати складні закономірності та взаємозв'язки в даних;
- вихідний шар – виробляє остаточний результат або прогноз на основі обчислень, виконаних у прихованих шарах, кількість нейронів у вихідному шарі залежить від характеру і формату вихідних даних;
- зважені зв'язки: кожен нейрон у багатошаровому перцептроні з'єднаний з нейронами в попередньому та наступному шарі за допомогою вагових зв'язків. ці ваги відображають силу зв'язків між нейронами і коригуються в процесі навчання для мінімізації помилок прогнозування;
- функції активації, які вносять нелінійність в мережу і визначають вихід кожного нейрона на основі його зважених входів [25].

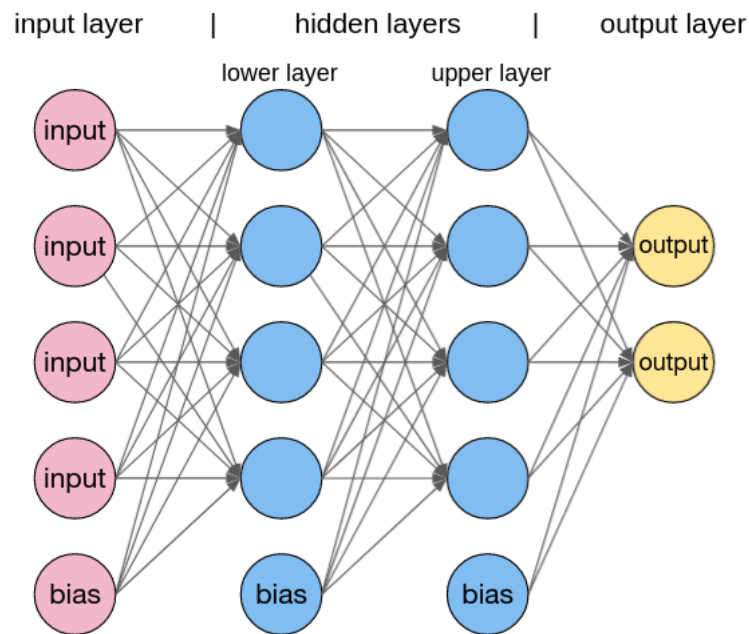


Рисунок 2.1 – Архітектура багат шарового перцептрона [26]

Серед переваг багат шарового перцептрона (MLP) для прогнозування часових рядів слід виділити наступні:

- моделювання нелінійних взаємозв'язків: MLP можуть вловлювати складні нелінійні взаємозв'язки між вхідними змінними та цільовими результатами;
- здатність вивчати ієрархічні характеристики: MLP з декількома прихованими шарами можуть вивчати ієрархічні представлення даних часових рядів, виділяючи абстрактні ознаки на різних рівнях абстракції;
- гнучкість в архітектурі моделі: такі моделі пропонують гнучкість в архітектурі моделі, дозволяючи налаштовувати кількість прихованих шарів, кількість нейронів у кожному шарі та вибір функцій активації;
- адаптивність до різних типів вхідних даних: MLP можуть обробляти широкий спектр типів вхідних даних, включаючи числові, категоріальні та послідовні дані.

Слабкі сторони багатошарового перцептрона для прогнозування часових рядів наступні:

- MLP мають декілька гіперпараметрів, таких як кількість прихованих шарів, кількість нейронів у шарі та швидкість навчання, які потрібно ретельно налаштувати;
- MLP схильні до перенавчання, особливо коли навчаються на невеликих наборах даних або коли архітектура моделі занадто складна порівняно з кількістю доступних даних;
- побудова MLP може ускладнити інтерпретацію того, як модель робить свої прогнози, особливо для складних архітектур з декількома прихованими шарами;
- навчання MLP може бути дорогим в обчислювальному плані, особливо для великих наборів даних і складних архітектур моделей.

Загалом, хоча MLP мають ряд переваг для задач прогнозування часових рядів, таких як здатність моделювати нелінійні зв'язки та вивчати ієрархічні особливості, вони також мають обмеження, включаючи чутливість до гіперпараметрів, ризик перенавчання, обмежену інтерпретацію та обчислювальну складність.

2.2.4 Рекурентна нейронна мережа

Рекурентні нейронні мережі (RNN) – це клас штучних нейронних мереж, призначених для моделювання послідовних даних шляхом виявлення часових залежностей і закономірностей. RNN широко використовуються для прогнозування часових рядів завдяки їхній здатності обробляти послідовні дані різної довжини та фіксувати довгострокові залежності. Архітектура RNN складається з декількох ключових компонентів (рис. 2.2).

1. Рекурентні зв'язки: RNN містять рекурентні зв'язки, які дозволяють інформації зберігатися на різних часових кроках. На кожному часовому кроці вихід мережі подається назад на вхід мережі, що дозволяє мережі зберігати пам'ять про минулі спостереження та включати їх у майбутні прогнози.
2. Прихований стан: RNN зберігають вектор прихованого стану, також відомий як вектор пам'яті або контексту, який представляє внутрішній стан мережі на кожному часовому кроці. Прихований стан слугує узагальненням минулих спостережень і фіксує часовий контекст, необхідний для прогнозування на поточному часовому кроці.
3. Розгортання в часі: RNN часто концептуалізуються як розгорнуті в часі, де кожен часовий крок представляє окремий екземпляр мережі. Під час навчання та формування висновків RNN розгортається на фіксовану кількість часових кроків, що дозволяє мережі обробляти послідовні дані в часі.
4. Навчання з поширенням у часі: RNN навчаються за допомогою алгоритму поширення у часі (BPTT), який є варіантом стандартного алгоритму поширення, адаптованого для послідовних даних. BPTT передбачає поширення помилок назад у часі, оновлення параметрів

мережі (ваг та зсувів) для мінімізації різниці між прогнозованими та фактичними значеннями [27].

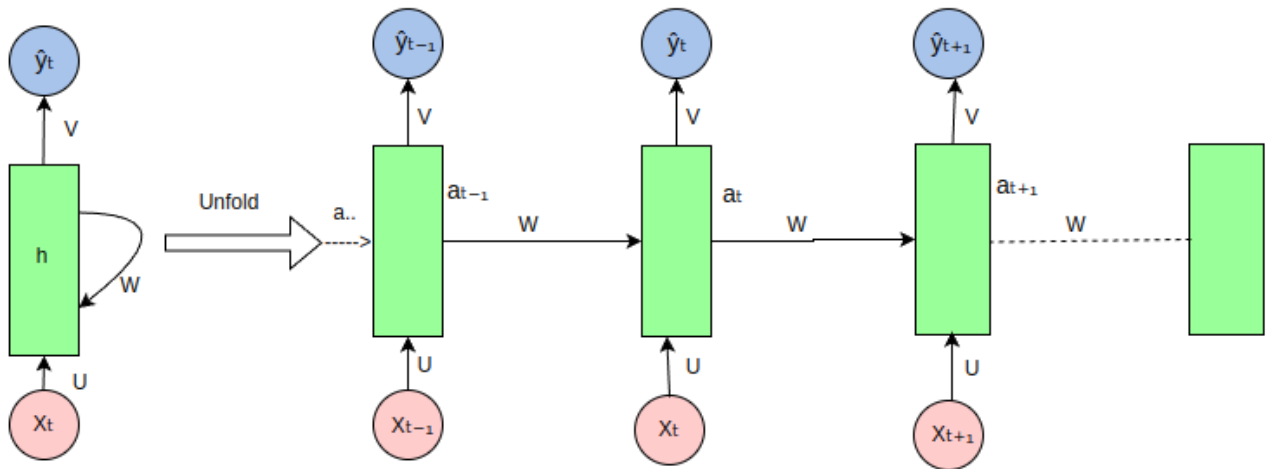


Рисунок 2.2 – Архітектура Рекурентної нейронної мережі [28]

Базова архітектура стандартного RNN.

1. Вхід: на кожному часовому кроці t RNN отримує вхідний вектор x_t , що представляє спостережувані ознаки або змінні на цьому часовому кроці.
2. Прихований стан: RNN зберігає вектор прихованого стану h_t , який відображає внутрішній стан або пам'ять мережі на кроці t . Прихований стан оновлюється на кожному кроці на основі поточного входу та попереднього прихованого стану:

$$h_t = f(W_h \cdot h_{\{t-1\}} + W_x \cdot x_t + b_h) \quad (2.2)$$

де W_h та W_x – вагові матриці;

b_h – вектор зсуву;

f – функція активації (зазвичай нелінійна функція, така як гіперболічний тангенс або сигмоїдна функція).

3. Вихід: RNN продукує вихідний вектор y_t на кожному часовому кроці, який представляє передбачення або оцінку цільової змінної на цьому кроці:

$$y_t = g(W_y \cdot h_t + b_y) \quad (2.3)$$

де W_y – вагова матриця, що зв'язує прихований стан з виходом;

b_y – вектор зсуву;

g – функція активації виходу.

Рекурентні нейронні мережі (RNN) мають кілька сильних і слабких сторін для задач прогнозування часових рядів, а також ризик *overfitting*, який є поширеною проблемою в моделях машинного навчання.

Переваги RNN для прогнозування часових рядів:

- RNN призначені для моделювання послідовних даних, що робить їх добре придатними для задач прогнозування часових рядів, де часові залежності та закономірності відіграють вирішальну роль;
- такі моделі можуть обробляти вхідні послідовності різної довжини, що дозволяє їм обробляти дані часових рядів з нерегулярними або нерівномірно розподіленими спостереженнями;
- RNN підтримують стан внутрішньої пам'яті (прихований стан), який зберігає інформацію з попередніх часових кроків; така пам'ять дозволяє RNN фіксувати довгострокові залежності в даних і включати історичний контекст у майбутні прогнози.
- дані моделі можуть вловлювати нелінійні взаємозв'язки та часові закономірності в даних часових рядів, що дозволяє їм моделювати складну динаміку та варіації.

Слабкі сторони RNN для прогнозування часових рядів полягають у наступному:

- стандартні архітектури RNN чутливі до проблем зникаючих та вибухаючих градієнтів, коли градієнти стають або надзвичайно малими, або великими під час навчання, що перешкоджає навчанню та збіжності; зазначені проблеми можуть обмежити здатність RNN ефективно фіксувати довгострокові залежності в послідовних даних;
- не зважаючи на те, що RNN здатні зберігати пам'ять протягом тривалого часу, на практиці вони можуть мати труднощі з виявленням довготривалих залежностей, особливо коли часовий проміжок між релевантними подіями великий;
- навчання RNN для задач прогнозування часових рядів може вимагати значних обчислювальних затрат і тривалого часу, особливо для великих наборів даних і складних архітектур. Налаштування гіперпараметрів, таких як швидкість навчання та архітектура мережі, може вимагати значних експериментів для досягнення оптимальної продуктивності;
- RNN, як і інші моделі машинного навчання, схильні до перенавчання, особливо коли архітектура моделі занадто складна порівняно з наявними навчальними даними.

RNN широко застосовуються в задачах прогнозування часових рядів, включаючи прогнозування цін на акції, прогнозування погоди та попиту, серед інших. Однак стандартні RNN можуть не вловлювати довгострокові залежності в послідовних даних через такі проблеми, як зникаючі градієнти та труднощі зі збереженням інформації в довгих послідовностях. Хоча RNN мають ряд переваг для задач прогнозування часових рядів, таких як послідовне моделювання та збереження пам'яті, вони також мають обмеження і можуть бути схильні до перенавчання. Для подолання цих обмежень були розроблені більш досконалі архітектури ШНМ, такі як мережі з довгою короткочасною пам'яттю (LSTM) та закритими рекурентними блоками (GRU), які пропонують покращену продуктивність у виявленні довгострокових залежностей та уникають проблеми зникаючого градієнта [29].

2.2.5 Моделі довгої короткочасної пам'яті (LSTM)

Мережі з довгою короткочасною пам'яттю (LSTM) представляють особливий тип архітектури рекурентних нейронних мереж (RNN). Вони були розроблені з метою вирішення проблеми зникаючого градієнта та фіксації довгострокових залежностей у послідовних даних. LSTM часто використовуються в контексті прогнозування часових рядів завдяки їхній здатності ефективно моделювати часові залежності, враховувати нерівномірні часові інтервали та зберігати інформацію в довгих послідовностях.

Архітектура мережі LSTM представлено на рис. 2.3.

1. Одним з ключових компонентів LSTM є «стан комірки» – вектор, який служить каналом для потоку інформації через мережу. Стан комірки здатний передавати інформацію в довгих послідовностях і вибірково зберігати або відкидати інформацію відповідно до вхідних даних і внутрішніх воріт мережі.
2. Архітектура LSTM складається з трьох типів вентилів, які регулюють потік інформації: вентиль забування, вхідний вентиль і вихідний вентиль. Кожен вентиль складається з сигмоїдної функції активації, за якою слідує поелементне множення. Це дозволяє мережі регулювати потік інформації через стан комірки:
 - вентиль забуття визначає, яку інформацію з попереднього стану комірки слід відкинути або забути, виходячи з поточного вхідного та прихованого стану;
 - вхідний вентиль визначає, яку нову інформацію з поточного входу слід додати до стану комірки;
 - вихідний вентиль визначає, яка інформація з поточного стану комірки буде виведена в прихований стан наступного кроку.

3. Прихований стан підтримується LSTM аналогічно до стандартних RNN, коли внутрішній стан мережі фіксується на кожному часовому кроці. Прихований стан оновлюється на основі поточного входу, попереднього прихованого стану та виходів вентилів.
4. Обробка довготривалої пам'яті: LSTM спеціально розроблені для вирішення проблеми зникаючого градієнта в стандартних RNN шляхом введення механізмів для збереження інформації в довгих послідовностях. Поєднання функцій стану комірки, вентилів та активації дозволяє LSTM вибірково оновлювати та поширювати інформацію протягом часу, уникаючи при цьому зникнення градієнта [30].

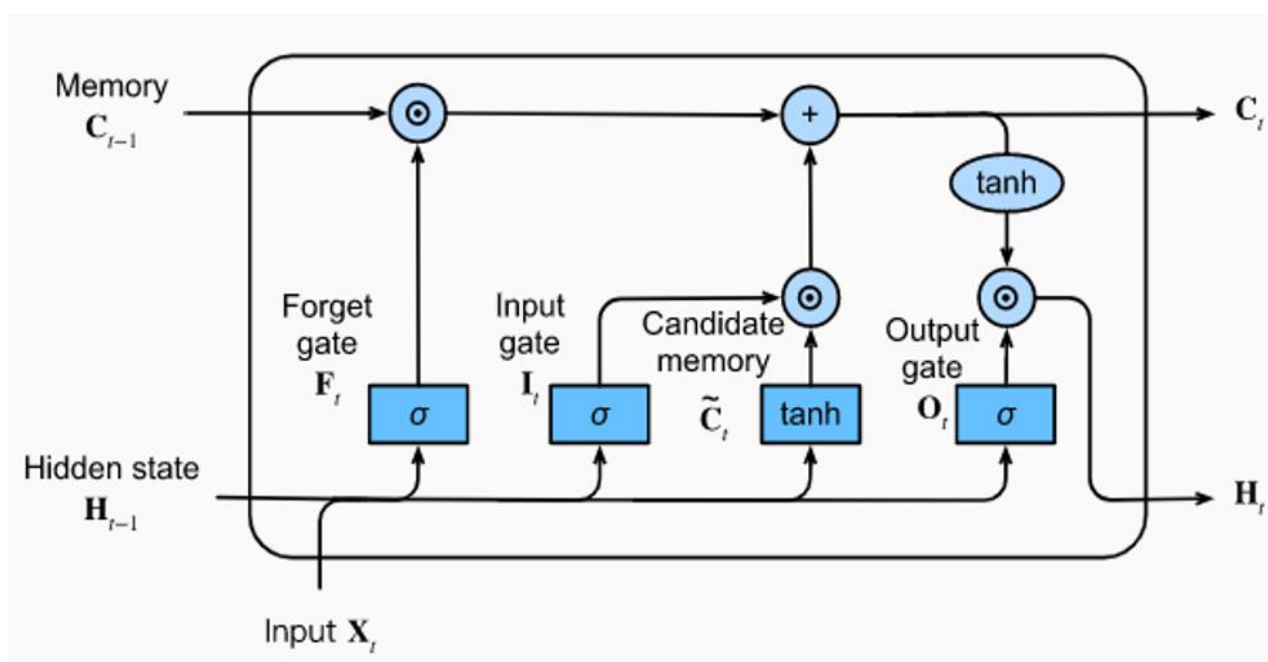


Рисунок 2.3 – Архітектура комірки LSTM [30]

Рівняння оновлення для комірки LSTM на часовому кроці t можна записати наступним чином:

$$\text{Вентиль забування: } f_t = \sigma(W_f \cdot [h_{\{t-1\}}, x_t] + b_f)$$

$$\text{Вентиль входу: } i_t = \sigma(W_i \cdot [h_{\{t-1\}}, x_t] + b_i)$$

Вентиль виходу: $o_t = \sigma(W_o \cdot [h_{\{t-1\}}, x_t] + b_o)$

Оновлення стану комірки: $C_t = f_t \odot C_{\{t-1\}} \odot i_t \odot \tanh(W_c \cdot [h_{\{t-1\}}, x_t] + b_c)$

Оновлення прихованого стану: $h_t = o_t \odot \tanh(C_t)$,

де x_t – вхідні дані на кроці t ;

$h_{\{t-1\}}$ – попередній прихований стан;

$C_{\{t-1\}}$ – попередній стан комірки;

W_f, W_i, W_o, W_c – вагові матриці;

b_f, b_i, b_o, b_c – вектори зсуву;

σ – сигмоїдна функція активації;

\odot – представляє поелементне множення.

Мережі з довгою короткочасною пам'яттю (LSTM) мають ряд сильних і слабких сторін в контексті задач прогнозування часових рядів. До сильних сторін LSTM належать наступні:

- LSTM призначені для виявлення довгострокових залежностей у послідовних даних, що робить їх добре придатними для задач прогнозування часових рядів, де історичний контекст і часові зв'язки мають вирішальне значення;
- вони здатні ефективно обробляти нерегулярні часові інтервали та відсутні значення в даних часових рядів;
- LSTM здатні моделювати складні патерни, включаючи нелінійні зв'язки, сезонність, тренди та нерівномірності;
- такі моделі вирішують проблему зникаючого градієнта, яка часто зустрічається в стандартних рекурентних нейронних мережах (RNN), шляхом введення таких механізмів, як блоки – вентиля та «карусель» з постійною похибкою. Ці механізми дозволяють LSTM пом'якшити проблему зникаючих градієнтів і сприяють більш стабільному та ефективному навчанню на довгих послідовностях.

