

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

**ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ ТА МАРКЕТИНГУ  
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

«На правах рукопису»  
УДК 336.76

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Катерина БОЯРИНОВА  
« 13 » \_\_\_\_\_ грудня \_\_\_\_\_ 2022 р.

**МАГІСТЕРСЬКА ДИСЕРТАЦІЯ**

**на здобуття ступеня магістра  
за освітньо-професійною програмою  
«Економічна кібернетика»  
спеціальності 051 Економіка**

**на тему: «Моделювання інвестиційного портфеля E-commerce  
компаній»**

**Виконала:**

Студентка 2-го курсу, групи УК-11мп  
ЮХИМЕНКО Ганна Костянтинівна

**Науковий керівник:**

Доцент кафедри економічної кібернетики,  
к.ф.-м.н, ЛАЗАРЕНКО Ірина Сергіївна \_\_\_\_\_

**Рецензент:**

Завідувачка кафедри економіки і підприємництва,  
д.е.н., проф. ТУЛЬЧИНСЬКА Світлана Олександрівна \_\_\_\_\_

*Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань*

Студентка \_\_\_\_\_

Київ – 2022 року

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

**ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ ТА МАРКЕТИНГУ  
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

**Рівень вищої освіти – другий (магістерський)**

**Спеціальність – 051 Економіка**

**Освітньо-професійна програма «Економічна кібернетика»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Катерина БОЯРИНОВА

« 01 » вересня \_\_\_\_\_ 2022 р.

## **ЗАВДАННЯ**

**на магістерську дисертацію студентки**

**ЮХИМЕНКО ГАННИ КОСТЯНТИНІВНИ**

**1. Тема дисертації:** «Моделювання інвестиційного портфеля E-commerce компаній», керівник роботи Лазаренко Ірина Сергіївна, к.ф.-м.н., затвержені наказом по університету від «10» листопада 2022 року, № 4143-с.

**2. Термін подання студентом дисертації:** 13 грудня 2022 року.

**3. Об'єкт дослідження:** формування інвестиційного портфелю фондових акцій з найменшим ризиком, який враховує інші зовнішні фактори окрім макроекономічних.

**4. Вихідні дані:** інвестиційний портфель цінних паперів, прогнозування вартості фондового активу, оцінка адекватності моделі та прогнозів на основі історичних фактичних даних.

**5. Перелік завдань, які потрібно розробити:**

*а) теоретико-методологічна частина:*

- Оцінити поточний стан фондового ринку;
- Оцінити загальні настрої учасників ринку.

б) дослідницько-аналітична частина:

- Пошук дієвих методів моделювання інвестиційного портфелю;
- Моделювання подальшого розвитку фондового ринку.

в) проектно-рекомендаційна частина:

- Вибір оптимального методу прогнозування вартості акцій із врахуванням зовнішніх факторів;
- Вибір оптимальної моделі для формування інвестиційного портфелю.

#### **6. Орієнтовний перелік ілюстративного матеріалу:**

- 1) алгоритм роботи методу Метрополіса-Хастінгса (Монте Карло з марковськими ланцюгами);
- 2) схема блоку LSTM;
- 3) структура моделі-генератора та моделі-дискримінатора в мережі GAN та загальна схема алгоритму роботи GAN;
- 4) результати прогнозування вартостей акцій досліджуваними методами та порівняння їх точності;
- 5) показники технічних індикаторів розглянутих фондових активів;
- 6) вигляд сету даних з дописами про фондовий актив в соціальній мережі Twitter;
- 7) результати роботи моделі для прогнозування для обраних фондових активів інвестиційного портфелю;
- 8) ваги сформованого портфеля за Марковіцем;
- 9) ваги сформованого портфеля за методом Хуанга-Літценбергера;
- 10) порівняння дохідності портфелів на прогнозованих та реальних даних.

#### **7. Орієнтовний перелік публікацій за напрямом роботи:**

- 1) Стаття: «Прогнозування цін акцій на фондовому ринку за допомогою генеративних змагальних мереж та сентимент-аналізу соціальних мереж» у Науковому журналі «*Modeling the development of the economic systems*» (фахове видання, категорії Б);
- 2) Тези: «Моделювання процесу розвитку фондових активів за допомогою генеративних нейронних мереж», участь у X Міжнародній науково-практичній конференції «*Modern research in world science*», м. Львів.

**8. Дата видачі завдання:** 01 вересня 2022 року.

### 9. Календарний план

<i>№ з/п</i>	<i>Назва етапів виконання магістерської дисертації</i>	<i>Термін виконання етапів магістерської дисертації</i>	<i>Примітка</i>
1	Пошук первинних наукових матеріалів	01.09.2022-22.09.2022	
2	Вивчення основної теоретичної бази для роботи з цінними паперами.	23.09.2022-09.10.2022	
3	Дослідження основних моделей керування інвестиційним портфелем.	10.10.2022-24.10.2022	
4	Пошук методів прогнозування вартості фондового активу.	25.10.2022-05.11.2022	
5	Створення модифікованої моделі прогнозування цін акцій на фондовому ринку та її імплементація у модель керування портфелем.	06.11.2022-16.11.2022	
6	Створення програмного додатку на мові програмування Python.	17.11.2022-24.11.2022	
7	Перевірка моделі на фактичних історичних даних та порівняння результатів.	25.11.2022-03.12.2022	
8	Формулювання висновків після проведення дослідження.	04.12.2022-12.12.2022	

Студент

Ганна ЮХИМЕНКО

Керівник

Ірина ЛАЗАРЕНКО

## РЕФЕРАТ

Магістерська дисертація на тему «Моделювання інвестиційного портфеля E-commerce компаній» містить 89 сторінок, 34 рисунки, 1 таблицю. Перелік посилань нараховує 61 найменування.

*Актуальність теми.* Для сучасного суспільства характерна потреба в точному прогнозуванні. Наприклад, уряди хочуть передбачити тенденцію багатьох показників, таких як безробіття, інфляція, промислове виробництво, а також очікувані податкові надходження, щоб сформулювати ефективну політику.

За своєю суттю прогнозування дуже пов'язане з розпізнаванням закономірностей: припущення про те, що може статися в майбутньому, залежить від розпізнавання повторюваних шаблонів у попередніх реалізаціях. Зараз, особливо за сучасних макроекономічних умов, є непростим завдання передбачити процес розвитку фінансових ринків, просто спостерігаючи за попередніми їх цінами.

*Зв'язок роботи з науковими програмами, планами, темами.* Магістерська дисертація на здобуття ступеня магістра виконувалась в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» відповідно до планів наукових досліджень кафедри економічної кібернетики за темою «Глобалізація напрямків формування промислового потенціалу в умовах постіндустріальних трансформацій» (№ДР 011U007817).

*Мета та завдання роботи.* Визначення дієвих методів прогнозування цін акцій та моделювання інвестиційного портфеля, а також створення програмного продукту, який зможе автоматизувати процес виконання таких цілей.

*Об'єкт та предмет дослідження.* Об'єктом дослідження даної роботи є процес формування інвестиційного портфелю фондових акцій з найменшим ризиком, який враховує інші зовнішні фактори окрім макроекономічних, а предметом – оцінка ситуації на фондовому ринку, прогнозування ринкових цін фондових акцій у майбутньому та оцінка потенційних ризиків на ринку.

У ході виконання роботи було розглянуто 3 акції, зокрема, Microsoft, Amazon, та Facebook (Meta). Модель адаптується до типів акцій, що розглядаються та коригує свої передбачення з метою отримання мінімальної похибки від фактичних. На виході модель видає стратегію керування портфелем на протязі наступного місяця, побудовану на основні прогнозованих даних.

*Методи дослідження.* В ході виконання магістерської дисертації було досліджено різні методи прогнозування ринкових цін акцій, було розроблено модель для визначення оптимального інвестиційного портфелю та його вартість. Після цього даний алгоритм було автоматизовано у вигляді розробленого програмного додатку.

*Наукова новизна.* Наукова новизна роботи полягає у тому, що до цього не було апробовано генеративні адверсійні мережі для прогнозування цін акцій із додаванням сентимент аналізу соціальних мереж. До цього були апробовані роботи лише на історичних ринкових даних без показнику настрою ринку.

*Практична значущість.* Дана робота показує що тип нейронних мереж GAN дає гарні результати для прогнозування цін акцій. Але при сучасній нестабільній ситуації на ринку просто використати нейронну мережу недостатньо, тому після додавання сентимент аналізу учасників ринку модель є більш стійкою до несподіваних змін на фондовому ринку.

*Апробація результатів роботи.* Результатом дослідження є створення моделі керування портфелем, яка прогнозує розвиток використаних фондових активів та створює стратегію керування ними на основі отриманих прогнозованих даних. Ефективність моделі перевіряється на прогнозованих та фактичних історичних даних за розглянутий часовий період.

– Стаття: Юхименко Г.К., Лазаренко І.С. Прогнозування цін акцій на фондовому ринку за допомогою генеративних змагальних мереж та сентимент-аналізу соціальних мереж. *Modeling the development of the economic systems*. 2022. (фахове видання, категорії Б);

– Тези: Юхименко Г.К. Моделювання процесу розвитку фондових активів за допомогою генеративних нейронних мереж. *Modern research in world science*: матеріали X Міжнародної науково-практичної конференції, 25-27 грудня 2022 року. м. Львів, 2022.

**Ключові слова:** акція, фондовий ринок, інвестиційний портфель, модель Марковіца, модель Хуанга-Літценбергера, рекурентні нейронні мережі, згорткові нейронні мережі, генеративні нейронні мережі, сентимент аналіз, соціальні мережі, метод ARIMA, метод Метрополіса-Хастінгса.

## SUMMARY

Master's thesis on "Modeling the investment portfolio of E-commerce companies" contains 89 pages, 34 figures, 1 table. The list of references includes 61 title.

*The relevance of the topic.* Modern society is characterized by the need for accurate forecasting. For example, governments want to predict the trend of many indicators, such as unemployment, inflation, industrial production, as well as expected tax revenues in order to formulate effective policies.

At its core, forecasting is very much about pattern recognition: guessing what might happen in the future depends on recognizing repeating patterns in previous realizations. Now, especially in the current macroeconomic environment, it is a difficult task to predict the development of financial markets simply by observing their previous prices.

*Connection of work with scientific programs, plans, topics.* Master's thesis for a master's degree was carried out at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" in accordance with the research plans of the Department of Economic Cybernetics on the topic "Optimization of investment portfolio management using neural networks".

*Purpose and objectives of the work.* Determination of effective methods of stock price forecasting and investment portfolio modeling, as well as the creation of a software product that can automate the process of achieving such goals.

*Object and subject of research.* The object of study of this work is the process of forming an investment portfolio of stocks with the lowest risk, which takes into account other external factors other than macroeconomic, and the subject is to assess the situation on the stock market, forecasting market prices of stocks in the future and assessing potential risks in the market.

In the course of the work, 3 stocks were considered, in particular, Microsoft, Amazon, and Facebook (Meta). The model adapts to the types of stocks considered and adjusts its predictions in order to obtain a minimum error from the actual ones. As an output, the model produces a portfolio management strategy for the next month based on the main predicted data.

*Research methods.* During the master's thesis, various methods of forecasting stock market prices were investigated, a model was developed to determine the optimal investment portfolio and its value. After that this algorithm was automated in the form of developed software application.

*Scientific novelty.* The scientific novelty of the work is that generative adversarial networks for forecasting stock prices with the addition of sentiment

analysis of social networks have not been tested before. Before that, the work was tested only on historical market data without an indicator of market sentiment.