Також можна визначити такі недоліки моделей LSTM:

- обчислювальна складність, особливо при роботі з великими наборами даних і складними архітектурами з декількома шарами і прихованими блоками – навчання глибоких LSTM може вимагати використання значних обчислювальних ресурсів і часу, тим самим обмежуючи їх масштабованість для великомасштабних задач прогнозування часових рядів;
- LSTM чутливі до гіперпараметрів, таких як швидкість навчання, розмір batch, кількість шарів і кількість прихованих одиниць, тому неоптимальні налаштування параметрів можуть призвести до нестабільності навчання, проблем зі збіжністю або перенавчанням;
- LSTM часто розглядаються як моделі «чорного ящика», що ускладнює інтерпретацію основних механізмів, які керують їхніми прогнозами;
- моделі даного типу можуть мати труднощі з обробкою зашумлених даних часових рядів з високим рівнем випадкових відхилень або викидів, тому наявність шуму в даних може перешкоджати здатності мережі розпізнавати значущі закономірності, що призводить до неточних прогнозів або перенавчання.

Загалом, архітектура мереж LSTM дозволяє їм ефективно моделювати довгострокові залежності, охоплювати складні часові патерни і генерувати точні прогнози для даних часових рядів. Проте такі моделі також не позбавлені обмежень. Вони включають обчислювальну складність, чутливість до гіперпараметрів, інтерпретацію та обробку зашумлених даних.

2.3 Критерії оцінки результатів та інтерпретація прогнозів

Критерії оцінки результатів та інтерпретації прогнозів мають важливе значення для оцінки ефективності та надійності моделей прогнозування часових рядів. Перш ніж використовувати модель прогнозування для прогнозування показників з новими даними, дуже важливо оцінити її точність, використовуючи історичні дані.

Середня абсолютна похибка (MAE) є загальнозживаною метрикою для оцінки точності прогнозів або передбачень. Вона вимірює середню величину похибки між прогнозованими значеннями та фактичними спостережуваними значеннями. MAE виражається в тих самих одиницях, що й вихідні дані, що полегшує його інтерпретацію. [31]

Формула для середньої абсолютної похибки (MAE) має вигляд:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.4)$$

де n – загальна кількість спостережень;

y_i – фактичне спостережуване значення в момент часу i ;

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

$|\cdot|$ позначає абсолютне значення.

Нижче значення MAE вказує на кращу точність, оскільки це означає, що прогнози в середньому ближчі до фактичних спостережуваних значень. І навпаки, вищий MAE свідчить про більші розбіжності між прогнозованими та спостережуваними значеннями, що вказує на нижчу точність моделі прогнозування.

Ще однією метрикою, яку часто використовують для оцінки точності прогнозів або передбачень, є середньоквадратична похибка (MSE) – це статистичний показник, який кількісно вимірює середньоквадратичну різницю

між прогнозованими значеннями та фактичними спостережуваними значеннями. MSE часто використовується в регресійному аналізі та прогнозуванні часових рядів. Середньоквадратична помилка (MSE) розраховується наступним чином:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

де n – загальна кількість спостережень;

y_i – фактичне спостережуване значення в момент часу i ;

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

Нижче значення MSE вказує на більшу точність, оскільки це означає, що прогнози в середньому ближчі до фактичних спостережуваних значень. Однак MSE є більш чутливим до більших помилок, ніж MAE, оскільки він зводить у квадрат різницю між прогнозованими та спостережуваними значеннями. Отже, MSE може бути більш чутливим до впливу викидів або значних розбіжностей у даних.

Середньоквадратична похибка (RMSE) – це метрика, яка часто використовується для оцінки точності прогнозів або передбачень. В даній формулі обчислюється квадратний корінь із середнього квадрата різниці між прогнозованими значеннями та фактичними спостережуваними значеннями. RMSE є аналогом MSE, але надає міру похибки в тих самих одиницях, що й вихідні дані, що робить його більш зрозумілим для інтерпретації [32]. Формула для середньоквадратичної похибки (RMSE) виглядає наступним чином:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.6)$$

де n – загальна кількість спостережень;

y_i – фактичне спостережуване значення в момент часу i ;

\hat{y}_i – передбачене значення в момент часу i .

RMSE обчислює квадратичну різницю між кожним прогнозованим значенням і відповідним йому фактичним спостереженим значенням, підсумовує ці квадратичні різниці за всіма спостереженнями, бере середнє значення, а потім обчислює квадратний корінь з отриманого результату. Остаточне значення RMSE представляє середню величину помилок у тих самих одиницях, що й вихідні дані.

Як і у випадку MAE та MSE, нижче значення RMSE вказує на вищу точність, оскільки означає, що прогнози в середньому ближчі до фактичних спостережуваних значень. RMSE є особливо корисним для порівняння ефективності різних моделей прогнозування або оцінки впливу вдосконалення моделі, оскільки він забезпечує стандартизовану міру помилки прогнозування, яку легко інтерпретувати.

Середня абсолютна відсоткова похибка (MAPE) часто використовується для оцінки точності прогнозів або передбачень, особливо в тих випадках, коли масштаб даних значно відрізняється в різних спостереженнях. MAPE – це показник, який вимірює середню відсоткову різницю між прогнозованими значеннями та фактичними спостережуваними значеннями [33]. Формула середньої абсолютної відсоткової похибки (MAPE) виглядає наступним чином:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (2.7)$$

де n – загальна кількість спостережень;

y_i – фактичне спостережуване значення в момент часу i ;

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

Нижче значення MAPE вказує на більшу точність, оскільки це означає, що прогнози в середньому ближчі до фактичних спостережуваних значень. Однак слід зазначити, що MAPE може не підходити для ситуацій, коли фактичні

спостережувані значення близькі до нуля, оскільки це може призвести до невизначених або нескінченних значень. Крім того, MAPE може посилювати вплив значних похибок, особливо коли фактичні спостережувані значення є мінімальними. Отже, надзвичайно важливо інтерпретувати MAPE в контексті конкретного завдання прогнозування та масштабу даних.

Показник Precision стосується узгодженості та надійності прогнозів у часі. Ця метрика оцінює, чи відповідають прогнози бажаному рівню точності або перевищують його.

Показник Recall – це міра стабільності та узагальнюваності моделі прогнозування для різних наборів даних, часових періодів та умов.

Також використовується аналіз залишків – це метод, який досліджує розбіжності між прогнозованими значеннями та фактичними спостереженнями (залишками). При цьому оцінюється, чи демонструють залишки будь-які систематичні закономірності, такі як автокореляція, гетероскедастичність або ненормальність, які можуть свідчити про те, що модель невірно визначена.

Процес інтерпретації прогнозів передбачає аналіз прогнозів моделі в контексті вихідних даних, бізнес-цілей і процесів прийняття рішень. Цей процес передбачає розуміння невизначеності, пов'язаної з прогнозами, виявлення потенційних джерел помилок або упередженості, а також оцінку впливу результатів прогнозу на результати рішень.

Висновки до розділу 2

Отже, процес розробки інтелектуальних систем для прогнозування фінансових ринків складається з низки важливих етапів. Ці кроки включають методи попередньої обробки даних, такі як фільтрація часових рядів, відбір релевантних ознак, визначення оптимальної кількості ознак, обробка пропущених даних та оцінка стаціонарності даних часових рядів.

Фільтрація часових рядів передбачає видалення шуму та нерелевантної інформації з даних, що покращує співвідношення сигнал/шум і підвищує точність прогнозування моделі. Відбір релевантних ознак має першорядне значення для виявлення значущих закономірностей і взаємозв'язків у наборі даних. І навпаки, визначення оптимальної кількості ознак слугує для запобігання надмірному пристосуванню та підвищенню узагальнюваності моделі.

Обробка пропущених даних має вирішальне значення для збереження цілісності даних і забезпечення стійкості моделі до недосконалості реальних даних. Оцінка стаціонарності часових рядів передбачає оцінку стабільності статистичних властивостей у часі, що є передумовою для ефективного застосування певних методів прогнозування.

Критерії оцінки результатів та інтерпретації прогнозів відіграють ключову роль в оцінці точності, достовірності, стійкості та надійності моделей прогнозування. Такі показники, як середня абсолютна похибка (MAE), середня квадратична похибка (MSE), середньоквадратична похибка (RMSE) і середня абсолютна відсоткова похибка (MAPE) забезпечують кількісну оцінку точності прогнозу.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДЕЛІ

3.1 Вибір методології для прогнозування фінансових ринків

Вибір відповідної методології для прогнозування фінансових ринків має першочергове значення для розробки точних і надійних моделей для прогнозування. Python, у поєднанні з широким набором різноманітних бібліотек та інструментів, став дуже привабливим варіантом серед розробників та аналітиків. Це пов'язано з його універсальністю, зручністю у використанні, широкою екосистемою та надійною підтримкою для аналізу даних, машинного навчання та глибокого навчання.

Python оснащений широким набором бібліотек та фреймворків, які були розроблені з урахуванням специфічних потреб Data Science, машинного навчання та аналітики. Такі бібліотеки, як pandas, NumPy та scikit-learn надають важливі інструменти для маніпулювання даними, чисельних обчислень та машинного навчання. Pandas є універсальною бібліотекою для маніпулювання та аналізу даних, особливо добре підходить для роботи з даними часових рядів. Вона пропонує ефективні структури даних, такі як DataFrame та Series, а також інтуїтивно зрозумілі функції для очищення, фільтрації, агрегування та перетворення даних, що робить її незамінною для попередньої обробки даних для побудови моделей.

Широкі можливості Python для візуалізації ще більше розширюються завдяки включенню потужних бібліотек візуалізації, таких як Matplotlib. Ці бібліотеки дозволяють створювати наочні та інформативні діаграми, графіки, полегшуючи, таким чином, представлення складних даних у зрозумілій та привабливій формі. Використання візуалізації має першочергове значення в контексті дослідницького аналізу даних, оцінки моделей та ефективного подання інформації. Matplotlib – популярна бібліотека для побудови графіків, яка дозволяє створювати широкий спектр високоякісних візуалізацій, включаючи графіки часових рядів, гістограми, діаграми розсіювання тощо. Використання

візуалізації має велике значення в контексті дослідницького аналізу даних, діагностики моделей та ефективного представлення результатів досліджень.

Python пропонує надійну підтримку статистичного аналізу, аналізу часових рядів та методів прогнозування. Бібліотеки для Python включають повний набір статистичних функцій, фільтрів та алгоритмів для аналізу даних часових рядів, підбору моделей прогнозування та оцінки їхньої ефективності. До них належать бібліотеки statsmodels, scipy та scikit-learn, які надають ряд статистичних інструментів для аналізу даних часових рядів. Фільтри Scipy використані для обробки сигналів. До них відносяться фільтри Баттерворта, фільтри Савицького-Голея та інші фільтри. Дані фільтри дуже корисні для видалення шуму і згладжування сигналів.

Інтеграція Python з популярними фреймворками глибокого навчання, такими як TensorFlow, дозволяє розробляти та впроваджувати передові нейромереві моделі для прогнозування часових рядів. Ці фреймворки пропонують масштабовані та ефективні інструменти для побудови складних архітектур, включаючи рекурентні нейронні мережі (RNN), мережі з довгою короткочасною пам'яттю (LSTM) та згорткові нейронні мережі (CNN). Keras і TensorFlow, два найпоширеніші фреймворки глибокого навчання на Python, є найкращими варіантами для побудови та навчання нейромеревих моделей для задач прогнозування фінансових показників.

Гнучкість і простота використання Keras, бібліотеки глибокого навчання з відкритим вихідним кодом, є двома її ключовими перевагами. Зручний та інтуїтивно зрозумілий перелік функцій дозволяє користувачам створювати та навчати нейронні мережі з мінімальними зусиллями. Високорівневий API дозволяє дослідникам і користувачам швидко створювати прототипи та експериментувати з різними архітектурами та конфігураціями. TensorFlow, як основний внутрішній механізм Keras, надає низькорівневі функції для визначення кастомних операцій, оптимізації продуктивності та впровадження моделей у робочі середовища.

TensorFlow використовує концепцію обчислювальних графів для полегшення виконання обчислень на CPU і GPU, що робить його чудовим вибором для навчання глибоких нейронних мереж з великими наборами даних. Його можливості автоматичного диференціювання полегшують використання алгоритмів градієнтної оптимізації, таких як стохастичний градієнтний спуск (SGD) та Adam, для ефективного навчання нейромережових моделей.

Keras і TensorFlow пропонують користувачам гнучкість у визначенні кастомних шарів, функцій втрат і метрик, що дозволяє створювати власні архітектури нейронних мереж, пристосовані до характеристик даних фінансових часових рядів. Ця гнучкість дозволяє включати специфічні знання, методи інженерії особливостей і стратегії регуляризації, які в сукупності покращують продуктивність і надійність моделі.

Бібліотека Transformers, разом з відповідними моделями та токенизаторами, надає можливість отримати набір потужних інструментів для обробки природної мови, зокрема, для аналізу настроїв, класифікації новин і прогнозування на основі тексту. Використовуючи попередньо навчені трансформери, такі як BERT, GPT або RoBERTa, можна отримати цінну інформацію з текстових джерел даних, таких як новини, стрічки соціальних мереж та звіти про прибутки і збитки. Це дозволяє їм глибше зрозуміти інформацію, що міститься в них. Це дозволяє інтегрувати якісну інформацію та аналіз ринкових настроїв у моделі фінансового прогнозування, підвищуючи їх точність та надійність.

Додатково, бібліотека ufinance пропонує зручний доступ до історичних ринкових даних, включаючи ціни на акції через Yahoo Finance API. Завдяки ній отримано датасет із історичними цінами акцій компанії.

3.2 Робота із даними

Під час дослідження було проаналізовано історичні архіви цін на акції TESLA, щоб оцінити ефективність поточних методів прогнозування та визначити можливості для вдосконалення шляхом включення заголовків новин. Набір даних охоплює щоденні показники акцій, включаючи дату, відкриття, максимум, мінімум, закриття, обсяг, дивіденди та дроблення акцій, забезпечуючи широке та детальне джерело історичних ринкових даних для аналізу та прогнозування (рис. 3.1).

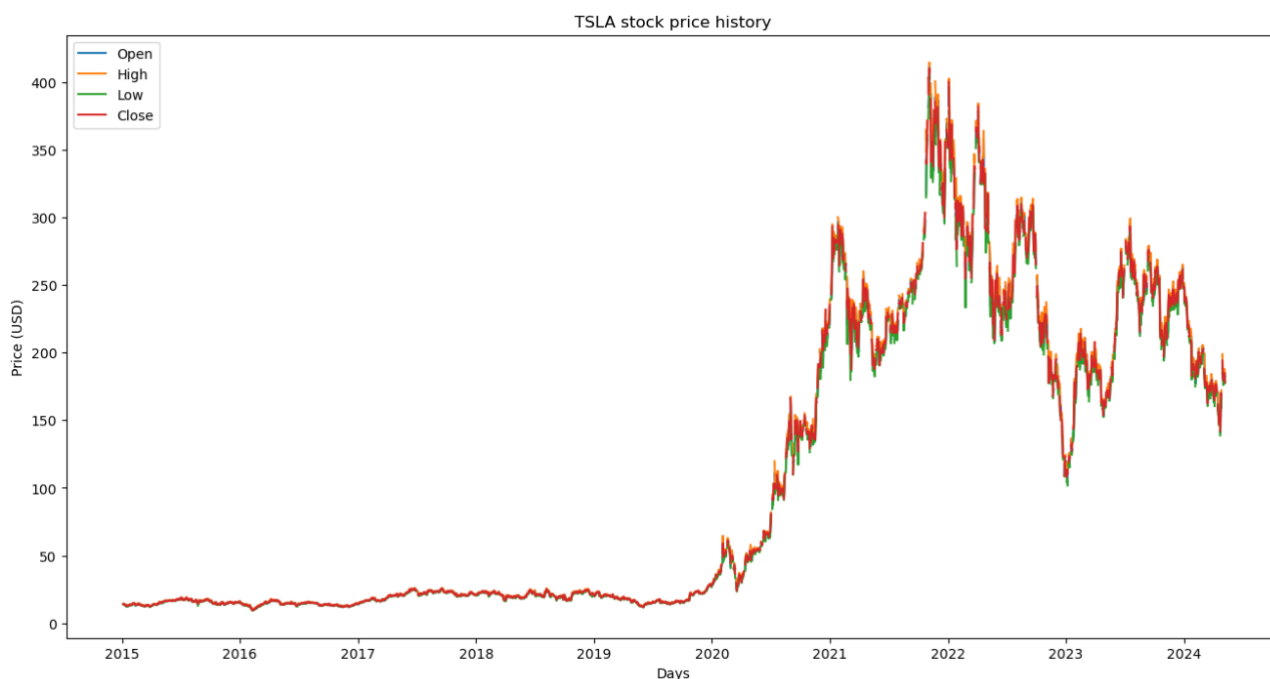


Рисунок 3.1 – Графік зміни цін акцій компанії (побудовано за даними [34])

Бібліотека `matplotlib` використовується для створення інформативних графіків, які відображають часову еволюцію цін на акції TESLA. Зокрема, створюються графіки, які ілюструють тенденції цін відкриття, максимуму, мінімуму та закриття протягом усього періоду часу, охопленого набором даних (з 1 січня 2015 року по 8 травня 2024 року). На графіку (рис. 3.1) можна побачити

характерні закономірності та коливання цін відкриття, максимуму, мінімуму та закриття, що дає цінну інформацію про історичні зміни цін на акції Tesla.

3.2.1 Обґрунтування вибору даних

Вибір цін на акції TESLA в якості набору даних для прогнозування показників фінансового ринку підтримується кількома вагомими причинами, кожна з яких робить свій внесок в актуальність і придатність набору даних для теми дослідження.

TESLA посідає чільне місце на світових фінансових ринках. Її акції широко торгуються і за ними уважно стежать інвестори, аналітики та учасники ринку. Вибравши ціни на акції TESLA в якості набору даних, дослідження фокусується на гучному та впливовому активі, який відображає реальну динаміку та тенденції на фінансових ринках.

Ціни на акції TESLA забезпечують джерело історичних ринкових даних, що охоплюють багато років і різні ринкові умови, економічні цикли і геополітичні події. Наявність вичерпних історичних даних полегшує поглиблений аналіз, моделювання та прогнозування ефективності акцій TESLA, дозволяючи виявляти довгострокові тенденції та закономірності.

Акції TESLA відомі своєю волатильністю та чутливістю до широкого спектру факторів, включаючи технологічні інновації, регуляторні зміни, макроекономічні тенденції та настрої інвесторів. Вивчаючи ціни на акції TESLA, дослідження досліджує тонкощі ринкової динаміки та фактори, що впливають на рух цін на акції, надаючи цінну інформацію про поведінку фінансових ринків.

Ціни на акції TESLA можуть бути доповнені альтернативними джерелами даних, такими як заголовки новин, настрої в соціальних мережах і фундаментальні показники, щоб підвищити точність і надійність прогнозу. Інтегруючи ці різні способи отримання даних, у ході дослідження були

проаналізовані інноваційні підходи до прогнозування фінансових ринків, використовуючи синергію між традиційними фінансовими показниками та якісними спостереженнями.