*Practical significance.* This work shows that the type of neural networks GAN gives good results for predicting stock prices. But in the current unstable market situation, it is not enough to use a neural network, so after adding sentiment analysis of market participants, the model is more resistant to unexpected changes in the stock market.

*Approbation of work results.* The result of the research is the creation of a portfolio management model that predicts the development of the used stock assets and creates a strategy for managing them based on the obtained forecast data. The effectiveness of the model is tested on the predicted and actual historical data for the considered time period.

– Article: Yukhymenko H.K., Lazarenko I.S. (2022) Prohnozuvannia tsin aksii na fondovomu rynku za dopomohoiu heneratyvnykh zmahalnykh merezh ta sentyment-analizu sotsialnykh merezh. [Forecasting the stock prices using generative adversarial networks and sentiment analysis of social networks]. *Modeling the development of the economic systems.* (in Ukrainian)

– Theses: Yukhymenko H.K. (2022) Modeliuvannia protsesu rozvytku fondovykh aktyviv za dopomohoiu heneratyvnykh neironnykh merezh. [Modeling the process of development of stock assets using generative neural networks]. *Modern research in world science.* X International Scientific and Practical Conference, December 25-27<sup>th</sup>, 2022. Lviv, 2022. (in Ukrainian)

**Keywords:** stock, stock market, investment portfolio, Markowitz model, Huang-Litzenberger model, recurrent neural networks, convolutional neural networks, generative neural networks, sentiment analysis, social networks, ARIMA method, Metropolis-Hastings method.

## ЗМІСТ

ВСТУП .....	10
1 ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ ІНВЕСТИЦІЙНОЇ ДІЯЛЬНОСТІ.....	14
1.1 Опис інвестиційного портфелю та методів його формування .....	14
1.2 Основні моделі формування інвестиційних портфелів.....	19
1.3 Оцінка ефективності побудованого портфелю. ....	23
Висновки до розділу 1. ....	24
2 ПОСТАНОВКА ЕКОНОМІЧНОЇ ЗАДАЧІ ТА ОПИС МЕТОДУ РОЗВ'ЯЗАННЯ.....	25
2.1 Постановка завдання.....	25
2.2 Дослідження методів прогнозування вартості акцій.....	26
2.3 Апробація методів прогнозування вартості акцій .....	40
Висновки до розділу 2. ....	45
3 ОЦІНКА РЕЗУЛЬТАТІВ АЛГОРИТМУ МОДЕЛЮВАННЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ .....	47
3.1 Підготовка даних для тренування моделі.....	47
3.2 Створення програмного додатку для оцінки адекватності моделі та розв'язку економічної задачі.....	56
3.3. Програмна реалізація моделі прогнозування вартості акцій за допомогою мережі GAN .....	64
Висновки до розділу 3. ....	80
ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	84

## ВСТУП

*Актуальність роботи.* Для сучасного суспільства характерна потреба в точному прогнозуванні. Наприклад, уряди хочуть передбачити тенденцію багатьох показників, таких як безробіття, інфляція, промислове виробництво, а також очікувані податкові надходження, щоб сформулювати ефективну політику. Менеджери з маркетингу хочуть передбачити попит на продукцію, обсяги продажів і зміни в уподобаннях споживачів, щоб прийняти правильні рішення щодо поточної та майбутньої політики та, загалом, сформулювати відповідне стратегічне планування.

За своєю суттю прогнозування дуже пов'язане з розпізнаванням закономірностей: припущення про те, що може статися в майбутньому, залежить від розпізнавання повторюваних шаблонів у попередніх реалізаціях. Звичайно, передбачення на основі минулого ґрунтується на вірі в те, що майбутнє не передбачає суттєвих інновацій порівняно з тим, що ми вже можемо спостерігати. Але зараз, особливо за сучасних макроекономічних умов, є непростим завдання передбачити процес розвитку фінансових ринків, просто спостерігаючи за попередніми їх цінами.

Протягом багатьох років з'являлося все більше і більше точних програм, які допомагали визначити, коли продавати чи купувати цінні папери, і як інвестиційні банки, так і компанії, зареєстровані на біржі, тепер значною мірою покладаються на алгоритмічну торгівлю, щоб визначити, як працювати на фінансових ринках. Якщо виключити новини, які можуть вплинути на поведінку акцій, на рух ціни значною мірою впливають висновки, які ці алгоритми роблять на основі коливань ціни. Певним чином той факт, що взаємодія на фінансових ринках дедалі більше керується алгоритмами, полегшує прогнозування тенденції ціни на закриття акцій.

Останніми роками генеративні моделі GAN показали багатообіцяючі результати у вирішенні багатьох складних проблем (наприклад, генерування реалістичних зображень і відео, переклад із зображень на зображення та

тексту на зображення), але питання ефективності використання даного типу мереж для прогнозування цін фондових активів все ще є суперечливим.

*Дослідження наукової літератури.* Перед проведенням зазначеного вище дослідження було проаналізовано сучасні наукові праці з даної теми в темах моделювання інвестиційних портфельів, прогнозування вартостей фондових активів за допомогою різних методів, зокрема, нейронних мереж, а також оцінки зовнішніх факторів впливу на ціну акції.

*Зв'язок роботи з науковими програмами, планами, темами.* Магістерська дисертація на здобуття ступеня магістра виконувалась в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» відповідно до планів наукових досліджень кафедри економічної кібернетики за темою «Глобалізація напрямків формування промислового потенціалу в умовах постіндустріальних трансформацій» (№ДР 011U007817).

*Мета дослідження та завдання роботи.* Визначення дієвого методу моделювання інвестиційного портфелю фондових акцій, а також створення алгоритму, який буде проводити дані дії автоматично.

Завдання дослідження:

- Оцінка сьогоденного становища на фондовому ринку;
- Вибір потенційних кандидатів у інвестиційний портфель, які можуть принести прибуток у майбутньому;
- Прогнозування майбутньої вартості обраного фондового активу;
- Розробка програми, яка автоматизує формування довгострокової інвестиційної стратегії;
- Реагування на можливі різкі позитивні чи негативні фактори впливу на ціну акції;
- Конфігурація портфельних стратегій для адаптації до перемін на фондовому ринку;

*Об'єкт дослідження.* Створення та планування портфелю фондових активів з мінімальною волатильністю, який враховує інші зовнішні фактори окрім макроекономічних.

*Предмет дослідження.* Аналіз стану ринку фондових акцій, їх можливих майбутніх інвестиційних ризиків та прогнозування руху їх ринкової вартості.

*Теоретична основа дослідження.* Для розроблення моделі формування портфелю було розглянуто дві моделі – Марковіца та Хуанга-Літценбергера. Спочатку пропонується варіант використання моделі Марковіца для розробки портфелю на старті інвестицій, а другий метод (Хуанга-Літценбергера) формує стратегію його керування. Запропонований варіант краще реагує на ринкову волатильність, оскільки ваги активів будуть коригуватись відповідно до змін цін акцій, більше того, другий метод розглядає оптимальну область портфелів з очікуваною прибутковістю від 0% до 100%.

Далі у ході виконання роботи було розроблено модель прогнозування вартості фондового активу, при створенні якої було досліджено чотири методи прогнозування – метод Бокса-Дженкінса (ARIMA), метод LSTM (довгочасна коротка пам'ять), метод Метрополіса-Хастінгса та генеративні нейронні мережі GAN. Далі було використано найкращий метод на основі модифікованих даних.

Використовуючи історичні дані, була отримана покращена модель об'єкту дослідження, яка була перевірена на адекватність і є більше ефективною у використанні.

В інформаційну базу для виконання роботи та дослідження входять історичні дані про вартість та продажі фондових активів, які є у відкритому доступі на порталі Yahoo Finance, а також інформаційно – аналітичні матеріали від організацій, які проводять наукові розробки щодо об'єкту дослідження, опубліковані у періодичних і спеціалізованих виданнях, а також розміщені у мережі Інтернет.

*Методи дослідження.* Під час роботи було використано декілька методів дослідження. По-перше, була використана техніка порівняльного аналізу для порівняння існуючих стратегій управління портфелем і бізнес-моделей для створення ефективного портфеля. Пізніше за допомогою гіпотетичних дедуктивних методів були виявлені можливості вдосконалення раніше розроблених моделей. Потім були висунуті гіпотези, які були підтверджені в цій роботі і стали основою для подальших досліджень. Завдяки методу формалізації було встановлено порядок результатів дослідження за допомогою формальних символів. Моделювання як удосконалений метод дослідження в навчальній практиці включає практичну частину проведення експериментів і перевірку валідності експериментів.

*Наукова новизна.* Наукова новизна роботи полягає у тому, що до цього не було апробовано генеративні адверсійні мережі для прогнозування цін акцій із додаванням сентимент аналізу соціальних мереж. До цього були апробовані роботи лише на історичних ринкових даних без показнику настрою ринку.

*Практична значущість.* Дана робота показує що тип нейронних мереж GAN дає гарні результати для прогнозування цін акцій. Але при сучасній нестабільній ситуації на ринку просто використати нейронну мережу недостатньо, тому після додавання сентимент аналізу учасників ринку модель є більш стійкою до несподіваних змін на фондовому ринку.

# 1 ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ ІНВЕСТИЦІЙНОЇ ДІЯЛЬНОСТІ

## 1.1 Опис інвестиційного портфелю та методів його формування

Інвестиції є важливою частиною нашого життя. На макrorівні вони допомагають нам досягти економічного зростання в країнах, у які ми інвестуємо. На мікрорівні вони допомагають збільшити наше багатство та зробити життя комфортнішим. Багато акціонерів сприймають інвестиції як можливість швидко збільшити свій капітал. Якщо все зроблено правильно, інвестори можуть заробити більшу частину своїх грошей завдяки ROI.

Сьогодні замість того, щоб інвестувати в один фінансовий актив, люди вважають за краще вкладати свої гроші в те, що ми називаємо портфелем; групу фінансових активів (акції, облігації, опціони тощо). Раціональні інвестори диверсифікують свої портфелі, щоб мінімізувати несистематичний ризик. Великою проблемою залишається розподіл активів, який визначається як «інвестиційна стратегія, призначена для балансування ризику та прибутку шляхом розподілу портфельних активів відповідно до індивідуальних цілей, толерантності до ризику та інвестиційного горизонту». Інвестори шукають найкраще поєднання інвестицій відповідно до своїх смаків. Іншими словами, вони завжди шукають найкращий результат.

У такому світі, як наш, майже неможливо підрахувати всі різні параметри, які впливають на ринок. Ринки в усьому світі постійно змінюються, адаптуючись до різноманітних подій щосекунди. Акції Tesla впали в серпні 2018 року після того, як Ілон Маск дав знамените інтерв'ю The New York Times у п'ятницю. Ціна акцій впала в результаті короткого інтерв'ю. Можливе пояснення полягає в тому, що інвестори боялися, що Tesla не працюватиме на повну потужність, або що тодішній генеральний директор

відчував себе невідповідним і перевтомленим, що призвело до нижчої якості продукції.

Щоб подолати цю ситуацію та врахувати якомога більше параметрів, математики та статистики розробили низку моделей, які можуть найкраще вирішити проблему розподілу активів. Важливо мати на увазі, що кожна представлена модель або теорія має власний набір припущень. З цієї причини перед використанням запропонованої моделі слід розглянути індивідуальні припущення. Крім того, інвестори повинні мати можливість оцінити своє уникнення ризику. Управління ризиками не слід розглядати як тривіальне, оскільки це важливий критерій правильного аналізу розподілу активів. Інвестування в дуже мінливі акції може бути поганим для потенційно небезпечних і ризикованих акціонерів [18].