На додаток до зазначених причин, вибір цін на акції TESLA з Yahoo Finance [34] ще більше підвищує придатність набору даних для прогнозування показників фінансового ринку. Кілька ключових факторів вплинули на придатність цього джерела даних:

- набір даних про акції TESLA, отриманий з Yahoo Finance, включає всі поля, необхідні для прогнозування, в тому числі дату, відкриття, максимум, мінімум, закриття та обсяг;
- незважаючи на проблеми, пов'язані з реальними фінансовими даними, набір даних про акції TESLA від Yahoo Finance було перевірено для усунення пропущених точок даних і забезпечення повноти даних. Аналогічно дані можуть бути оброблені для будь-якого іншого датаеста, що вказує на універсальність даного набору даних.

3.2.2 Підготовка даних для дослідження

Підготовка даних для дослідження передбачає забезпечення чистоти, повноти та готовності набору даних до аналізу. У випадку з набором даних про акції компанії TESLA з Yahoo Finance, важливо заповнити відсутні значення, щоб зберегти цілісність даних і сприяти точному прогнозуванню.

Першим кроком є визначення та локалізація відсутніх значень у наборі даних про акції TESLA. Відсутні значення можуть виникати з різних причин, включаючи помилки при введенні даних, технічні проблеми зі збором даних або інші причини. Вивчаючи набір даних, було визначено місця та закономірності

відсутніх значень для різних показників ціни акцій, зокрема, ціни відкриття, закриття, максимуму, мінімуму, обсягу та дивідендів.

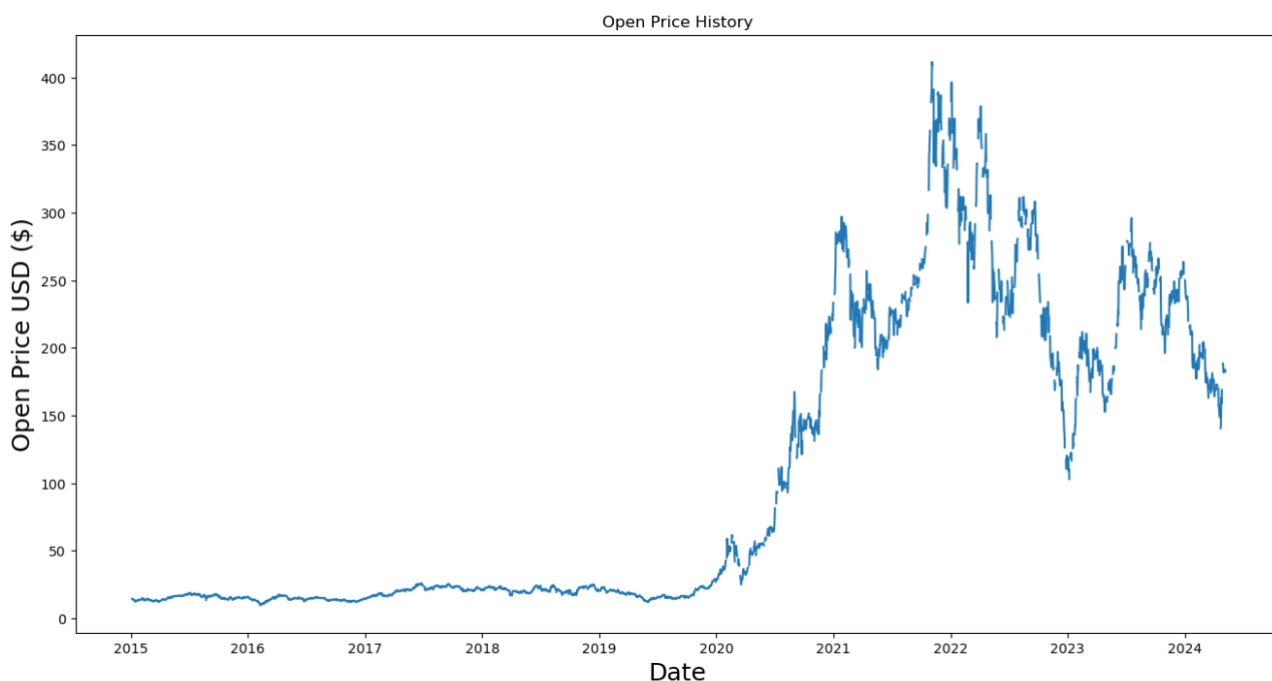


Рисунок 3.2 – Графік відкриття цін на акції із відсутніми значеннями
(побудовано за даними [34])

Для заповнення пропущених значень було використано метод заповнення середнім значенням із найближчих спостережуваних значень за попередні та наступні торгові дні. Такий підхід гарантує, що відсутні значення інтерполюються на основі навколишніх точок даних, тим самим зберігаючи загальну тенденцію та динаміку цін на акції.

В аналізованому наборі даних відсутні цілі рядки даних через відсутність значень на певні дати торгів. Щоб вирішити цю проблему, було заповнено цілі рядки даних шляхом екстраполяції значень для всіх метрик цін акцій (відкриття, закриття, максимум, мінімум, обсяг, дивіденди) на основі середнього значення найближчих спостережуваних значень.

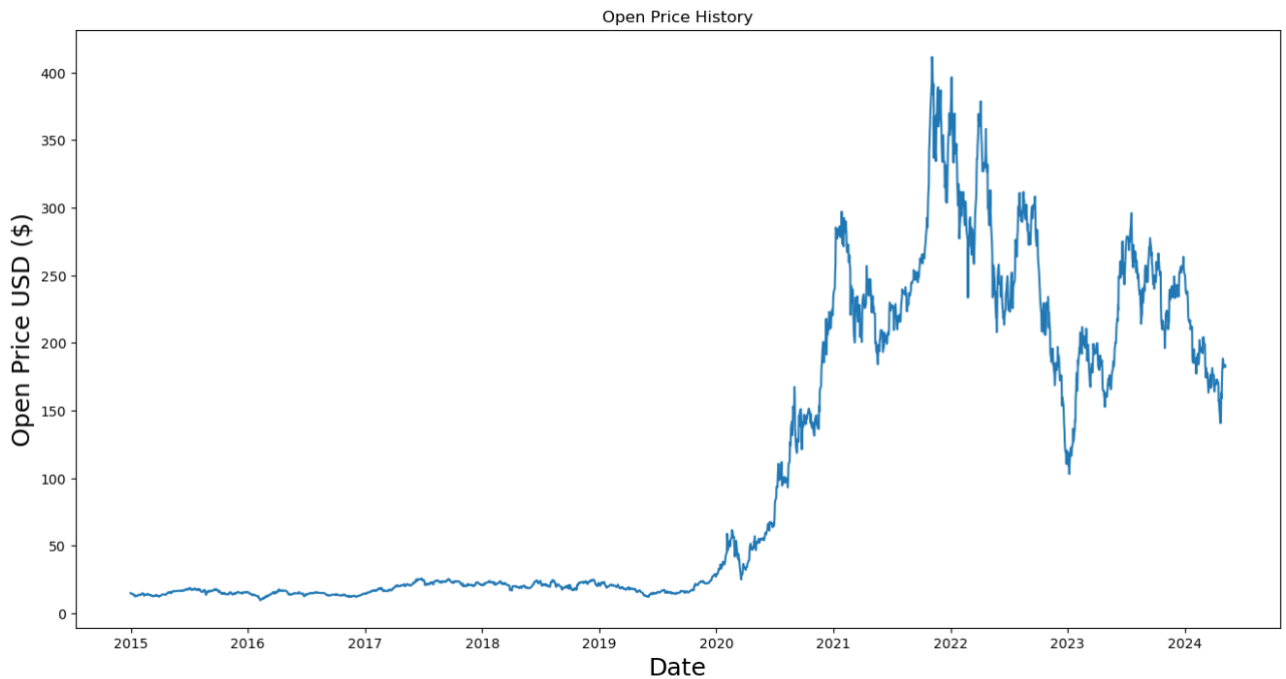


Рисунок 3.3 – Графік відкриття цін на акції після інтерполяції

Після інтерполяції відсутніх значень у наборі даних TESLA наступним кроком у підготовці є фільтрація даних для покращення якості набору даних для моделювання та аналізу. Для цього було застосовано цифровий низькочастотний фільтр з використанням функцій `butter` та `filter` бібліотеки SciPy.

Цифровий фільтр низьких частот – це метод обробки сигналу, який використовується для послаблення високочастотних компонентів у сигналі зі збереженням низькочастотних компонентів. У контексті фінансових даних, таких як ціни на акції, застосування фільтра низьких частот допомагає згладити короткострокові коливання або шум у даних, виявляючи основні тенденції та закономірності, які є більш релевантними для цілей прогнозування.

Бібліотека SciPy надає функцію `Butter` для створення цифрових фільтрів Баттерворта, які зазвичай використовуються для фільтрації низьких частот завдяки своїй пласкій частотній характеристиці та мінімальним фазовим спотворенням. Після обчислення коефіцієнтів фільтра за допомогою функції `Butter`, функція фільтра використовується для застосування фільтра до набору

даних про акції TESLA, ефективно послаблюючи високочастотний шум і створюючи дещо згладжену версію графіка цін на акції.

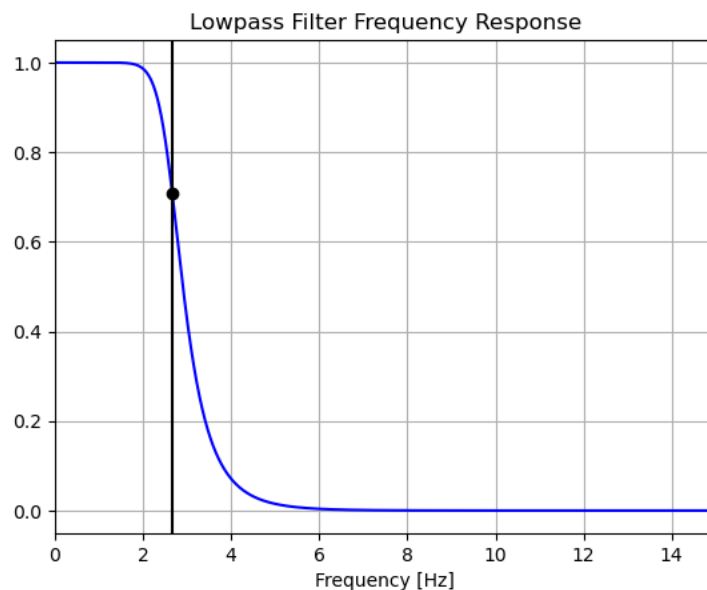


Рисунок 3.4 – Частотна характеристика фільтра низьких частот

Застосовуючи низькочастотний фільтр до набору даних про акції TESLA, було отримано більш згладжене представлення базової динаміки цін на акції, що є більш сприятливим для навчання та аналізу моделі. Відфільтрований набір даних допомагає пом'якшити вплив короткострокових коливань і шуму, забезпечуючи більш чіткий сигнал для моделей прогнозування, який можна вловлювати і на якому можна вчитися.

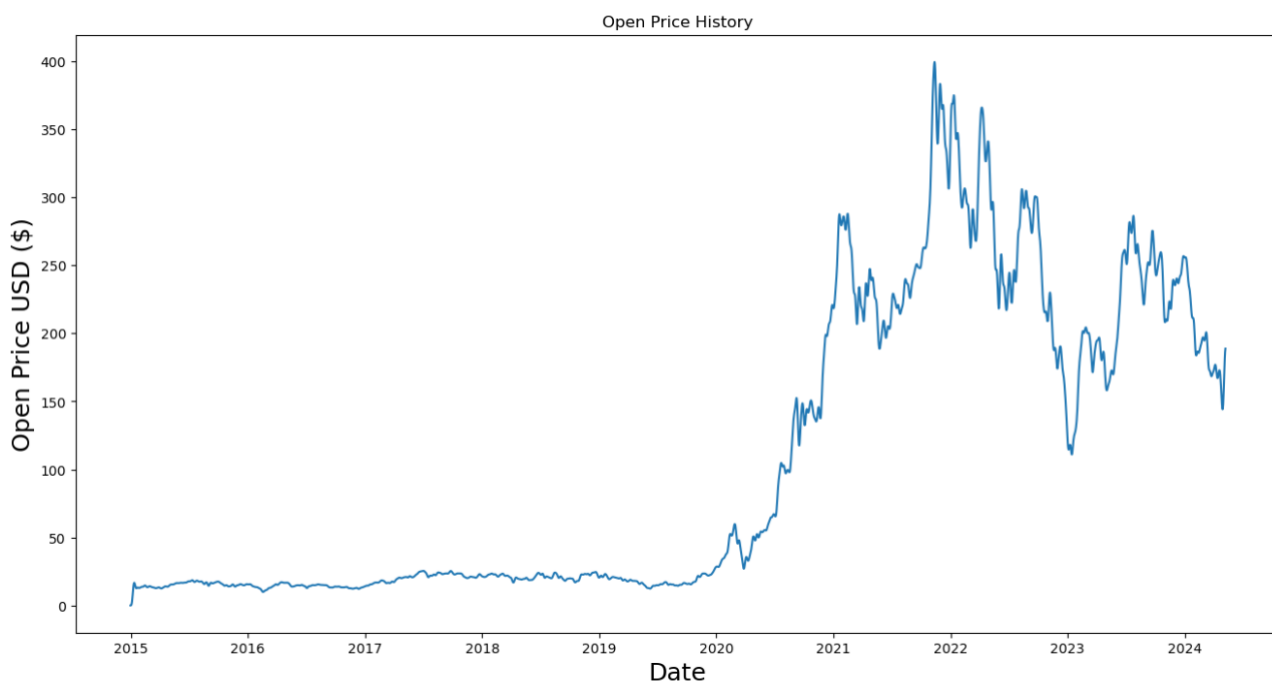


Рисунок 3.5 – Графік відкриття цін на акції після фільтрації

Використання відфільтрованого набору даних покращує продуктивність прогнозних моделей, зменшуючи перенавчання на короткочасний шум і покращуючи здатність моделі виявляти значущі тенденції та закономірності в даних. Навчаючи моделі на згладженій версії набору даних, сприяємо узагальненню та покращуємо точність прогнозування моделі на нових даних.

Після застосування цифрового фільтра низьких частот до набору даних про акції TESLA, наступним важливим кроком у попередній обробці даних є оцінка стаціонарності часового ряду. Стаціонарність – це фундаментальне припущення в аналізі часових рядів, яке вказує на те, що статистичні властивості даних залишаються незмінними в часі. Широко використовуваним методом перевірки стаціонарності є тест Дікі-Фуллера.

Запустивши тест Дікі-Фуллера на відфільтрованому наборі даних про акції TESLA, можемо оцінити, чи демонструє набір даних стаціонарну поведінку. Отримаємо значення p-value 0,634, що більше за критичне значення 0.05, що вказує на те, що гіпотеза про стаціонарність часового ряду відхилена (рис. 3.6).

```

Results of Dickey–Fuller Test:
Test Statistic           -1.289062
p-value                  0.634162
#Lags Used               30.000000
Number of Observations Used  3385.000000
Critical Value (1%)      -3.432283
Critical Value (5%)      -2.862394
Critical Value (10%)     -2.567225
dtype: float64
No, the time series is not stationary.

```

Рисунок 3.6 – Перевірка стаціонарності часового ряду

Після аналізу стаціонарності за допомогою тесту Дікі-Фуллера, наступним кроком на етапі попередньої обробки даних є перевірка наявності сезонності в наборі даних про акції TESLA. Сезонність – це повторювані закономірності або коливання, які відбуваються через регулярні проміжки часу в даних часових рядів, часто зумовлені зовнішніми факторами, такими як календарні ефекти або циклічні тенденції на ринку.

Щоб дослідити наявність сезонності в наборі даних про акції TESLA, було використано сезонну декомпозицію – метод, який розкладає часовий ряд на три компоненти: тренд, сезонність і залишки (або шум). Ця декомпозиція дозволила виокремити та проаналізувати основні сезонні закономірності в даних (рис. 3.7).

Сезонна декомпозиція розділяє вихідні дані часового ряду на складові компоненти, що дозволяє дослідити кожен компонент окремо. У цьому аналізі було використано мультиплікативну сезонну декомпозицію – метод, який передбачає, що сезонний компонент змінюється пропорційно до рівня часового ряду. Цей підхід особливо підходить для даних часових рядів, де амплітуда сезонних коливань змінюється з рівнем ряду.

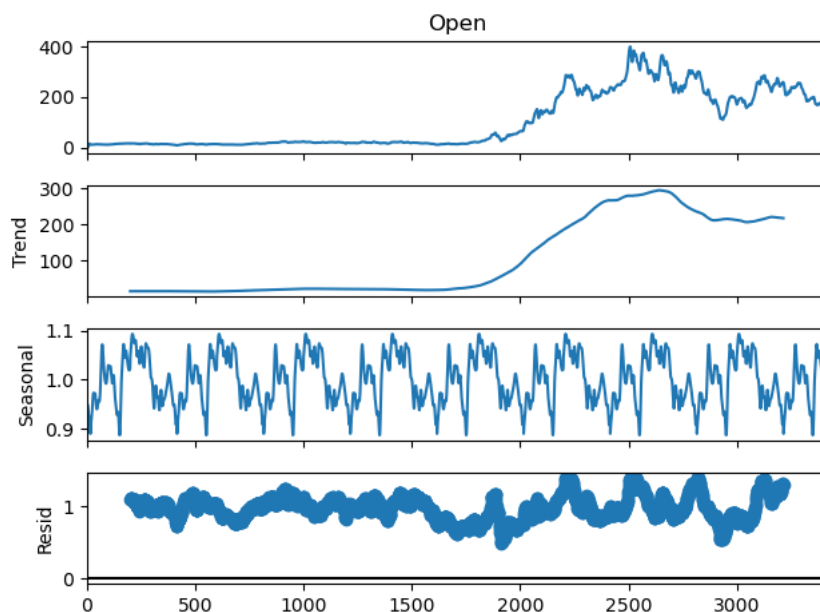


Рисунок 3.7 – Сезонна декомпозиція часового ряду

Ретельно підготувавши набір даних про акції TESLA за допомогою цих кроків попередньої обробки, було створено фундамент для подальшого аналізу та прогнозування. Чистий і попередньо оброблений набір даних, вільний від пропущених значень і короткострокового шуму, дозволив розробляти точні і надійні моделі прогнозування, що в кінцевому підсумку дає можливість учасникам приймати обґрунтовані рішення на фінансових ринках.

3.3 Аналіз інтелектуальних систем для прогнозування показників

Для прогнозування показників фінансових ринків на прикладі часового ряду акцій TESLA, буде використано такі моделі як WMA (weighted median average), SARMIAX, MLP, RNN, LSTM. Також буде наведено запропоновану модель для прогнозування показників часового ряду із використанням новин, оброблених за допомогою LLM для визначення настроїв даних у новинах та використання оброблених да них додатково для покращення показників

прогнозування. Загальною метою є оцінка сильних і слабких сторін існуючих моделей та пропозиція нових підходів до їх вдосконалення.

Для порівняння будуть використані різні метрики ефективності, що використовуються для оцінки предиктивних моделей, зокрема середня абсолютну похибку (MAE), середня квадратичну похибку (RMSE), середня абсолютна відсоткова похибка (MAPE).

3.3.1 Порівняльний аналіз результатів використання наявних методик

Для вирішення поставлено завдання спочатку було використано метод середньозваженої медіанної середньої (Weighted Median Average, WMA). Проте WMA не може враховувати складні закономірності або довгострокові залежності в даних. Особливо це помітно при прогнозуванні цін на акції на довшому проміжку часу (рис. 3.8).



Рисунок 3.8 – Графік прогнозу із використанням WMA

Відразу можна побачити, що дана модель не впоралась із прогнозуванням наступних показників цін відкриття. Дана модель оцінювала наступні значення на основі 5 останніх спостережень із відповідними вагами 1, 2, 3, 4, 5 для кожного з попередніх значень.

Наступною моделлю, яка була використана – SARIMAX (сезонна авторегресійна інтегрована ковзна середня з екзогенними змінними). Дана модель включає сезонні, авторегресійні та ковзні середні компоненти разом з екзогенними змінними. Вона може моделювати як лінійні, так і нелінійні зв'язки в даних і може враховувати зовнішні фактори, що впливають на часовий ряд (рис. 3.9).



Рисунок 3.9 – Графік прогнозу із використанням SARIMAX

Незважаючи на те, що дана модель показала кращий результат, ніж WMA і модель показує тренд у зміні ціни, результат не достатній для прогнозування показників на великому проміжку часу.

Далі використано модель багатошарового перцептрона (MLP). Він може вловлювати складні нелінійні зв'язки в даних і адаптуватися до різних вхідних даних. Проте він не може знаходити гарно зв'язки у послідовностях, тому для нього використано період ковзного вікна у 5 значень (рис. 3.10; 3.11).

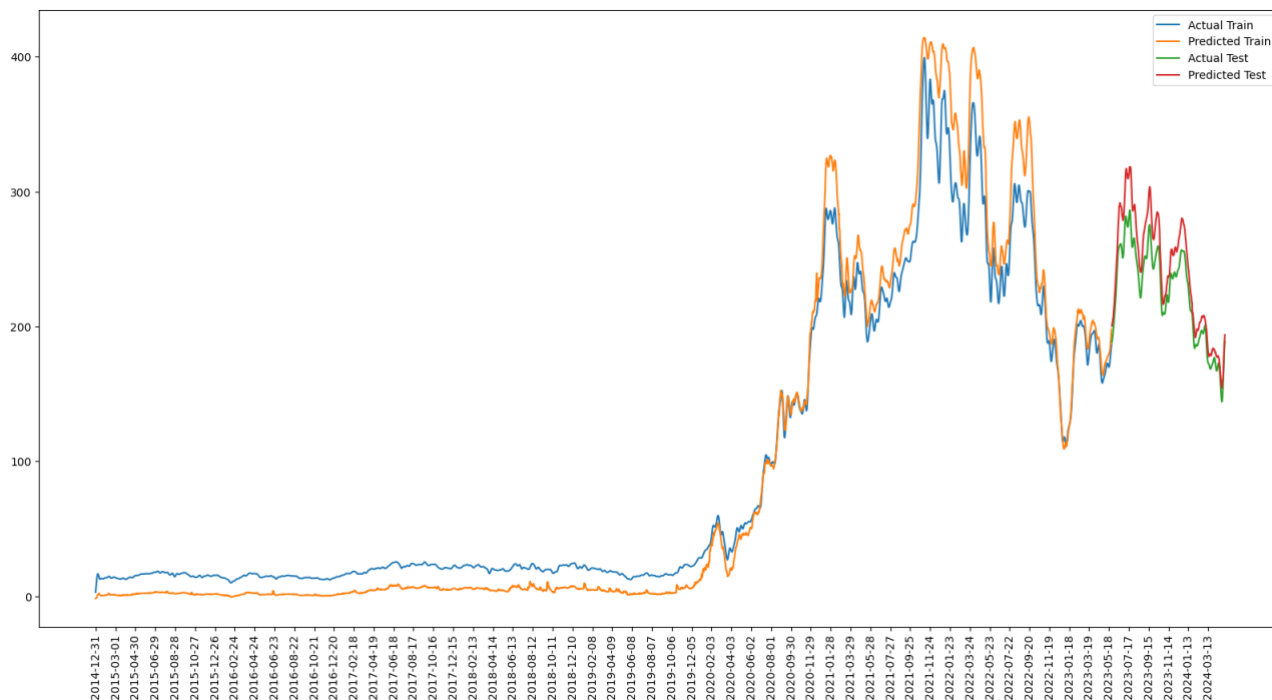


Рисунок 3.10 – Графік прогнозу із використанням MLP

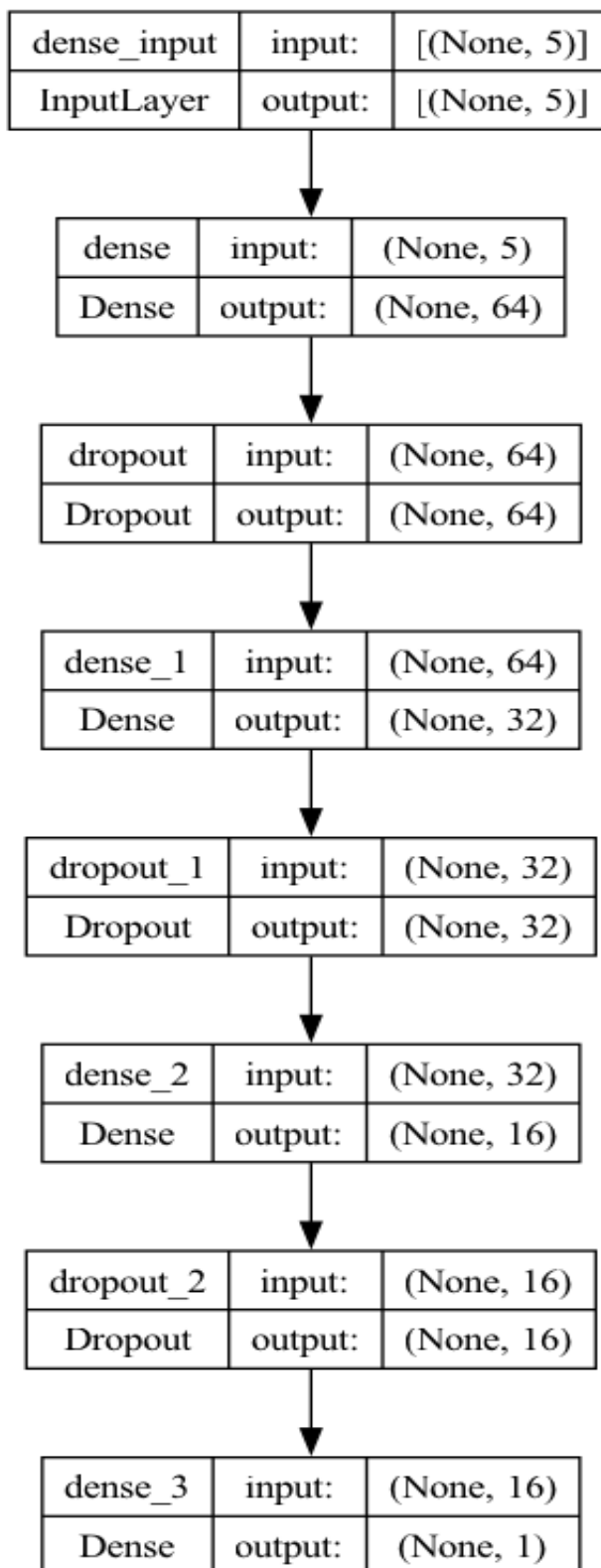


Рисунок 3.11 – Граф архітектури створеного багаточарового перцептрона

Як можна побачити з графіку прогнозу та реальних значень (рис. 3.10), багатошаровий перцептрон непогано впорався із задачею прогнозування значень часового ряду. Проте є досить сильні неточності із прогнозуванням підвищення значень часового ряду.

Для покращення прогнозу MLP, використано рекурентну нейронну мережу (RNN). Дана архітектура призначена для обробки послідовних даних шляхом включення циклів зворотного зв'язку. Проте RNN можуть стикатися з проблемами зникаючого або вибухаючого градієнта, що призводить до труднощів у навчанні довгих послідовностей даних (рис. 3.12; 3.13).

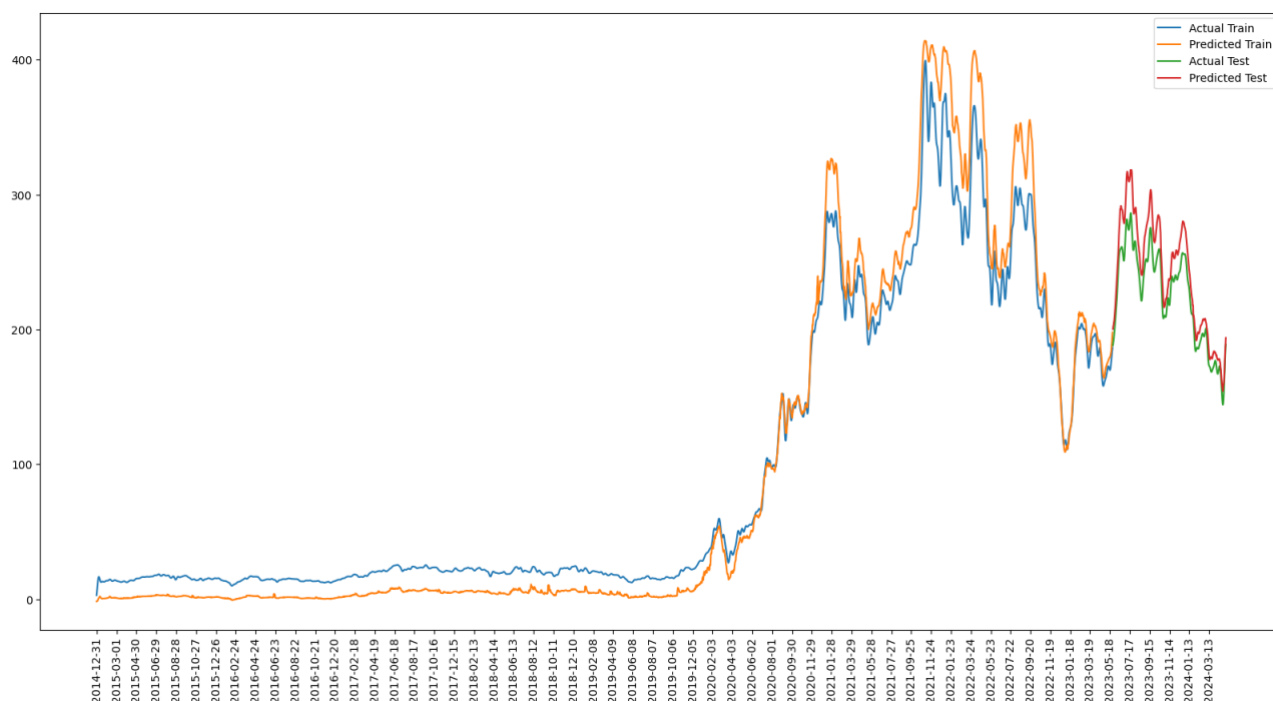


Рисунок 3.12 – Графік прогнозу із використанням RNN

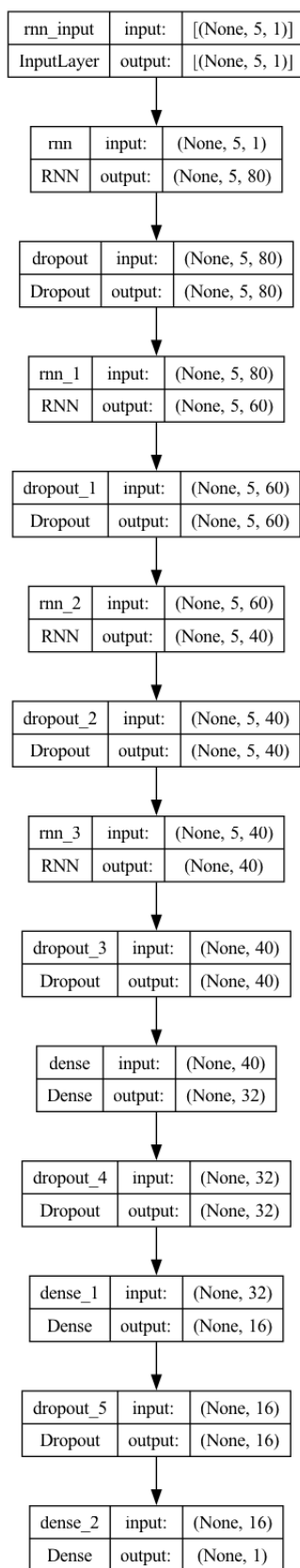


Рисунок 3.13 – Граф архітектури створеної моделі з RNN

Як можна побачити на рис. 3.12, RNN краще впоралась із задачею прогнозування значень часового ряду, адже ця архітектура створена із метою прогнозування послідовних значень ряду, на відміну від архітектури MLP.

Для вирішення проблем, які виникають при навчанні RNN, і покращення прогнозування значень часового ряду, використаємо мережу з довгою короткочасною пам'яттю (LSTM). Це спеціалізований тип архітектури RNN, який вирішує проблему зникаючого градієнта і здатний навчатися довготривалим залежностям (рис. 3.14; 3.15).

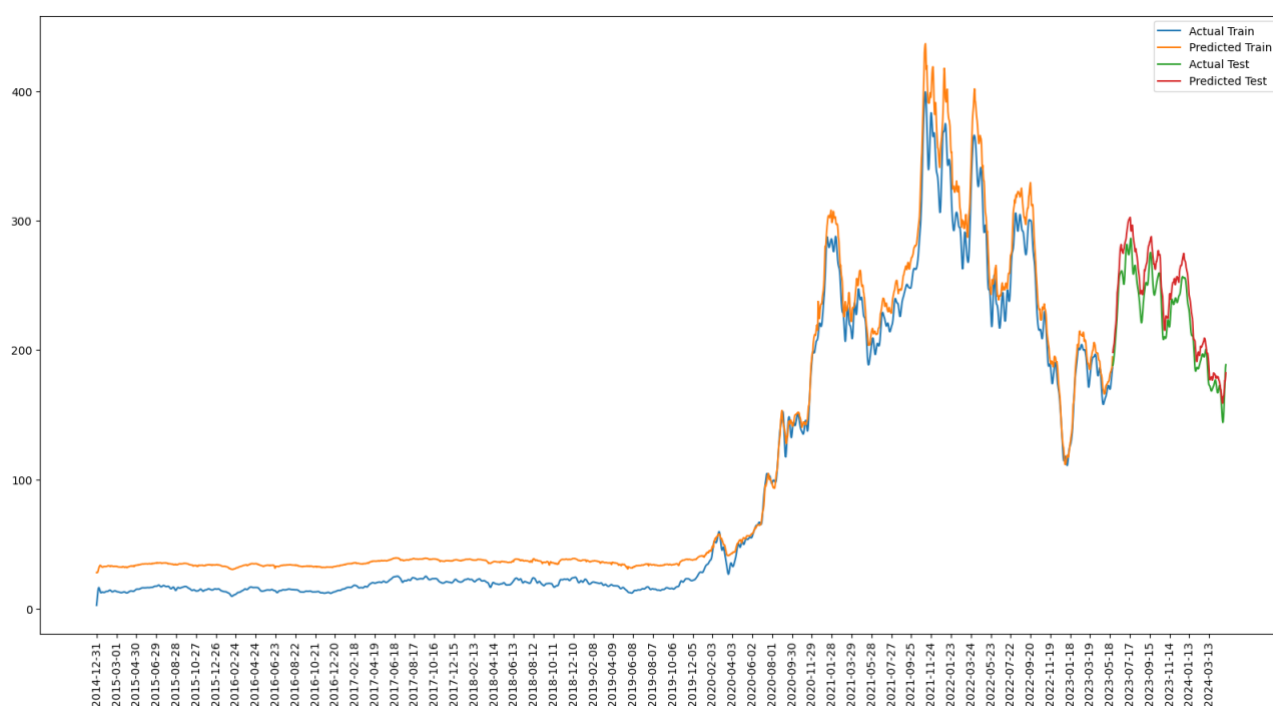


Рисунок 3.14 – Графік прогнозу із використанням LSTM

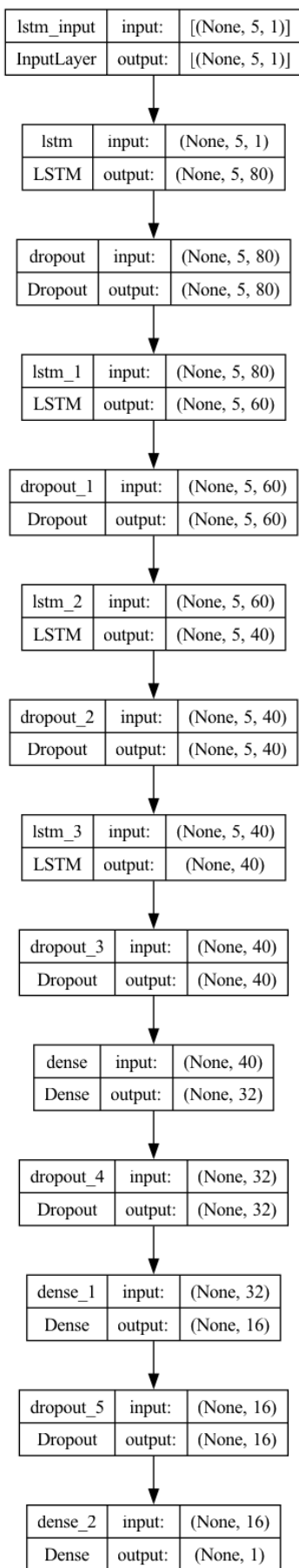


Рисунок 3.15 – Граф архітектури створеної моделі з LSTM

На рис. 3.14 також видно покращення прогнозу у порівнянні із прогнозом моделі RNN. При різкій зміні значення активу часового ряду, прогноз LSTM краще відповідає реальному значенню, ніж прогноз з RNN. При тому кількість параметрів навчання у блоках даних мережах залишилась незмінною. Єдине що змінилось – архітектура самого блоку мережі. Підсумкові порівняльні результати наведено у табл. 3.1.

Таблиця 3.1 – Результати оцінки точності на тестовому сеті

Модель	RMSE	Accuracy	MAPE
WMA	16.9626031214389	83.94%	1510.47%
SARIMAX	51.96154917116397	76.91%	20.40%
MLP	25.14	89.59%	37.28%
RNN	19.26	91.44%	7.37%
LSTM	15.14	93.27%	6.07%

Тоді можемо зробити порівняльний висновок щодо точності моделей на обробленому датасеті із оцінкою точності на тестовій частині даних:

- найкраще спрацювала модель LSTM через її покращення над звичайною рекурентною моделлю;
- моделі WMA та SARIMA спрацювали досить погано на такому складному датасеті через те, що в даному випадку необхідно знаходити закономірності у даних для прогнозу на великі проміжки часу;
- багатошарова нейронна мережа спрацювало достатньо добре, зважаючи на те, що дана модель в основному використовується для прогнозування інших типів даних.