Вибір портфеля є однією з найпоширеніших проблем, з якими стикаються різні інвестори з різним рівнем капіталу, і це також одна з найскладніших проблем у фінансовому світі. Проблема вибору портфеля є моделлю, яка врівноважує ризик і прибутковість. Він складається з групи цінних паперів і намагається визначити пропорцію інвестицій у кожну з них, яка мінімізує інвестиційний ризик і максимізує віддачу від інвестицій. Однак високі доходи зазвичай пов'язані з високим ризиком.

Ефективне управління активами – це більше, ніж просто визначення необхідних інвестицій: воно також вимагає оптимальної структури активів у портфелі. Це тому, що інвестиційна поведінка портфеля зазвичай відрізняється від активів у ньому. Наприклад, ризик портфеля облігацій США. На акції, як правило, припадає половина середнього ризику в акціях. Обачливі інвестори несуть відповідальність за ризики та прибутки своїх інвестиційних портфелів.

Розуміння ефективної структури портфеля має вирішальне значення для оптимального управління поверненням інвестицій портфеля. Ефективне управління портфелем знижує ризик, одночасно збільшуючи прибутковість.

Ефективність портфеля так само важлива для вдумливого інвестора, як оцінка ризику та прибутковості активів.

Більшість інституційних інвесторів і фінансових економістів визнають інвестиційні переваги ефективної диверсифікації портфеля. Оптимальне управління портфельними ризиками є важливою частиною сучасного управління активами. Марковіц (1959, 1987) дав класичне визначення оптимальності портфеля: якщо портфель має найвищий очікуваний (середній або очікуваний) дохід при даному рівні ризику (дисперсії) або, що еквівалентно, при певному рівні з найменшим ризиком, портфель є ефективним з урахуванням очікуваної прибутковості всіх портфелів універсальних цінних паперів [19].

Інвестиційний портфель – це група матеріальних і фінансових активів, спеціально підготовлена для інвестиційної діяльності відповідно до обраної стратегії. Більшість інвесторів вирішують інвестувати в кілька фінансових інструментів.

Інвестиційний портфель компанії складається з усього основного капіталу компанії, зібраного та керованого в цілому (з єдиної точки зору) відповідно до її інвестиційної політики. Це означає, що портфель – це набір інструментів, обраних для інвестування відповідно до стратегічного напрямку інвестора.

Інвестиційний портфель цінних паперів – це цільове інвестування в цінні папери з метою загального управління цінними паперами. Портфель може складатися з однотипних цінних паперів (акції) або різних активів (акції, облігації, ощадні облігації тощо). Портфелі цінних паперів, таким чином, являють собою безліч різних фінансових цінностей, які в поєднанні служать засобами для досягнення інвесторами конкретних цілей. Основним призначенням інвестиційного портфеля є реалізація політики фінансового інвестування шляхом вибору найбільш прибуткових і безпечних фінансових інструментів. Новаторська робота Гаррі Марковіца в 1952 році поклала початок сучасній теорії портфеля (H.P. Markowitz, Portfolio Selection, Journal

of Finance 1, 1952). Результати Марковіца були розвинені й доповнені не менш відомими роботами Джеймса Тобіна, Вільяма Шарпа та інших дослідників сучасної теорії портфеля. Американський учений Гаррі Марковіц (1952), За портфельною теорією, інвестори вкладаються у різні типи активів. Усі інвестори намагаються максимізувати прибутковність свого портфеля, інвестуючи в цінні папери, але прибутковність завжди прямо пропорційна ризику, який готовий прийняти інвестор. Тому мета кожного інвестора – знайти прийнятну комбінацію цих факторів. Основною метою створення інвестиційного портфеля є реалізація політики фінансового інвестування компанії шляхом вибору найбільш прибуткових і безпечних фінансових інструментів. Після уточнення цілей інвестиційної стратегії ми розставляємо пріоритети при створенні інвестиційних портфелів та оптимізуємо співвідношення інвестиційного капіталу для кожного типу портфеля.

Найважливішими кроками у створенні портфеля цінних паперів є:

- вибір оптимального типу портфеля;
- оцінка прийнятного коефіцієнта дохідності ризику;
- визначення початкового складу портфеля;
- вибір системи управління портфелем.

Кожен інвестор створює портфель цінних паперів відповідно до власних уявлень та уподобань. Проте існують загальноприйняті принципи побудови інвестиційних портфелів. Одним з найважливіших принципів інвестування є прибутковність, яка визначається як здатність інвестора отримувати прибуток у майбутньому. Це включає різницю в ціні, дивіденди або відсотки за цими цінними паперами. Він визначається як чистий прибуток компанії та відсоток її статутного капіталу, який використовується для розподілу доходу від інвестицій у цінні папери (виражений у відсотках). Далі порівняйте дивідендну дохідність із середньою процентною ставкою за довгостроковими депозитами для домогосподарств і підприємств. Якщо співвідношення цих двох коефіцієнтів менше 1, прибутковність цих цінних паперів є недостатньою, що призводить до відтоку інвесторів [2].

Норму прибутку від інвестицій за певний період часу можна виразити як відношення загального доходу (або відсотка) до різниці в ціні придбання цінного паперу. Найприбутковішими є звичайні акції молодих компаній, які беруться за ризиковані проекти, але обіцяють високі прибутки за сприятливих умов.

Ще одна мета інвесторів при виборі цінних паперів – інвестувати в цінні папери. Безпека – це здатність емітента брати на себе відповідальність за кошти, накопичені інвесторами, і захистити їх від можливих потрясінь. Держава забезпечує захист інвесторів через законодавчу базу (наприклад, українське законодавство забороняє випуск цінних паперів для покриття збитків компаній). Найменш безпечними є звичайні акції, а найбільш стабільними (за безпекою) – державні облігації, привілейовані акції та облігації (в стабільній економіці).