У підсумку, в цьому підрозділі представлено порівняльний аналіз результатів, отриманих за допомогою існуючих методів прогнозування, включаючи WMA, SARIMAX, MLP, RNN та LSTM. Оцінивши їхню

ефективність за різними метриками та розглянувши їхні переваги та обмеження, показано результати наразі створених підходів.

3.3.2 Запропонована модель прогнозування показників фінансових ринків на основі методів штучного інтелекту

За результатами проведеного дослідження було запропоновано нову модель для прогнозування індикаторів фінансового ринку, яка використовує методи штучного інтелекту, зокрема методи обробки природної мови (NLP) з архітектурою глибокого навчання (рис. 3.16). Запропонована нами модель поєднує передові можливості NLP, такі як аналіз настроїв та контекстуальне розуміння заголовків новин, з традиційними методами прогнозування часових рядів для підвищення точності та надійності прогнозів.

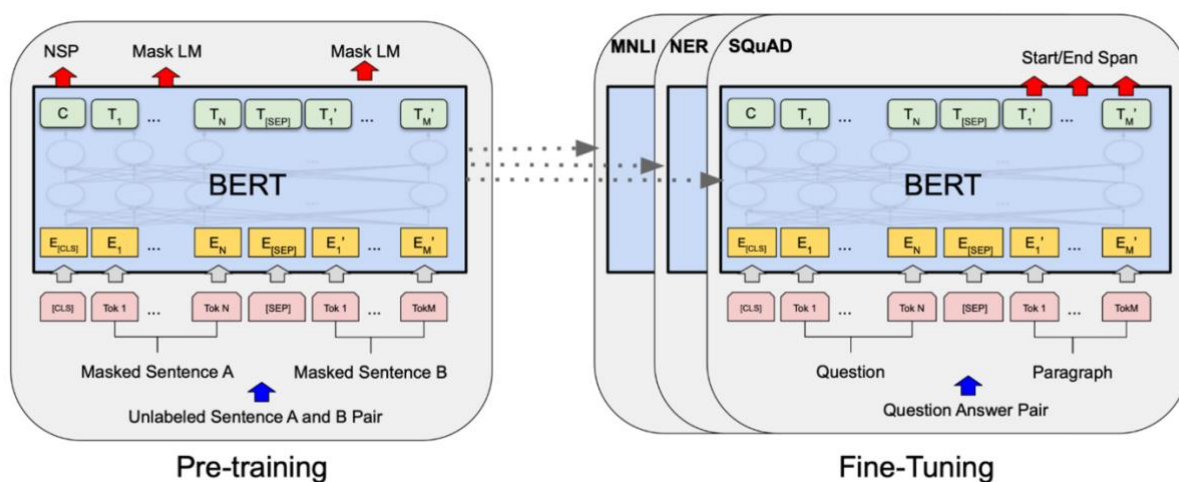


Рисунок 3.16 – Модель BERT [35]

Пропонуємо використовувати переднавчену архітектуру Bidirectional Encoder Representations from Transformers (BERT) від `nlptown`, найсучаснішої моделі NLP, відомої своєю здатністю розуміти контекст і семантику тексту природною мовою. BERT буде використовуватися для обробки заголовків новин фінансового ринку, вилучаючи з текстових даних цінні ідеї, настрої та контекстну інформацію (рис. 3.17).

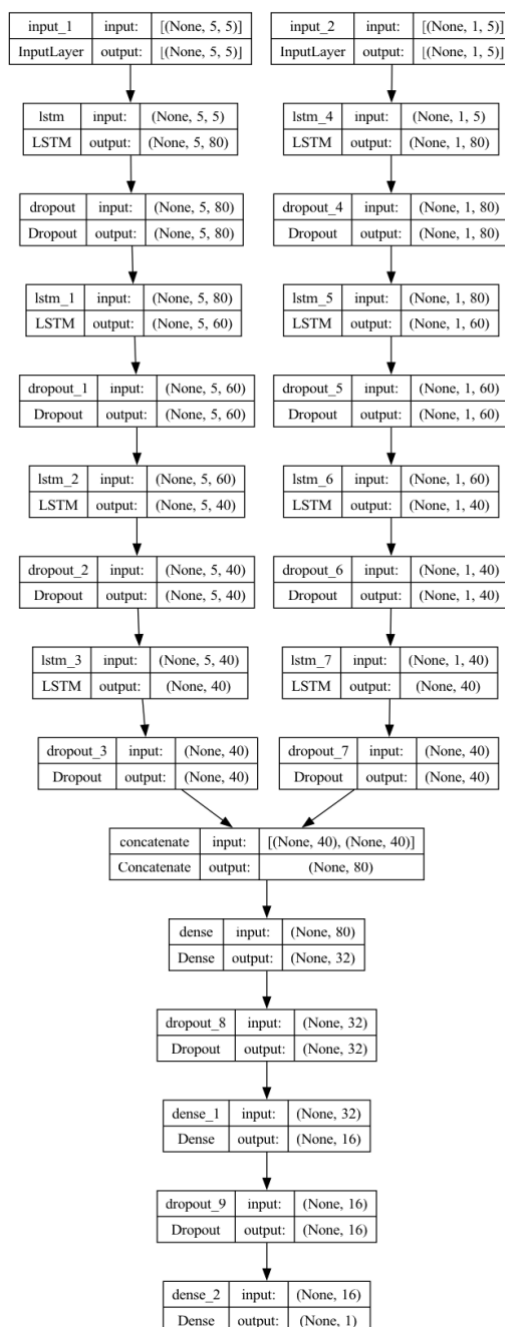


Рисунок 3.17 – Граф архітектури створеної моделі

Запропонована модель складається з двох основних компонентів:

- часові ряди LSTM: мережа з довгою короткостроковою пам'яттю (LSTM), призначена для аналізу та прогнозування даних фінансових часових рядів шляхом фіксації часових залежностей та закономірностей, притаманних ринковим індикаторам;
- NLP-Enhanced LSTM: ще одна LSTM-мережа, доповнена результатами BERT-обробки, спеціально розроблена для аналізу текстових даних та їхнього впливу на динаміку фінансового ринку.

Отримана модель далі поєднує дані компоненти і використовує повнозв'язні шари для прогнозування кінцевого значення виводу нейронної мережі.

Поєднуючи можливості аналізу часових рядів LSTM з інформацією, отриманою з оброблених BERT заголовків новин, запропонована модель може ефективно відображати складний взаємозв'язок між ринковими тенденціями, настроями та зовнішніми факторами (рис. 3.18).

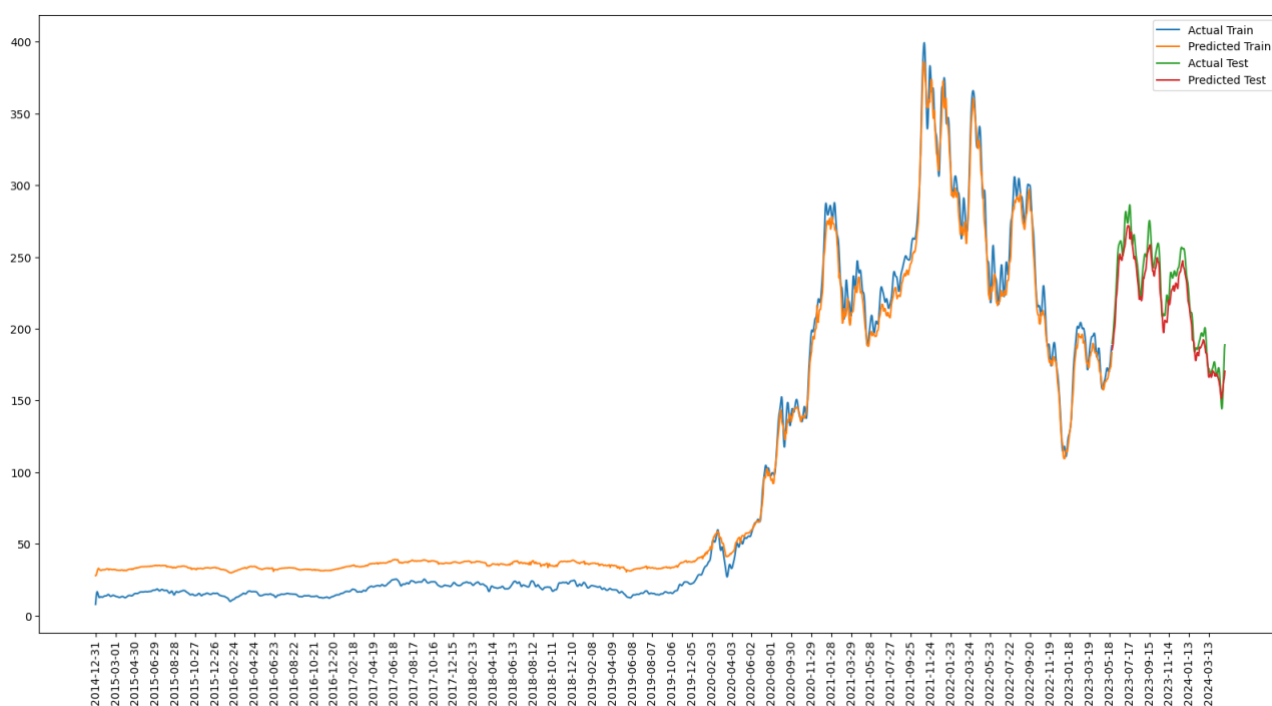


Рисунок 3.18 – Графік прогнозу із використанням запропонованої моделі

Таблиця 3.2 – Порівняння точності запропонованої моделі на тестовому сеті

Модель	RMSE	Accuracy	MAPE
WMA	16.9626031214389	83.94%	1510.47%
SARIMAX	51.96154917116397	76.91%	20.40%
MLP	25.14	89.59%	37.28%
RNN	19.26	91.44%	7.37%
LSTM	15.14	93.27%	6.07%
BERT-LSTM	9.64	95.72%	3.64%

Використана модель досить суттєво підвищує точність прогнозування моделі. Особливо зважаючи на можливість передбачення різких змін у цінах акцій через ті причини, які неможливо передбачити із самого часового ряду. При чому, незважаючи на досить великий проміжок передбачення, дані настроїв із заголовків новин дозволили передбачити із неймовірною точністю навіть значення цін акцій на 2024 рік.

Успіх запропонованої нами моделі підкреслює потенціал інтеграції методів NLP з традиційними методами прогнозування для підвищення ефективності прогнозування на фінансових ринках.

Майбутні напрямки досліджень можуть включати вивчення додаткових джерел текстових даних, вдосконалення архітектури з використанням передових методів глибокого навчання, а також розширення застосування моделей прогнозування з використанням NLP до ширшої сфери прогнозування показників фінансових ринків.

Таким чином, запропонована модель являє собою перспективний підхід до прогнозування показників фінансових ринків шляхом використання можливостей методів штучного інтелекту, зокрема інтеграції NLP з архітектурою LSTM. Використовуючи інсайти з текстових даних поряд з традиційним аналізом часових рядів, наша модель пропонує розширені

можливості прогнозування та цінну інформацію для прийняття рішень на фінансових ринках.

3.4 Аналіз отриманих результатів

Хоча основна увага приділяється прогнозуванню цін на акції, важливо визнати, що корисність запропонованої моделі виходить за рамки цього єдиного застосування. Дана модель може ефективно прогнозувати різні показники фінансового ринку, такі як валовий внутрішній продукт (ВВП), рівень інфляції та індекс ефективності, і як вона може потенційно перевершити традиційні методи прогнозування завдяки інтеграції даних на основі новин.

Адаптивність запропонованої моделі дозволяє їй прогнозувати широкий спектр індикаторів фінансових ринків, надаючи цінну інформацію про різні аспекти економічної діяльності та динаміки ринку.

Заголовки новин є багатим джерелом інформації, що відображає ринкові настрої, економічні тенденції, геополітичні події та інші фактори, які можуть впливати на показники фінансового ринку. Дані, отримані за допомогою запропонованої моделі, мають значний потенціал для подальшого застосування, зокрема, у сфері управління портфелями та алгоритмічної торгівлі.

Сучасна теорія портфеля, започаткована Гаррі Марковіцем, є широко використовуваною основою для побудови інвестиційних портфелів, метою яких є максимізація прибутковості при мінімізації ризику. Дані, отримані за допомогою моделі, які включають точні прогнози різних показників фінансового ринку, можуть оптимізувати фінансовий портфель інвестора.

Алгоритмічна торгівля, також відома як автоматизована торгівля або алготрейдинг, передбачає використання комп'ютерних алгоритмів для автоматичного виконання угод на основі заздалегідь визначених правил і критеріїв. Прогнози, згенеровані запропонованою моделлю, можуть бути легко

інтегровані в алгоритмічні торгові стратегії для прийняття торгових рішень та оптимізації торгових показників. Включаючи прогнози індикаторів фінансового ринку в реальному часі в торгові алгоритми, трейдери можуть використовувати неефективність ринку, отримувати вигоду з нових тенденцій і здійснювати угоди з більшою точністю та ефективністю.

Таким чином, дані, отримані за допомогою запропонованої моделі, виходять за рамки простого прогнозування показників фінансового ринку - вони можуть бути потужним інструментом для оптимізації інвестиційних стратегій, підвищення ефективності торгівлі та відкриття нових можливостей в управлінні портфелем та алгоритмічній торгівлі. Використовуючи інформацію, яку надає дана модель, інвестори та трейдери можуть приймати більш обґрунтовані рішення, зменшувати ризики та отримувати вигоду з нових тенденцій на сучасних динамічних та складних фінансових ринках.

Висновки до розділу 3

Проведено комплексний порівняльний аналіз, оцінивши ефективність моделей зваженого середнього (WMA), SARIMAX, багат шарового перцептрона (MLP), рекурентних нейронних мереж (RNN) та мереж з довгою та короткою пам'яттю (LSTM) у прогнозуванні динаміки фінансових ринків. Крім того, запропоновано нову модель, яка інтегрує методи обробки природної мови (NLP) на основі BERT з архітектурою LSTM для підвищення точності прогнозування показників фінансового ринку.

Аналіз включав різні метрики, такі як середня абсолютна похибка (MAE), середня квадратична похибка (MSE) та інші, що дають уявлення про сильні та слабкі сторони кожної моделі.

Запропонована модель використовує можливості BERT для обробки заголовків новин та вилучення цінної інформації, настроїв та контекстної

інформації, яка потім включається в прогнозування на основі LSTM для підвищення точності прогнозу. Модель BERT-NLP Enhanced LSTM демонструє багатообіцяючі результати, перевершуючи традиційні моделі і пропонуючи більш цілісний підхід до прогнозування фінансових ринків, використовуючи текстову інформацію.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

Функціонально-вартісний аналіз (ФВА) – це методологія, яка використовується для систематичної оцінки функціональності та цінності продукту. Вона пропонує структурований підхід до розуміння та оцінки здатності програмного продукту задовольняти потреби користувачів і надання цінності для бізнесу. ФВА має на меті проаналізувати ринок та оцінити доцільність розробки програмного продукту. Також буде оцінено витрати на розробку та оцінено економічну ефективність впровадження програмного продукту. Основна задача полягає у визначенні функцій, які є найбільш важливими і цінними як для користувачів, так і для бізнесу, а також оцінити витрати на реалізацію цих функцій і потенційні області для оптимізації та вдосконалення. ФВА може допомогти виявити недоліки у функціональності програмного продукту, які можуть знизити його цінність або викликати незадоволення користувачів. Виявивши ці проблеми, аналіз дає змогу зосередити зусилля на їх вирішенні та вдосконаленні.

У цьому розділі буде проведено оцінку основних функціональних можливостей та особливостей програмного продукту, призначеного для прогнозування фінансових показників підприємства з використанням часової ієрархії та нейронних мереж. Для цього будуть визначені ключові етапи його розробки та впровадження, а також необхідні фінансові та часові ресурси. Аналіз завершиться оцінкою кінцевої вартості програмного продукту.

4.1 Постановка задачі проектування

Метод функціонально-вартісного аналізу використано для виконання техніко-економічного аналізу створеного продукту для прогнозування показників фінансових ринків. Враховуючи той факт, що розробка та реалізація кожного компонента системи впливає на всю систему, вкрай важливо, щоб кожна підсистема відповідала своїм функціональним вимогам. Отже, в цьому дослідженні представлено детальний аналіз функціональних вимог до програмного продукту. Продукт призначений для збору, обробки та побудові прогнозу на основі введених даних обраного фінансового показника.

Технічні вимоги до програмного продукту включають наступні критерії:

- він повинен бути сумісним з персональними комп'ютерами, оснащеними стандартним набором компонентів;
- програмне забезпечення має бути зручним та інтуїтивно зрозумілим;
- воно має швидко обробляти дані та надавати доступ до інформації в режимі реального часу;
- продукт має бути легко масштабованим і підтримуваним;
- витрати на впровадження програмного забезпечення мають бути мінімальними.

Крім того, програмне забезпечення повинно включати надійні засоби для захисту конфіденційних даних і відповідати галузевим стандартам і нормам.

4.2 Обґрунтування функцій програмного продукту

Основною функцією F_0 є розробка програмного продукту, здатного генерувати надійні прогнози показників фінансових ринків з використанням

даних часових рядів. Ця функція є головною, надалі можна визначити інші функції:

- F_1 – вибір мови програмування;
- F_2 – вибір джерела для збору та підготовки даних;
- F_3 – вибір платформи з відкритим вихідним кодом для побудови та розгортання моделей машинного навчання.

Кожна функція має кілька варіантів імплементації. Функція F_1 : Python, Java. Функція F_2 : збір та обробка даних власноруч, збір та обробка даних спеціалізованою компанією. Функція F_3 : Tensorflow, PyTorch.

Різні способи виконання основних функцій зображені на морфологічній карті системи (рис. 4.1).

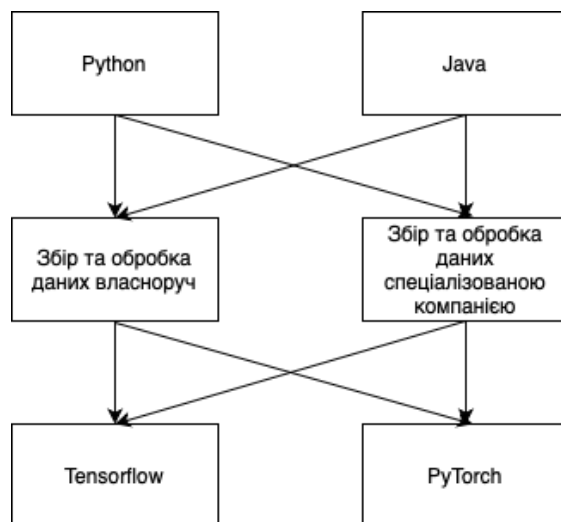


Рисунок 4.1 – Морфологічна карта

Морфологічна карта надає всебічний огляд потенційної варіативності основних функцій. Карта використовується для побудови позитивно-негативної матриці, яка класифікує зазначені варіанти функцій на ті, що є доцільними, і ті, що є недоцільними (таблиця 4.1).

Таблиця 4.1 – Тлумачення поняття «фінансовий ринок»

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	<i>A</i>	Швидкість написання продукту, великий вибір бібліотек, якісний набір функцій	Низька продуктивність
	<i>B</i>	Продуктивність коду, простота підтримки	Дуже багато часу необхідно витратити на написання продукту
F_2	<i>A</i>	Більший контроль над процедурами збору та обробки даних, підвищена безпека даних	Великі витрати часу, коштів через необхідність створення власної інфраструктури, потреба у експертах, проблеми масштабованості
	<i>B</i>	Доступ до експертних знань та досвіду, більша швидкість виконання, легша масштабованість	Менший контроль за методами збору та обробки даних, підвищений ризик витоку даних
F_3	<i>A</i>	Поширеність, можливості для розгортання, готовність до впровадження, швидкість виконання	Складність до вивчення, менша лаконічність, недостатня гнучкість
	<i>B</i>	Простота використання, простота налагодження, інтегрованість із популярними бібліотеками	Менш готовий до розгортання, недостатньо розгалужена екосистема, недостатні можливості оптимізації продуктивності

Результати аналізу свідчать про те, що певні варіанти реалізації програмних продуктів мають бути відхилені через їхню невідповідність поставленим цілям. Ці варіанти, які були визнані неприйнятними, позначені на морфологічній карті.

Функція F_1 : перевагу надаємо швидкості написання продукту та великому вибору необхідних бібліотек. Тому для пришидження написання програми варіант Б має бути відкинутий.

Функція F_2 : обидва представлені варіанти джерела для збору та обробки даних можна використовувати в розробці, оскільки кожен з них має свої плюси і недоліки. Причому недоліки не є критичними для даної задачі.

Функція F_3 : необхідно відкинути варіант Б, оскільки готовність для впровадження і можливості оптимізації продуктивності є більш важливими.

Отже, необхідно розглянути такі варіанти імплементації ПП:

$$F_{1A} - F_{2A} - F_{3A}$$

$$F_{1A} - F_{2B} - F_{3A}$$

Для того, щоб оцінити якість обраних функцій, було обрано набір параметрів. Далі ці параметри описані більш детально.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, були визначені основні параметри відбору для розрахунку коефіцієнта технічного рівня. Для характеристики програмного продукту будемо використовувати наступні параметри:

- X1 – швидкодія мови програмування;
- X2 – утилізація пам'яті для обчислень та зберігання даних;
- X3 – час навчання;
- X4 – потенційний розмір програмного коду;
- X5 – легкість інтеграції з іншими системами.

Найгірше, середнє та найкраще значення цих параметрів обирається відповідно до вимог замовника та умов, що визначають роботу програмного продукту, як показано в таблиці 4.2.

Таблиця 4.2 – Головні параметри ПП

Назва	Позначення	Одиниці вимірювання	Значення		
			гірші	середні	кращі
Швидкодія мови	X1	оп/мс	9500	16200	22600
Утилізація пам'яті	X2	Мб	2048	1024	512
Час навчання	X3	с	240	200	150
Потенційний розмір коду	X4	к-сть рядків	2300	1600	1200
Легкість інтеграції	X5	год	100	50	10

За отриманими даними з таблиці 4.2 можна побудувати графічні характеристики параметрів на рисунках 4.2 – 4.6.

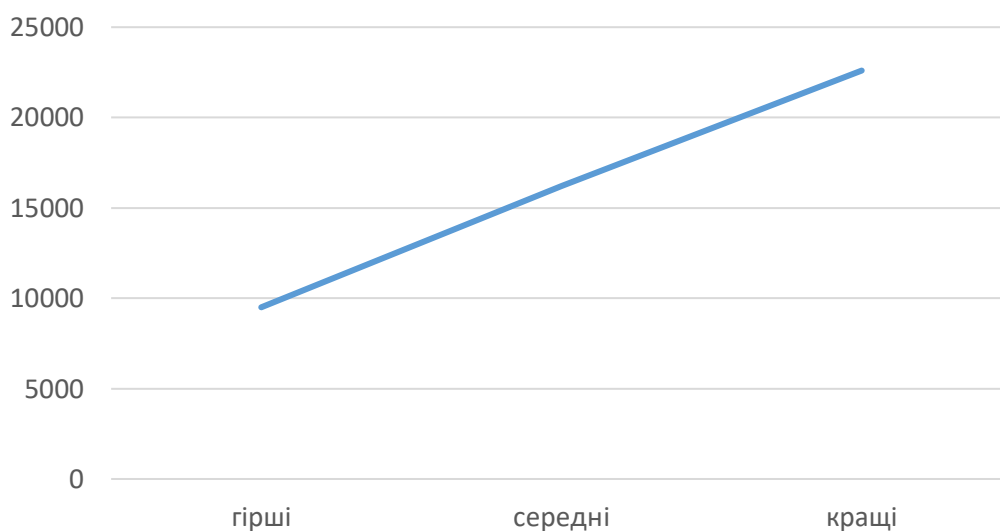


Рисунок 4.2 – X1, швидкодія мови, оп/мс

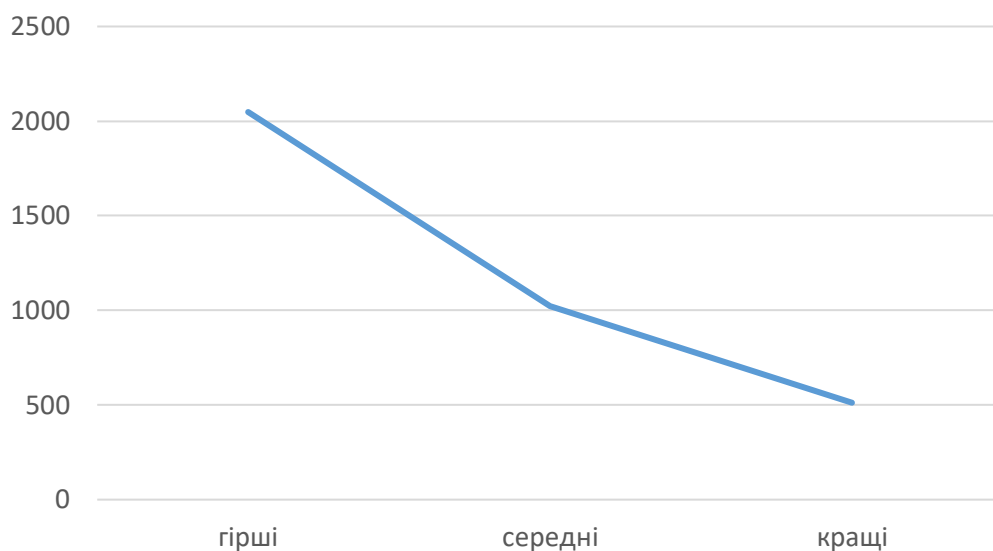


Рисунок 4.3 – X2, утилізація пам'яті, Мб

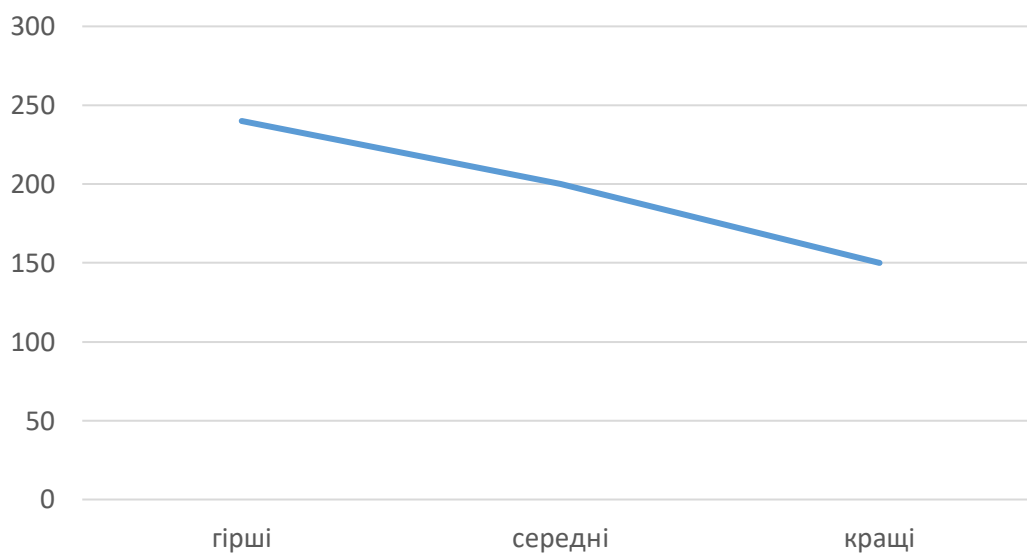


Рисунок 4.4 – X3, час навчання, с

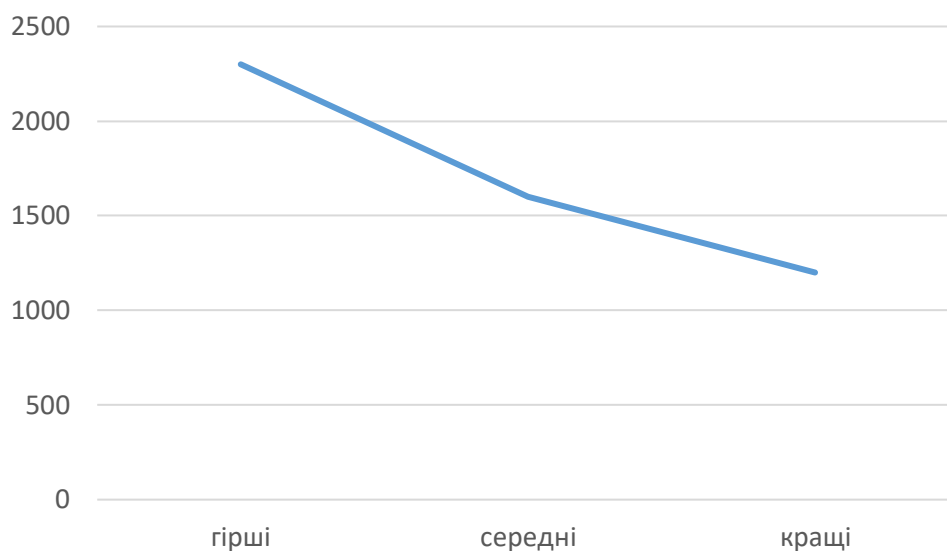


Рисунок 4.5 – X4, потенційний розмір коду, кількість рядків

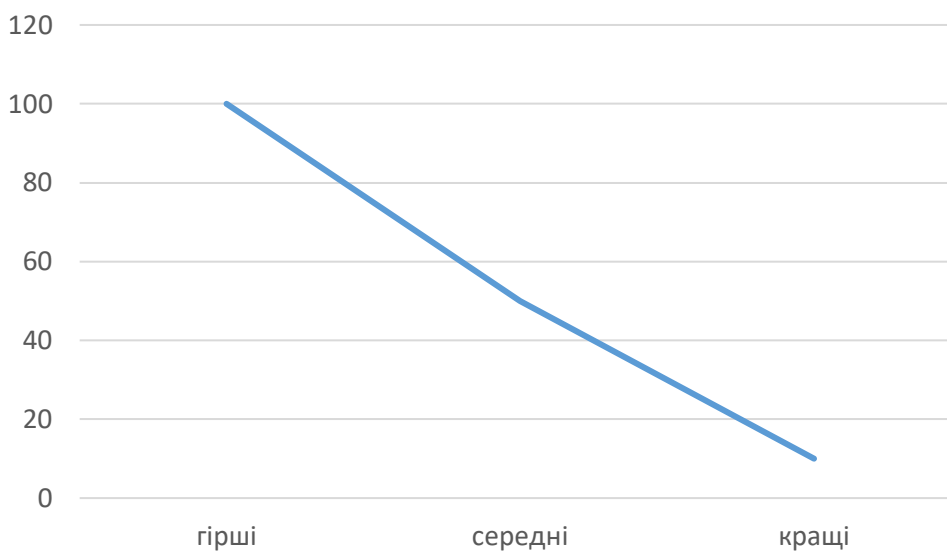


Рисунок 4.6 – X5, легкість інтеграції, год

4.4 Аналіз експертного оцінювання параметрів

Після всебічного вивчення та оцінки кожен експерт визначає важливість конкретних змінних для розробки адаптивної моделі прогнозування, яка може

точно ідентифікувати ці змінні та обчислити прогностні значення.

Значимість окремих параметрів оцінювалася комісією з п'яти експертів за допомогою методу попарних порівнянь. Процедура визначення коефіцієнтів значущості включає наступні кроки:

- присвоєння різних рангів для встановлення рівня значущості для кожного параметра;
- перевірка узгодженості та придатності суджень експертів для подальшого аналізу;
- визначення попарної пріоритетності параметрів;
- обробка результатів для розрахунку коефіцієнта значущості.

Результати експертного ранжування представлені в таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Назва	Позначення	Одиниці виміру	Ранг параметра за судженням експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
Швидкодія мови	X1	оп/мс	2	5	4	3	3	5	3	25	4	16
Утилізація пам'яті	X2	Мб	4	3	3	5	4	3	4	26	5	25
Час навчання	X3	с	5	4	5	4	5	4	5	32	11	121
Потенційний розмір коду	X4	к-сть рядків	1	2	1	2	1	2	1	10	-11	121
Легкість інтеграції	X5	год	3	1	2	1	2	1	2	12	-9	81
	Разом		15	15	15	15	15	15	15	105	0	364

Для того, щоб з'ясувати достовірність експертних оцінок, будуть встановлені наступні параметри:

- сума рангів кожного з параметрів і загальна сума рангів у формулі (4.1):

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 105 \quad (4.1)$$

де N – кількість експертів;

n – кількість параметрів.

- середня сума рангів у формулі (4.2):

$$T = \frac{1}{n} R_{ij} = 21. \quad (4.2)$$

- розбіжність між загальною сумою рангів кожної змінної та середньою загальною сумою рангів у формулі (4.3):

$$\Delta_i = R_i - T \quad (4.3)$$

Сума відхилень для всіх параметрів повинна дорівнювати нулю.

- загальна сума квадратичних відхилень у формулі (4.4):

$$S = \sum_{i=1}^N \Delta_i^2 = 364 \quad (4.4)$$

Обчислимо коефіцієнт узгодженості згідно з формулою (4.5):

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 364}{7^2(5^3-5)} = 0.742857 > W_k = 0,67 \quad (4.5)$$

Ранжування вважається достовірним, оскільки коефіцієнт узгодженості перевищує нормативне значення 0,67.

На основі результатів ранжування проведемо попарне порівняння всіх параметрів, результати якого відобразимо в таблиці 4.4.

Таблиця 4.4 – Результати ранжування параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 та X2	<	>	>	<	<	>	<	<	0.5
X1 та X3	<	>	<	<	<	>	<	<	0.5
X1 та X4	>	>	<	>	>	>	>	>	1.5
X1 та X5	<	>	<	>	>	>	>	>	1.5
X2 та X3	<	<	<	>	<	<	<	<	0.5
X2 та X4	>	>	>	>	>	>	>	>	1.5
X2 та X5	>	>	>	>	>	>	>	>	1.5
X3 та X4	>	>	>	>	>	>	>	>	1.5
X3 та X5	>	>	>	>	>	>	>	>	1.5
X4 та X5	<	>	<	>	<	>	<	<	0.5

Числове значення, яке відображає відносну перевагу i -го параметра порівняно з j -м параметром, визначається за допомогою наступної формули:

$$a_{ij} = \begin{cases} 1.5 \text{ якщо } X_i > X_j \\ 1.0 \text{ якщо } X_i = X_j \\ 0.5 \text{ якщо } X_i < X_j \end{cases} \quad (4.6)$$

Використовуючи отримані числові оцінки переваг, побудуємо матрицю $A = \|a_{ij}\|$.

Для кожного параметра обчислимо вагомість K_{bi} , використовуючи наступні формули:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.7)$$

$$b_i = \sum_{i=1}^N a_{ij} \quad (4.8)$$

Відносні оцінки перераховуються кілька разів, доки не буде досягнуто збіжності, тобто наступні значення відрізняються від попередніх не більше, ніж на 2%. На наступних ітераціях відносні оцінки розраховуються за вказаними формулами (4.9) та (4.10).

$$K_{\text{Ві}} = \frac{b'_i}{\sum_{i=1}^n b'_i} \quad (4.9)$$

$$b'_i = \sum_{i=1}^N a_{ij} b_j \quad (4.10)$$

Як показано в таблиці 4.5, розбіжність у вагових коефіцієнтах становить менше 2%, що свідчить про відсутність необхідності в подальших ітераціях.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j					Перша ітерація		Друга ітерація	
	X1	X2	X3	X4	X5	b_i	$K_{\text{Ві}}$	b_i^1	$K_{\text{Ві}}^1$
X1	1,0	0,5	0,5	1,5	1,5	5	0,2	22	0,1913
X2	1,5	1,0	0,5	1,5	1,5	6	0,24	27,5	0,239
X3	1,5	1,5	1,0	1,5	1,5	7	0,28	34	0,2956
X4	0,5	0,5	0,5	1,0	0,5	3	0,12	14	0,1217
X5	0,5	0,5	0,5	1,5	1,0	4	0,16	17,5	0,15217
Разом						25	1	115	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Якість кожного варіанту основних функцій оцінюється індивідуально. Абсолютні значення показників узгоджені з технічними характеристиками програмного продукту.

Коефіцієнт технічного рівня для кожного варіанту реалізації програмного забезпечення розраховується відповідно до методики, викладеної в таблиці 4.6, за формулою (4.11).

$$K_K(j) = \sum_{i=1}^n K_{vi,j} B_{i,j}, \quad (4.11)$$

де n – кількість параметрів;

K_{vi} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Головні функції	Варіант реалізації	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	19400	4	0.1913	0.7652
F2	A	X2	1024	3	0.239	0.717
	B	X2	512	5	0.239	1.195
F3	A	X3	170	4	0.2956	1.1824

Для того, щоб визначити рівень якості кожного з варіантів, необхідно використати дані з таблиці 4.6 відповідно до формули (4.12):

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}], \quad (4.12)$$

$$K_{K1} = 0.7652 + 0.717 + 1.1824 = 2.6646$$

$$K_{K2} = 0.7652 + 1.195 + 1.1824 = 3.1426$$

Розрахунки показують, що другий варіант є більш оптимальним, бо значення, розраховане за коефіцієнтом технічного рівня, вище.

4.6 Економічний аналіз варіантів розробки програмного продукту

Для того, щоб з'ясувати фінансові витрати на розробку програмного додатку, спочатку необхідно визначити його трудомісткість. Усі варіанти охоплюють два окремих завдання: проектування програмного продукту та розробку програмного інтерфейсу.

Завдання 1 віднесено до групи А через його новизну, тоді як завдання 2 віднесено до групи В. Що стосується складності, то алгоритми для завдання 1 віднесено до групи 1, тоді як алгоритми для завдання 2 віднесено до групи 3.

Завдання 1 використовує довідкову інформацію, тоді як завдання 2 працює з даними. Норми часу на розробку та програмування будуть розраховані для кожного завдання.