Іншим принципом інвестування є збільшення інвестувань. Акції із високим зростанням вважаються простими акціями молодих компаній, які впроваджують нові технології та ноу-хау, хоча у той же час такі цінні папери також дуже ризиковані для інвестора. Довгострокові облігації та привілейовані акції забезпечують середнє зростання капіталу. Короткострокові облігації пропонують найменший приріст капіталу.

Коефіцієнт ліквідності для інвестора – це здатність швидко і безболісно перетворювати цінні папери в гроші. Ліквідність – це володіння фінансовими активами, які необхідно перетворити на грошові кошти. При визначенні ступеня ліквідності враховується час, необхідний для перетворення фінансового активу в грошові кошти та пов'язані з цим витрати. Тому для визначення ступеня ліквідності інвестори використовують такі показники:

- час перетворення коштів у грошові кошти;
- сума відповідних фінансових витрат інвестора. Самі гроші мають найвищу ліквідність.

Забезпечення ліквідності фінансових інвестицій є важливою умовою захисту інвестицій від інфляції та своєчасного перетворення їх у фінансові активи. Ліквідність облігацій та ощадних сертифікатів залежить від дати погашення, а акцій та інвестиційних сертифікатів – від попиту на фондовій біржі. Для того щоб зберегти високу ліквідність необхідно інвестувати у свій портфель лише ті акції, ліквідність яких є стабільною та перевіреною.

Отже, створюючи портфель цінних паперів, інвестор повинен дотримуватись основних правил для створення інвестиційного портфелю:

- доходність;
- безпека вкладень;
- ріст капіталу;
- ліквідність [1].

## **1.2 Основні моделі формування інвестиційних портфелів**

### 1) Модель Марковіца

У сучасній науці математичних фінансів оптимізація портфеля стала фундаментальною та центральною темою для розуміння фондового ринку та прийняття рішень. Теорія Гаррі Марковіца відома як сучасна теорія портфеля (MPT) – ця теорія дає відповідь на фундаментальне питання про те, як інвестори мають розподіляти свої гроші за всіма можливими комбінаціями активів. Усі теорії, що передували теорії Марковіца, не допускали компромісу між прибутковістю інвестицій та ризиком. Марковіц використовував середнє значення та дисперсію для кількісної оцінки прибутковості та ризику і показав, що і прибутковість, і ризик слід розглядати разом, щоб зробити оптимальний вибір.

Є кілька причин, через які ця теорія зробила революцію в інвестиційних рішеннях. По-перше, теорія показує, що ризик портфеля залежить не лише від ризиків його окремих складових, а й їх спільних рухів.

За основу методу береться несистематична диверсифікація ризику. Класична думка полягає в тому, що інвестори намагаються вибрати цінний папір, який максимізує майбутній грошовий потік за поточної ціни. Однак портфельна теорія Марковіца передбачає, що кореляції між компонентами портфеля відіграють важливу роль у визначенні загального ризику, що є ще одним фактором, який слід враховувати під час прийняття рішень. По-друге, теорія визначила процес прийняття рішень як оптимізацію. Це говорить про те, що за певного рівня очікуваної прибутковості оптимальний або ефективний портфель повинен мати найменший ризик. Інші комбінації цінних паперів вважаються неефективними портфелями через їхню високу волатильність [6].

Портфель інвестора складається з певної кількості видів цінних паперів. Кожен цінний папір характеризується певною очікуваним прибутком ( $return - r_i$ ) за період  $n$  (формула 1.1):

$$r_i = \frac{C_{in} - C_{io}}{C_{io}}, \quad (1.1)$$

де  $C_{io}$  – вартість активу у початковий часовий період.

Очікувана доходність активу розраховується як (формула 1.2):

$$r^* = \frac{1}{t} \sum_{i=1}^t r_i, \quad (1.2)$$

де  $t$  – кількість минулих спостережень дохідності активу.

Ризик активу розраховується через середнє квадратичне відхилення (дисперсію) його дохідності (формула 1.3):

$$\sigma_i = var(r_i) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (r_i - \bar{r})^2}, \quad (1.3)$$

де  $\bar{r}$  – середнє арифметичне ринкової вартості активу.

Для того щоб визначити залежність дохідності між активами необхідно розрахувати коефіцієнт коваріації (формула 1.4):

$$\text{cov}(r_i, r_j) = \sigma_{ij} = \frac{1}{t-1} \sum_{i=1}^t \sum_{j=1}^k (r_i - \bar{r}_i)(r_j - \bar{r}_j). \quad (1.4)$$

Для постановки задачі оптимізації позначимо  $i=1, \dots, n$  типів портфельних активів, долі активів (ваги) як  $w_i$ , а цільова функція буде виражатись через формулу дохідності портфелю, яку ми будемо максимізувати (формула 1.5):

$$r_p = \sum_{i=1}^n w_i * r_i \rightarrow \max, \quad (1.5)$$

де  $w_i$  – доля активу у інвестиційному портфелі.

Обов'язковим обмеженням у задачі є умова (формула 1.6):

$$\sum_{i=1}^n w_i = 1. \quad (1.6)$$

Якщо модель Марковіца вважається без урахування можливості коротких продаж (значення ваги є від'ємним), то додаються такі обмеження як (формула 1.7):

$$w_i \geq 0. \quad (1.7)$$

Під короткими продажами мається на увазі ситуація, коли актив купується в борг за визначеною ціною, потім продається коли його ринкова вартість падає і через деякий час купується за нижчою ціною, що дає змогу отримати прибуток при зниженні вартості активу [3, 60].

## 2) Метод Хуанга-Літценбергера

Якщо завдання оптимізації не має обмежень на ваги активів (тобто короткі продажі не заборонені), то ефективна множина може бути отримана математично. Хуанг і Літценбергер показали, як знайти дві точки допустимої множини і отримати з цих точок всю ефективну множину. Цей розділ бере цей математичний підхід за основу та представляє серію розрахунків у

матричній формі для отримання загального рішення для портфельів із будь-якою кількістю активів.

Вектор очікуваних доходностей назвемо  $e$ , ваговий вектор –  $w$ , а одиничний вектор –  $u$ . Коваріаційну матрицю назвемо  $V$ . Дисперсія портфеля в матричній формі обчислюється як  $w^T V w$ .

У методі Хуанга і Літценбергера для пошуку ефективних портфельів потрібно обчислити матрицю  $V^{-1}$ , обернену до коваріаційної матриці. Щоб знайти два ефективних портфеля (позначимо їх  $g$  і  $g + h$ ), Хуанг і Літценбергер обчислюють чотири скалярні величини ( $A, B, C$  і  $D$ ). Перші три є добутками векторів і матриць, а четверта залежить від трьох попередніх (формула 1.8):

$$A = u^t V^{-1} e; B = e^t V^{-1} e; C = u^t V^{-1} u; D = BC - A^2. \quad (1.8)$$

Якщо визначити два проміжних вектора-стовпця  $l = V^{-1} e$  і  $m = V^{-1} u$ , то матричні вирази зводяться до такого вигляду (формула 1.9):

$$A = u^t l; B = e^t l; C = u^t m. \quad (1.9)$$

Нижче наведені формули для обчислення ваг активів, що представляють дві точки на кривій ефективної множини рішень – портфель  $g$  (з очікуваною прибутковістю 0%) і портфель  $g + h$  (з очікуваною прибутковістю 100%), які мають такий вигляд (формула 1.10):

$$g = \frac{Bm - Al}{D}; h = \frac{Cm - Al}{D}. \quad (1.10)$$

Таким чином,  $g$  з вагами для акцій відповідно повертає портфель на ефективній кордоні з нульовою очікуваною прибутковістю. Аналогічно: ваги вектора  $g + h$  повертають другий ефективний портфель з очікуваною прибутковістю 100%. Для отримання вагового вектору ефективного портфеля з будь-якої заданої очікуваною прибутковістю  $T$  можна використовувати лінійну комбінацію  $g + h * T$  відомих векторів  $g$  і  $h$ . Таким чином, портфель з мінімальним ризиком для очікуваної прибутковості

складається з комбінації різних акцій, що є результатом, отриманим за методом Хуанга і Літценбергера [4–5, 28].

### 1.3 Оцінка ефективності побудованого портфелю.

Коефіцієнт Шарпа являє собою відносний показник дохідність-ризик інвестиційного фонду і відображає у скільки разів рівень надлишкової дохідності вище рівня ризику інвестиції (формула 1.11):

$$\gamma_{sharp} = \frac{(P_i(t) - R_f)}{\sigma^2}, \quad (1.11)$$

де  $P_i(t)$  – очікуваний дохід портфелю;

$R_f$  – безризикова ставка дохідності;

$\sigma^2$  – стандартне відхилення портфелю (міра ризику) [25, 26].

Стандартне відхилення – це міра середньої статистичної дисперсії того, наскільки змінилася прибутковість портфеля за певний період часу. Інвестори використовують стандартне відхилення минулої ефективності, щоб передбачити діапазон найімовірніших прибутків для конкретної інвестиції.

Діапазон очікуваних показників є широким, а високе стандартне відхилення інвестицій означає високу волатильність.

Більше стандартне відхилення вказує на те, що інвестиція є оптимальною щодо очікуваного співвідношення ризик/прибуток. Ускладнення можуть виникнути, лише якщо стандартне відхилення занадто високе. Таким чином, основна концепція коефіцієнта Шарпа полягає в оцінці суми додаткового доходу у випадку збільшення волатильності шляхом купівлі активів з високим ризиком, а не активів з низьким ризиком. Чим вище співвідношення, тим краще, отже:

- високий коефіцієнт Шарпа означає більший дохід на одиницю ризику;

– низький коефіцієнт Шарпа є ознакою того, що інвестор бере на себе великий ризик, щоб отримати додатковий дохід [27, 61].

### **Висновки до розділу 1.**

В першому розділі було досліджено теоретичну базу в сфері моделювання інвестиційної діяльності на фондовому ринку. Для цього було розглянуто основні моделі інвестиційних портфельів та їх показники якості.

За портфельною теорією Марковіца, за певного рівня очікуваної прибутковості оптимальний або ефективний портфель повинен мати найменший ризик. Інші комбінації цінних паперів вважаються неефективними портфелями через їхню високу волатильність. Отже, портфель Марковіца дає на вихід оптимальний портфель, який має найменший ризик і є найбільш пристосованим до волатильності ринку. Але проблема даного методу полягає в тому, що в даному методі оптимізаційна задача має обмеження тільки на позитивні ваги. Метод Хуанга-Літценбергера не має даного обмеження і дає змогу отримати як позитивні, так і негативні ваги, що є сигналом для коротких торговельних операцій (продаж/купівля). Перевагою даного методу є також те, що він враховує множину портфельів з очікуваною прибутковістю від 0% до 100% і повертає оптимальний портфель з даної області.

Коефіцієнт Шарпа є найбільш відповідним і використовуваним показником для оцінки ефективності отриманого інвестиційного портфеля. Це відносна міра ризику повернення взаємного фонду та відображає, як часто рівень надлишкової прибутковості перевищує рівень інвестиційного ризику. Основна концепція коефіцієнта Шарпа полягає в оцінці суми додаткового доходу від підвищеної волатильності при купівлі активів з високим рівнем ризику, а не активів з низьким рівнем. Чим вищий коефіцієнт, тим кращий портфель.