Загальна трудомісткість буде визначена за наступною формулою:

$$T_0 = T_p \times K_{\Pi} \times K_M \times K_{CT} \times K_{CT.M}, \quad (4.13)$$

де T_p – трудомісткість розробки програмного продукту;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт складності вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

K_{CT} – коефіцієнт використання стандартних модулів та прикладних програм;

$K_{CT.M}$ – коефіцієнт стандартного математичного забезпечення.

На початковому етапі норми часу, пов'язані із завданнями розрахункового характеру, що мають ступінь новизни А та рівень складності 1, оцінюються як

такі, що потребують трудомісткості $T_p = 33$ людино-днів. Крім того, $K_{\Pi} = 1,75$ є поправочним коефіцієнтом, що враховує специфічний тип нормативно-довідкової інформації, необхідної для виконання першого завдання. Крім того, $K_{СК} = 1$ є поправочним коефіцієнтом, що враховує складність механізмів контролю, необхідних для вхідних і вихідних інформаційних потоків, пов'язаних з усіма сімома завданнями в сукупності. Оскільки при розробці початкової задачі використовуються стандартні модулі, буде застосовано коефіцієнт $K_{СТ} = 0,85$. Отже, загальна складність програмування для початкової задачі становить:

$$T_1 = 33 \times 1.75 \times 0.85 = 49.0875 \text{ людино} - \text{днів}$$

Тепер необхідно виконати аналогічні розрахунки для наступних задач.

Для другої задачі, яка використовує алгоритм третьої групи складності та ступеня новизни Б, значення є наступними: Загальний необхідний час $T_p = 21$ людино-днів, поправочний коефіцієнт $K_{\Pi} = 0,95$, коефіцієнт складності вхідної інформації $K_{СК} = 1$, коефіцієнт використання стандартних модулів та прикладних програм $K_{СТ} = 0,85$.

$$T_2 = 21 \times 0.95 \times 0.85 = 16.9575 \text{ людино} - \text{днів}$$

Загальна трудомісткість кожного з обраних варіантів реалізації програми визначається шляхом підсумовування трудомісткості відповідних завдань:

$$T_I = (49.0875 + 16.9575 + 4.93 + 16.9575) \times 8 = 703.46 \text{ людино} - \text{годин}$$

$$T_{II} = (49.0875 + 16.9575 + 7.34 + 16.9575) \times 8 = 722.74 \text{ людино} - \text{годин}$$

Варіант II є найбільш трудомістким.

Процес розробки здійснюється двома програмістами із заробітною платою 25300 грн. та одним аналітиком даних із заробітною платою 20600 грн. Для того,

щоб визначити середню заробітну плату за годину, необхідно використати наступну формулу:

$$C_{\text{ч}} = \frac{M}{T_m * t}, \quad (4.14)$$

де M – щомісячна винагорода працівникам;

T_m – кількість робочих днів на тиждень;

t – загальна кількість годин на день, проведених на роботі.

$$C_{\text{ч}} = \frac{25300 + 25300 + 20600}{3 * 21 * 8} \approx 141.27$$

В подальшому заробітна плата розраховується відповідно до формули:

$$C_{\text{ЗП}} = C_{\text{ч}} * T_i * K_{\text{д}}, \quad (4.15)$$

де $C_{\text{ч}}$ – винагорода для програміста на погодинній основі;

T_i – ступінь навантаження, необхідний для виконання поставленого завдання;

$K_{\text{д}}$ – стандарт, який передбачає додаткову винагороду.

Відповідно до зазначених параметрів, винагорода розробників виглядає наступним чином:

$$C_{\text{ЗП1}} = 141.27 * 703.46 * 1.2 \approx 119253.35 \text{ грн}$$

$$C_{\text{ЗП2}} = 141.27 * 722.74 * 1.2 \approx 122521.78 \text{ грн}$$

Частина на єдиний соціальний внесок становить 22%:

$$C_{\text{ВД1}} = 119253.35 * 0.22 = 26235.74 \text{ грн}$$

$$C_{\text{ВІД2}} = 122521.78 * 0.22 = 26954.79 \text{ грн}$$

Наступним кроком є визначення вартості оплати однієї машино-години (C_M).

Враховуючи, що один комп'ютер обслуговує один програміст із заробітною платою 25300 грн та коефіцієнтом зайнятості 0,2, вартість однієї машини розраховується наступним чином:

$$C_G = 12 * M * K_3 = 12 * 25300 * 0.2 = 60720 \text{ грн}$$

Включаючи додаткову заробітну плату:

$$C_{3П} = C_G * (1 + K_3) = 60720 * (1 + 0.2) = 72864 \text{ грн}$$

Частина на єдиний соціальний внесок:

$$C_{\text{ВІД}} = C_{3П} * 0.22 = 72864 * 0.22 = 16030.08 \text{ грн}$$

Амортизаційні відрахування розраховуються при нормі амортизації 25% та вартості комп'ютера 32000 грн.

$$C_A = K_{\text{ТМ}} * K_A * C_{\text{ПР}} = 1.27 * 0.25 * 32000 = 10160 \text{ грн}$$

де $K_{\text{ТМ}}$ – коефіцієнт, що включає затрати на перевезення та облаштування пристрою у користувача;

K_A – річна амортизаційна ставка;

$C_{\text{ПР}}$ – контрактна ціна пристрою.

Розрахунок витрат на ремонт та обслуговування виглядає наступним чином:

$$C_P = K_{TM} \cdot C_{ПР} \cdot K_P = 1.27 * 32000 * 0.05 = 2032 \text{ грн}$$

де K_P – частка затрат на проведення поточних ремонтів.

За наступною формулою розраховується ефективний годинний фонд часу роботи ПК за рік:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) * t * K_B = (365 - 104 - 12 - 16) * 8 * 0.83 = 1547.12 \text{ год}$$

де D_K – число календарних днів у році;

D_B, D_C – числа вихідних і святкових днів відповідно;

D_P – число днів планових ремонтів обладнання;

t – число робочих годин на добу;

K_B – коефіцієнт використання приладу протягом зміни.

Вартість електроенергії розраховується за наступною формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} * N_C * K_3 * C_{\text{ЕН}} = 1547.12 * 0.7 * 0.5 * 5.23 = 2832 \text{ грн}$$

де N_C – середнє енергоспоживання приладу;

K_3 – коефіцієнт заповнення приладу;

$C_{\text{ЕН}}$ – вартість 1 кВт-год електроенергії.

Накладні витрати розраховуються за формулою:

$$C_H = C_{\text{ПР}} * 0.67 = 32000 * 0.67 = 21440 \text{ грн}$$

Річні операційні витрати визначаються за наступною формулою:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H, \quad (4.16)$$

$$C_{\text{ЕКС}} = 72864 + 16030.08 + 10160 + 2032 + 2832 + 21440 = 125358.08 \text{ грн}$$

Собівартість однієї машино-години дорівнює:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 125358.08 / 1547.12 = 81.03 \text{ грн./год.}$$

У контексті розробки програмних продуктів усі роботи, пов'язані з вищезазначеним, виконуються на комп'ютерах. Отже, витрати на оплату машинного часу залежать від обраного варіанту і розраховуються за наступною формулою:

$$C_{\text{М}} = C_{\text{М-Г}} * T, \quad (4.17)$$

$$C_{\text{М1}} = 81.03 * 703.46 = 57001.36 \text{ грн}$$

$$C_{\text{М2}} = 81.03 * 722.74 = 58563.62 \text{ грн}$$

Накладні витрати становлять 67% від заробітної плати, які визначаються за такою формулою:

$$C_{\text{Н}} = C_{\text{ЗП}} * 0.67, \quad (4.18)$$

$$C_{\text{Н1}} = 119253.35 * 0.67 = 79899.74 \text{ грн}$$

$$C_{\text{Н2}} = 122521.78 * 0.67 = 82089.59 \text{ грн}$$

Таким чином, вартість розробки ПП становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}}, \quad (4.19)$$

$$C_{\text{ПП1}} = 119253.35 + 26235.74 + 57001.36 + 79899.74 = 282390.19 \text{ грн}$$

$$C_{\text{ПП2}} = 122521.78 + 26954.79 + 58563.62 + 82089.59 = 290129.78 \text{ грн}$$

4.7 Вибір кращого варіанту програмного продукту техніко-економічного рівня

Коефіцієнт техніко-економічного рівня можна розрахувати за такою формулою:

$$K_{K1} = 0.7652 + 0.717 + 1.1824 = 2.6646$$

$$K_{K2} = 0.7652 + 1.195 + 1.1824 = 3.1426$$

$$K_{\text{ТЕР}j} = K_{Kj} / C_{\Phi j}, \quad (4.20)$$

$$K_{\text{ТЕР}1} = \frac{2.6646}{282390.19} = 9.436 * 10^{-6}$$

$$K_{\text{ТЕР}2} = \frac{3.1426}{290129.78} = 10.832 * 10^{-6}$$

Як бачимо, другий варіант реалізації програми з коефіцієнтом техніко-економічного $K_{\text{ТЕР}2} = 10.832 \times 10^{-6}$ рівня є найбільш ефективним.

За результатами функціонально-вартісного аналізу розробленого програмного комплексу можна зробити висновок, що з решти альтернатив оптимальним є другий варіант реалізації програмного продукту. Це було визначено після первинного відбору двох варіантів програмного комплексу. Він має оптимальний показник рівня техніко-економічної якості $K_{\text{ТЕР}} = 10.832 * 10^{-6}$.

Обраний варіант програмного продукту характеризується наступними параметрами:

- мова програмування – Python;
- джерело для збору та підготовки даних – збір та обробка даних спеціалізованою компанією;

- платформа з відкритим вихідним кодом для побудови та розгортання моделей машинного навчання – Tensorflow.

Висновки до розділу 4

Проведено функціонально-вартісний аналіз розробленого програмного продукту, під час якого визначено його основні характеристики та оцінено основні функції. На основі цього аналізу було обрано найбільш оптимальний варіант реалізації. Другий варіант реалізації, який використовує мову програмування Python, платформу Tensorflow та відбір даних спеціалізованою компанією, було обрано на основі комплексної оцінки його придатності, якості та економічної ефективності. Коефіцієнт техніко-економічного рівня був розрахований на рівні 10.832×10^{-6} .

Таким чином, даний варіант впровадження є найкращим з точки зору показників якості та економічної оцінки. Результати проведених досліджень свідчать про те, що даний варіант відповідає всім необхідним вимогам і володіє необхідними характеристиками для успішного виконання поставленого завдання.

ВИСНОВКИ

У дипломній роботі було проведено дослідження потенціалу моделей на основі штучного інтелекту у підвищенні точності та можливостей прогнозування індикаторів фінансового ринку. Вирішення поставлених завдань дозволило отримати наступні висновки.

В результаті дослідження наукових джерел було визначено фундаментальні принципи, що визначають динаміку фінансових ринків. З'ясовано ролі різних учасників ринку та окреслено ключові характеристики фінансових ринків. Крім того, наведено причини важливості прогнозування показників фінансового ринку за допомогою систем штучного інтелекту, враховуючи важливість цих показників у прийнятті економічних рішень та управлінні ризиками.

У другому розділі проведено ретельний аналіз теорії функціонування існуючих моделей прогнозування. Проведено ретельний аналіз методологій, алгоритмів та показників для оцінки результативності моделей з метою виявлення їх сильних та слабких сторін, а також можливостей для вдосконалення. В результаті виконаної роботи було систематизовано основні переваги та недоліки від застосування кожного з методів.

У третьому розділі виконано практичну реалізацію запропонованої моделі, поєднавши теоретичні висновки з практичним досвідом застосування моделей прогнозування. Проведено порівняльний аналіз існуючих моделей прогнозування, включаючи WMA, SARIMAX, MLP, RNN та LSTM. Крім того, запропоновано нову модель, яка інтегрує методи обробки природньої мови із використанням передтренуваної моделі BERT з архітектурою LSTM для підвищення точності прогнозування. Запропонована модель BERT-NLP Enhanced LSTM показала гарні результати, перевершивши традиційні моделі завдяки використанню текстової інформації поряд із найбільш передовою архітектурою блока нейронної мережі для прогнозування послідовних даних.

У даній роботі проаналізовано існуючі методи та запропоновано приклад інноваційної моделі, зроблено практичне порівняння моделей. Таким чином, закладено основу для більш точного та надійного прогнозування показників фінансового ринку. Запропонована модель має потенціал принесення користі інвесторам, фінансовим установам, політикам та економіці в цілому. За оцінками функціонально-вартісного аналізу визначено економічну доцільність впровадження створеного програмного продукту та витрати, які необхідні для впровадження та підтримання роботи додатку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Еш С. М. Фінансовий ринок: навч. посіб. К.: Центр учбової літератури, 2009. 528 с.
2. Hayes, A. 2023. Financial Markets: Role in the Economy, Importance, Types, and Examples. Investopedia. URL: <https://www.investopedia.com/terms/f/financial-market.asp> (дата звернення 07.05.2024)
3. Hussain, A. 2023. What Does an Investor Do? What Are the Different Types?. Investopedia. URL: <https://www.investopedia.com/terms/i/investor.asp> (дата звернення 07.05.2024)
4. Hayes A. 2024. What is a Financial Institution? Investopedia. URL: <https://www.investopedia.com/terms/f/financialinstitution.asp> (дата звернення 07.05.2024)
5. Schmidt Michael, 2021. Financial Regulators: Who They Are and What They Do. Investopedia. URL: <https://www.investopedia.com/articles/economics/09/financial-regulatory-body.asp> (дата звернення 07.05.2024)
6. Chen James. 2020. Financial Intermediary: What It Means, How It Works, Examples. URL: <https://www.investopedia.com/terms/f/financialintermediary.asp> (дата звернення 07.05.2024)
7. Закон України «Про інвестиційну діяльність». URL: https://zakon.rada.gov.ua/laws/show/1560-12?find=1&text=%D1%96%D0%BD%D0%B2%D0%B5%D1%81%D1%82%D0%BE%D1%80#w1_1 (дата звернення 07.05.2024)
8. NYSE: The New York Stock Exchange. URL: <https://www.nyse.com/index> (дата звернення 07.05.2024)

9. Nasdaq: Stock Market, Data Updates, Reports & News. URL: <https://www.nasdaq.com/> (дата звернення 07.05.2024)
10. London Stock Exchange. URL: <https://www.londonstockexchange.com/>
11. S&P 500. S&P Dow Jones Indices. URL: <https://www.spglobal.com/spdji/en/indices/equity/sp-500/> (дата звернення 07.05.2024)
12. Ткаченко О., Шендецька О. Якими бувають ETF фондового ринку. URL: <https://tradersunion.com/ua/interesting-articles/types-of-etfs/> (дата звернення 07.05.2024)
13. Кльоба Л.Г. Ефективний ризик-менеджмент – запорука фінансової безпеки банку. Ефективна економіка. 2017. № 6. URL: <http://www.economy.nayka.com.ua/?op=1&z=5641> (дата звернення 07.05.2024)
14. Діяльність з управління активами інституційних інвесторів (запитання та відповіді): навчальний посібник/ Під ред. Зверькова М.І. - Одеса: 2009. - 318 с.
15. Химич І.Г. Опорний конспект лекцій з дисципліни «Фінансовий ринок». Тернопіль: ТНТУ ім. І. Пулюя, 2018. – 171 с.
16. Буртняк І.В. Модель визначення волатильності фондового ринку. Вісник Хмельницького національного університету. 2022. № 1. С. 310-320
17. Klekovkin V. Time series filtering algorithms: a brief overview. URL: <https://medium.com/deelvin-machine-learning/time-series-filtering-algorithms-a-brief-overview-af2d3112cd03> (дата звернення 07.05.2024)
18. Erdem K.. Finding correlations in time series data. <https://erdem.pl/2020/06/finding-correlations-in-time-series-data> (дата звернення 07.05.2024)
19. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, C. 54(1-3), 159–178.

20. Kumar R., Kumar P., Kumar Y. (2021). Analysis of financial time series forecasting using deep learning model. Paper presented at 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Uttar Pradesh, India, January 28–29; pp. 877–81.
21. Regunath G., Lantos D. 10 Incredibly useful Time Series forecasting algorithms. URL: <https://www.advancinganalytics.co.uk/blog/2021/06/22/10-incredibly-useful-time-series-forecasting-algorithms> (дата звернення 07.05.2024)
22. Kaufman P. J. 2019. "Trading Systems and Methods," John Wiley & Sons.
23. Chavan A. Forecasting the Future: A Comprehensive Guide to Moving Averages in Time Series Analysis. URL: https://medium.com/@anuj_chavan/forecasting-the-future-a-comprehensive-guide-to-moving-averages-in-time-series-analysis-880fa17e877a (дата звернення 07.05.2024)
24. Zhang M. Time Series: Autoregressive models AR, MA, ARMA, ARIMA. URL: <https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class16.pdf> (дата звернення 07.05.2024)
25. Jaiswal S. Multilayer Perceptrons in Machine Learning: A Comprehensive Guide. URL: <https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning> (дата звернення 07.05.2024)
26. An S. Introduction to how an Multilayer Perceptron works but without complicated math. URL: <https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-but-without-complicated-math-a423979897ac> (дата звернення 07.05.2024)
27. Saeed M. An Introduction to Recurrent Neural Networks and the Math That Powers Them. URL: <https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/>(дата звернення 07.05.2024)

28. Poudel S. Recurrent Neural Network (RNN) Architecture Explained. URL: <https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef> (дата звернення 07.05.2024)
29. Nabi J. Recurrent Neural Networks (RNNs). URL: <https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85>
30. Calzone O. An Intuitive Explanation of LSTM. URL: <https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c> (дата звернення 07.05.2024)
31. Everitt, B. S.; Skrondal, A. (2010), The Cambridge Dictionary of Statistics, Cambridge University Press.
32. Bobbitt Z. How to Interpret Root Mean Square Error (RMSE). URL: <https://www.statology.org/how-to-interpret-rmse/> (дата звернення 07.05.2024)
33. Oracle. MAPE (Mean Absolute Percentage Error). URL: https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html (дата звернення 07.05.2024)
34. Офіційний сайт Yahoo Finance. URL: <https://finance.yahoo.com> (дата звернення 11.05.2024)
35. Krishnakumar M. An Introduction to BERT And How To Use It. URL: https://wandb.ai/mukilan/BERT_Sentiment_Analysis/reports/An-Introduction-to-BERT-And-How-To-Use-It--VmlldzoyNTIyOTA1 (дата звернення 11.05.2024)

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import json
import tkinter as tk
from datetime import datetime, timedelta
from tkinter import filedialog
from tkinter import messagebox

import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import tensorflow as tf
import yfinance as yf
from matplotlib.backends.backend_tkagg import
FigureCanvasTkAgg
from matplotlib.figure import Figure
from requests_html import HTMLSession, PyQuery as pq
from scipy.signal import butter, lfilter
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.layers import Dropout
from transformers import AutoTokenizer,
AutoModelForSequenceClassification

matplotlib.use('TkAgg')

KEYWORDS = ["business", "markets", "stocks", "wealth",
"economy", "finance", "money", "investing",
```

```
        "entrepreneurship", "treasurys", "tech", "foreign  
exchange", "tax report",  
        "the numbers", "u.s.", "commodities", "funds",  
"etfs", "futures", "currencies", ]  
source_url = 'https://www.wsj.com'
```

```
# Functions
```

```
def format_date(date):
```

```
    year = str(date.year)
```

```
    month = str(date.month).zfill(2)
```

```
    day = str(date.day).zfill(2)
```

```
    return year, month, day
```

```
def get_articles(session, year, month, day, page=1):
```

```
    url =
```

```
f'{source_url}/news/archive/{year}/{month}/{day}?page={page}'
```

```
    print(url)
```

```
    r = session.get(url)
```

```
    return r.html.find('article')
```

```
def get_num_pages(r):
```

```
    pages_number =
```

```
r.html.xpath('//*[@id="main"]/div[2]/div[1]/div/div/span')
```

```
    return int(str.split(pages_number[0].text, ' ')[-1]) if
```

```
pages_number else 1
```

```

def scrape_news(session, date):
    year, month, day = format_date(date)
    articles_pages = [get_articles(session, year, month, day)]
    num_pages =
get_num_pages(session.get(f'{source_url}/news/archive/{year}/{
month}/{day}'))
    print(num_pages)
    for page in range(2, num_pages + 1):
        articles = get_articles(session, year, month, day,
page)
        if articles:
            articles_pages.append(articles)
        else:
            break
    return articles_pages

def process_articles(articles_pages, date):
    allData = {str(date.date()): []}
    for articles in articles_pages:
        for article in articles:
            t = pq(article.html)
            headingText = t('div:nth-child(2) > div:nth-
child(2) > div > h2 > a > span').text()
            spanId = t('div:nth-child(2) > div:nth-child(1) >
div > span > div').text()
            print(f'{headingText} - {spanId}')

```

```

        label = spanId.lower() if spanId is not None else
None

        label_words = label.split()
        if len(headingText) > 0 and any(word in KEYWORDS
for word in label_words):
            allData[str(date.date())].append({
                "heading": headingText,
                "label": label,
            })
    return allData

def get_news_data(dateStart=datetime(2015, 1, 1),
dateEnd=datetime.now(), deltaTime=timedelta(days=1)):
    try:
        with open('./headlines.json', 'r') as f:
            data = json.load(f)
        if data:
            start = datetime.strptime(list(data.keys())[0],
'%Y-%m-%d')
            end = datetime.strptime(list(data.keys())[-1],
'%Y-%m-%d')
            secondDate =
datetime.strptime(list(data.keys())[1], '%Y-%m-%d')
            delta = secondDate - start
            if start == dateStart and end == dateEnd and delta
== deltaTime:
                messagebox.showinfo("Success", "News data has
already been scraped and saved to headlines.json.")

```

```

        return data, True
    except FileNotFoundError:
        pass

    session = HTMLSession()

    allData = {}
    for date in np.arange(dateStart, dateEnd,
deltaTime).astype(datetime):
        articles_pages = scrape_news(session, date)
        allData.update(process_articles(articles_pages, date))

    with open('./headlines.json', 'w', encoding='utf-8') as f:
        json.dump(allData, f, ensure_ascii=False)
        messagebox.showinfo("Success", "News data has been
successfully scraped and saved to headlines.json.")
    return allData, False

def fill_missing_with_mean(df, missing_col):
    for col in missing_col:
        for i in df.index:
            if pd.isnull(df.loc[i, col]):
                non_null_prev =
df[col].iloc[:i][df[col].iloc[:i].notna()].last_valid_index()

                non_null_next =
df[col].iloc[i:][df[col].iloc[i:].notna()].first_valid_index()

```

```
        if non_null_prev is not None and non_null_next
is not None:
            df.loc[i, col] =
np.mean([df.loc[non_null_prev, col], df.loc[non_null_next,
col]])

            elif non_null_prev is not None:
                df.loc[i, col] = df.loc[non_null_prev,
col]

            elif non_null_next is not None:
                df.loc[i, col] = df.loc[non_null_next,
col]
    return df

def butter_lowpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = butter(order, normal_cutoff, btype='low',
analog=False)
    return b, a

def butter_lowpass_filter(data, cutoff, fs, order=5):
    b, a = butter_lowpass(cutoff, fs, order=order)
    y = lfilter(b, a, data)
    return y
```

```
def filter_data(df, cutoff=2.667, fs=30.0, order=6):
    df2_lowpass = df.copy()
    for col in df2_lowpass.columns:
        if col == 'Date':
            continue
        df2_lowpass[col] =
butter_lowpass_filter(df2_lowpass[col].values, cutoff, fs,
order)
    return df2_lowpass

def calculateDailySentiment(headlines, model, tokenizer):
    texts = [headline['heading'] for headline in headlines]
    inputs = tokenizer(texts, return_tensors="pt",
truncation=True, padding=True, max_length=512,
                        return_attention_mask=True)
    outputs = model(**inputs)
    logits = outputs.logits
    scores = logits.softmax(dim=1)
    averageScore = scores.mean(dim=0).tolist()
    return averageScore

def analyzeAndSaveSentiment(inputFile, outputFile, model,
tokenizer):
    with open(inputFile, 'r') as file:
        data = json.load(file)
```

```
result = {}

for date, headlines in data.items():
    averageScore = calculateDailySentiment(headlines,
model, tokenizer)
    print(f"{date} > {averageScore}")
    result[date] = averageScore

with open(outputFile, 'w') as outputFile:
    json.dump(result, outputFile, indent=2)

def analyze_and_save_sentiment(input_file, output_file,
isOld=False):
    if isOld:
        try:
            with open(output_file, 'r') as file:
                data = json.load(file)
            if data:
                messagebox.showinfo("Success", "Sentiment data
has already been analyzed and saved.")
            return data
        except FileNotFoundError:
            pass

    modelName = "nlptown/bert-base-multilingual-uncased-
sentiment"
```

```

tokenizer = AutoTokenizer.from_pretrained(modelName)
model =
AutoModelForSequenceClassification.from_pretrained(modelName)

analyzeAndSaveSentiment(input_file, output_file, model,
tokenizer)

with open(output_file, 'r') as file:
    result = json.load(file)

return result

```

```

def create_dataset_with_sentiment(dataset, target,
lookBack=1):
    dataX, dataY = [], []
    for i in range(len(dataset) - lookBack):
        a = dataset[i:(i + lookBack), :]
        dataX.append(a)
        dataY.append(target[i + lookBack])
    return np.array(dataX), np.array(dataY)

```

```

def upload_csv():
    filename = filedialog.askopenfilename(filetypes=[("CSV
files", "*.csv")])
    if filename:
        df = pd.read_csv(filename)
        df.to_csv('./uploaded_data.csv')

```

```

        messagebox.showinfo("Success", f"Data has been
successfully uploaded from {filename}.")

```

```

def create_model(lookBack, featuresScaled, trainSize, trainX,
trainY, languageFeatures):
    inputTimeSeries = tf.keras.layers.Input(shape=(lookBack,
featuresScaled.shape[1]))
    inputLanguage = tf.keras.layers.Input(shape=(1,
languageFeatures.shape[2]))

    lstmTimeSeries = tf.keras.layers.LSTM(80,
return_sequences=True,
activation='leaky_relu')(inputTimeSeries)
    dropoutTimeSeries = Dropout(0.2)(lstmTimeSeries)
    lstmTimeSeries = tf.keras.layers.LSTM(60,
return_sequences=True,
activation='leaky_relu')(dropoutTimeSeries)
    dropoutTimeSeries = Dropout(0.2)(lstmTimeSeries)
    lstmTimeSeries = tf.keras.layers.LSTM(40,
return_sequences=True,
activation='leaky_relu')(dropoutTimeSeries)
    dropoutTimeSeries = Dropout(0.2)(lstmTimeSeries)
    lstmTimeSeries = tf.keras.layers.LSTM(40,
activation='leaky_relu')(dropoutTimeSeries)
    dropoutTimeSeries = Dropout(0.2)(lstmTimeSeries)

    lstmLanguage = tf.keras.layers.LSTM(80,
return_sequences=True, activation='leaky_relu')(inputLanguage)

```

```

dropoutLanguage = Dropout(0.2)(lstmLanguage)
lstmLanguage = tf.keras.layers.LSTM(60,
return_sequences=True,
activation='leaky_relu')(dropoutLanguage)
dropoutLanguage = Dropout(0.2)(lstmLanguage)
lstmLanguage = tf.keras.layers.LSTM(40,
return_sequences=True,
activation='leaky_relu')(dropoutLanguage)
dropoutLanguage = Dropout(0.2)(lstmLanguage)
lstmLanguage = tf.keras.layers.LSTM(40,
activation='leaky_relu')(dropoutLanguage)
dropoutLanguage = Dropout(0.2)(lstmLanguage)

concatenated =
tf.keras.layers.concatenate([dropoutTimeSeries,
dropoutLanguage])

dense = tf.keras.layers.Dense(32, activation='leaky_relu',
kernel_regularizer=tf.keras.regularizers.l2(0.01))(
concatenated)
dropout = Dropout(0.2)(dense)
dense = tf.keras.layers.Dense(16, activation='leaky_relu',
kernel_regularizer=tf.keras.regularizers.l2(0.01))(
dropout)
dropout = Dropout(0.2)(dense)
output = tf.keras.layers.Dense(trainY.shape[1],
activation='linear')(dropout)

```

```
    model = tf.keras.models.Model(inputs=[inputTimeSeries,
inputLanguage], outputs=output)
    model.compile(loss=tf.keras.losses.Huber(),

optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),
                metrics=['mae'])

    model.summary()

    lr_1_history = model.fit([trainX,
languageFeatures[:trainSize]], trainY, epochs=50,
batch_size=256, verbose=2)
    return model

def plot_predictions(data, Y_test, test_predict, trainY,
trainPredict, predictNext):
    fig = Figure(figsize=(20, 10))
    a = fig.add_subplot(111)

    dates = data['Date'].values
    sampleInterval = 60
    sampledDates = dates[::sampleInterval]

    trainY_first = trainY[:, 0]
    trainPredict_first = trainPredict[:, 0]
    test_predict_first = test_predict[:, 0]
    predictNext_first = predictNext[:, 0]
```

```

    a.plot(dates[:len(trainY_first)], trainY_first,
label='Actual Train')

a.plot(dates[len(trainPredict_first):len(trainPredict_first) +
len(test_predict_first)], test_predict_first,
        label='Predicted Test')
    nextDates = [dates[-1] + timedelta(days=i) for i in
range(1, len(predictNext_first) + 1)]
    a.plot(nextDates, predictNext_first, label='Predicted
Future')

plt.xticks(sampledDates, rotation=90)
a.legend()
return fig

```

```

def save_predictions(test_predict, filename):
    test_predict_columns = [f"Predicted_{col}" if col !=
'Date' else col for col in test_predict.columns]
    test_predict.columns = test_predict_columns
    test_predict.to_csv(filename, index=False)
    messagebox.showinfo("Success",
                        f"Predictions have been saved to
{filename}.")

```

```

def save_predictions_button(test_predict, filename):
    save_button = tk.Button(frame, text="Save Predictions",

```

```
        command=lambda:
save_predictions(test_predict, filename))
    save_button.grid(row=3, column=0, columnspan=2, padx=5,
pady=5, sticky='ew')

def load_data(ticker_symbol):
    tickerData = yf.Ticker(ticker_symbol)

    tickerDf = tickerData.history(period='1d', start='2014-12-
31')

    tickerDf = tickerDf.asfreq('D')
    tickerDf.to_csv(f"{ticker_symbol}_data.csv")
    df = pd.read_csv(f"{ticker_symbol}_data.csv")

    return df

def work():
    ticker_symbol = ticker_symbol_entry.get()
    prognosis_number = prognosis_number_entry.get()

    try:
        if not prognosis_number:
            messagebox.showerror("Error", "Please enter the
number of next values to predict.")
            return
        prognosis_number_val = int(prognosis_number)
```

```
except ValueError:
    messagebox.showerror("Error", "Please enter a valid
number for the number of next values to predict.")
    return

try:
    df = fetch_data(ticker_symbol)
    if df.empty:
        messagebox.showerror("Error", "No data found for
the given ticker symbol.")
    else:
        df.to_csv(f"{ticker_symbol}_data.csv")
        messagebox.showinfo("Success",
            f"Data for {ticker_symbol} has
been successfully downloaded and saved to
{ticker_symbol}_data.csv.")

        startDate = df['Date'].iloc[0]
        endDate = df['Date'].iloc[-1]
        if ticker_symbol:
            deltaTime = timedelta(days=1)
        else:
            secondDate = df['Date'].iloc[1]
            deltaTime = secondDate - startDate

        data, isOld = get_news_data(startDate, endDate,
deltaTime)
```

```
        sentimentScores =  
analyze_and_save_sentiment('./headlines.json',  
        './sentiment.json', isOld)  
  
        dfSentiment =  
pd.DataFrame(list(sentimentScores.items()),  
        columns=['jsonDate', 'sentiment'])  
  
        dfSentiment['Date'] =  
pd.to_datetime(dfSentiment['jsonDate'], format='%Y-%m-%d')  
  
        X, featuresScaled, languageFeatures, lookBack,  
scalerTarget, y = preprocess_data(df, dfSentiment)  
  
        datesPredictFuture, predictNext, testPredict,  
testY, trainPredict, trainSize, trainY = make_predictions(X,  
  
featuresScaled,  
  
languageFeatures,  
  
lookBack,  
  
prognosis_number_val,  
  
scalerTarget,  
  
y)
```

```

fig = plot_predictions(df, testY, testPredict,
trainY, trainPredict, predictNext)
    canvas = FigureCanvasTkAgg(fig, master=root)
    canvas.draw()
    canvas.get_tk_widget().pack()
    canvas.get_tk_widget().pack(side=tk.TOP,
fill=tk.BOTH, expand=1)
    canvas.get_tk_widget().pack()
    canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH,
expand=1)

    testPredict =
pd.concat([pd.DataFrame(testPredict),
pd.DataFrame(predictNext)], axis=0)

    testDates = df['Date'].values[trainSize + lookBack
+ 1:]
    testDates = np.append(testDates,
                        [testDates[-1] +
timedelta(days=i) for i in range(1, datesPredictFuture + 1)])

    allColumns = list(df.columns[1:])
    testPredict = pd.DataFrame(testPredict)
    testPredict['Date'] = testDates
    allColumns.append('Date')
    testPredict.columns = allColumns

    save_predictions_button(testPredict,
f"{ticker_symbol}_predictions.csv")

```

```

except Exception as e:
    messagebox.showerror("Error", str(e))
    print(e)

def make_predictions(X, featuresScaled, languageFeatures,
lookBack, prognosis_number_val, scalerTarget, y):
    trainSize = int(len(X) * 0.99)
    trainX, testX = X[0:trainSize, :], X[trainSize:len(X), :]
    trainY, testY = y[0:trainSize], y[trainSize:len(y)]
    model = create_model(lookBack, featuresScaled, trainSize,
trainX, trainY, languageFeatures)
    trainPredict = model.predict([trainX,
languageFeatures[:trainSize]])
    testPredict = model.predict([testX,
languageFeatures[trainSize:]])
    trainPredict =
scalerTarget.inverse_transform(trainPredict)
    trainY = scalerTarget.inverse_transform(trainY)
    testPredict = scalerTarget.inverse_transform(testPredict)
    testY = scalerTarget.inverse_transform(testY)
    datesPredictFuture = prognosis_number_val
    predictNext = []
    for i in range(datesPredictFuture):
        predictNext.append(
            model.predict([X[-1].reshape(1, lookBack,
featuresScaled.shape[1]),

```

```

        languageFeatures[-1].reshape(1, 1,
languageFeatures.shape[2]))))
        featuresScaled = np.append(featuresScaled,
predictNext[-1].reshape(1, featuresScaled.shape[1]), axis=0)
        X = np.append(X, featuresScaled[-lookBack:].reshape(1,
lookBack, featuresScaled.shape[1]), axis=0)
        languageFeatures = np.append(languageFeatures,
        languageFeatures[-
1].reshape(1, 1, languageFeatures.shape[2]),
        axis=0)
        predictNext = scalerTarget.inverse_transform(
        np.array(predictNext).reshape(datesPredictFuture,
predictNext[0].shape[1]))
        return datesPredictFuture, predictNext, testPredict,
testY, trainPredict, trainSize, trainY

def preprocess_data(df, dfSentiment):
    df_filled = fill_missing_with_mean(df, df.columns)
    df2_lowpass = filter_data(df_filled)
    df2_lowpass = df2_lowpass.iloc[1:]
    features = df2_lowpass.drop(['Date'], axis=1).values
    if len(features.shape) == 1:
        features = features.reshape(-1, 1)
    target = df2_lowpass.drop(['Date'], axis=1).values
    if len(target.shape) == 1:
        target = target.reshape(-1, 1)
    scalerFeatures = StandardScaler()
    scalerTarget = StandardScaler()

```

```

featuresScaled = scalerFeatures.fit_transform(features)
targetScaled = scalerTarget.fit_transform(target)
lookBack = 5
X, y = create_dataset_with_sentiment(featuresScaled,
targetScaled, lookBack)
    languageFeatures =
pd.DataFrame(dfSentiment['sentiment'].tolist(),
              columns=['feature1',
'feature2', 'feature3', 'feature4', 'feature5']).values
    languageFeatures = languageFeatures[lookBack:, :]
    languageFeatures = np.reshape(languageFeatures,
(languageFeatures.shape[0], 1, languageFeatures.shape[1]))
    return X, featuresScaled, languageFeatures, lookBack,
scalerTarget, y

```

```

def fetch_data(ticker_symbol):
    if ticker_symbol:
        df = load_data(ticker_symbol)
    else:
        df = pd.read_csv('./uploaded_data.csv')
    df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d
%H:%M:%S-%I:%f')
    df['Date'] = df['Date'].dt.date
    return df

```

```

root = tk.Tk()
root.title("Financial Data Prediction")

```

```
frame = tk.Frame(root)
frame.pack(padx=10, pady=10)

ticker_symbol_label = tk.Label(frame, text="Enter the ticker
symbol:")
ticker_symbol_label.grid(row=0, column=0, padx=5, pady=5)

prognosis_number_label = tk.Label(frame, text="Enter the
number of next values to predict:")
prognosis_number_label.grid(row=1, column=0, padx=5, pady=5)

prognosis_number_entry = tk.Entry(frame)
prognosis_number_entry.grid(row=1, column=1, padx=5, pady=5)
prognosis_number_entry.insert(0, "20")

ticker_symbol_entry = tk.Entry(frame)
ticker_symbol_entry.grid(row=0, column=1, padx=5, pady=5)

upload_button = tk.Button(frame, text="Upload CSV",
command=upload_csv)
upload_button.grid(row=2, column=0, padx=5, pady=5)

fetch_button = tk.Button(frame, text="Fetch Data",
command=work)
fetch_button.grid(row=2, column=1, padx=5, pady=5)

root.mainloop()
```