## 2 ПОСТАНОВКА ЕКОНОМІЧНОЇ ЗАДАЧІ ТА ОПИС МЕТОДУ РОЗВ'ЯЗАННЯ

### 2.1 Постановка завдання

На портфель фондових акцій інвестор виділяє грошовий фонд у розмірі 150 000\$, а список активів у портфелі містить у собі 3 акції:

- AMZN – Amazon;
- MSFT – Microsoft Inc.;
- META – Meta Inc..

Необхідно розрахувати початковий портфель для основних вкладень, а також оптимізувати його на наступні 30 днів. Далі калібрувати портфель відповідно до персональної оцінки ситуації на фондовому ринку від інвестора.

На першому етапі виконання завдання буде використана модель Марковіца з обмеженням на від'ємні значення вагів. Цільовою функцією в оптимізаційній задачі буде показник дохідності портфелю. Також розрахуємо коефіцієнт Шарпа для перевірки адекватності розв'язку.

Отже, при зміні ціни базового активу, тобто, на кожному часовому проміжку, необхідно змінювати стратегію інвестування та визначити, чи необхідно купувати/продавати певні акції. Для цього завдання використання моделі Марковіца є не зовсім доцільним, оскільки оптимізаційна задача цієї моделі не передбачає від'ємні ваги активів (що означало б продаж частки акцій) [12–13].

На основі отриманих прогнозів кожен часовий період ваговий вектор портфелю буде коригуватись за методом Хуанга-Літценбергера, відповідно будуть перераховані вартості куплених активів та портфелю.

В динаміці вартість та відповідно модель портфелю будуть мати вигляд (формула 2.1):

$$V(t) = \sum_{i=1}^n V'_i(t); \quad \sum_{i=1}^n w'_i(t) * r'_i(t) \rightarrow \max, \quad (2.1)$$

де  $V'_i(t)$  – об'єми вкладень в  $n$  активів на момент часу  $t$  ( $i = 1, \dots, n$ );

$r'_i(t)$  – дохідність  $i$ -ого активу в момент часу  $t$ .

Доля  $i$ -ого активу в момент часу  $t$  буде становити (формула 2.2):

$$w'_i(t) = \frac{V'_i(t)}{V(t)}. \quad (2.2)$$

Розрахунки виконуються на основі прогнозованих даних, які отримуються методом генеративних нейронних мереж, на вхід до якої подаються підготовлені до тренування дані із попереднім додаванням додаткових незалежних змінних, наприклад, різні технічні індикатори (ковзна середня тощо) [60, 61].

## 2.2 Дослідження методів прогнозування вартості акцій

Оскільки модель робить прогноз на певний часовий період, то вважаємо, що модель буде прогнозувати часовий ряд.

Часовий ряд – це послідовний набір точок даних, що вимірюється, як правило, протягом послідовних часів. Це математично визначається як набір векторів  $x(t)$ ,  $t = 0, 1, \dots$  де  $t$  являє собою час, що минув. Змінна  $x(t)$  розглядається як випадкова величина. Вимірювання, проведені під час події в часовому ряді, розташовані у відповідному хронологічному порядку.

Часовий ряд, у якому одна змінна записана у часі, називається одновимірним. Але коли розглядаються записи більш як однієї змінної, це називається багатовимірним. Часові ряди можуть бути безперервними чи дискретними. У безперервних часових рядах спостереження вимірюються в

довільні моменти часу, тоді як дискретні ряди часу спостереження, виміряні в дискретні моменти часу. Наприклад, концентрації хімічних викидів у річках, показники вологості повітря та інших можна позначити як безперервний часовий ряд. З іншого боку, населення конкретного міста, виробництво фірми або обмінний курс двох різних валют можуть бути дискретними часовими рядами. Дискретні часові ряди зазвичай записують послідовні спостереження через рівні проміжки часу, такі як вартові, денні, щотижневі, місячні або річні часові інтервали. Вважають, що змінна, що спостерігається в дискретному часовому ряду, вимірюється як безперервна змінна з використанням реальної шкали. Крім того, безперервні часові ряди можуть бути легко перетворені на дискретні часові ряди шляхом об'єднання даних через певні інтервали часу.

Моделювання та прогнозування часових рядів – багатообіцяюча сфера досліджень, яка привертає увагу дослідницької спільноти в останні десятиліття. Основна мета моделювання часових рядів полягає у ретельному зборі та ретельному аналізі історичних спостережень за часовими рядами для розробки відповідних моделей, що пояснюють внутрішню структуру часових рядів. Потім цю модель використовують для створення майбутніх значень ряду, тобто для створення прогнозів. Отже, прогнозування часових рядів можна назвати актом передбачення майбутнього шляхом розуміння минулого.

Оскільки прогнозування часових рядів є важливим у багатьох практичних областях, таких як бізнес, економіка, фінанси, наука та технології, моделі повинні бути написані так, щоб відповідати базовим часовим рядам. Зрозуміло, що успішне прогнозування часових рядів залежить від правильного налаштування параметрів моделі. За минулі роки дослідники доклали чимало зусиль для розробки ефективних моделей, що підвищують точність прогнозів. У результаті наукової літератури розробили різні важливі моделі прогнозування часових рядів [14, 20].

При виборі методу прогнозування даних було розглянуто 4 варіанти: метод Бокса-Дженкінса (ARIMA), рекурентні нейронні мережі LSTM, метод Метрополіса-Хастінгса (Монте-Карло для марковських ланцюгів) та генеративні змагальні мережі GAN. Лінійна регресія та рандомізовані марковські ланцюги можуть генерувати близькі до правдивих значення, але не розпізнають різкі зміни у цінах. Інші два методи мають більшу залежність від даних, поданих на тренування, тому при неточних даних поданих на вхід різниця між фактичними та прогнозованими даними може бути занадто високою.

– Метод Монте-Карло з марковськими ланцюгами

Метод Монте-Карло з ланцюгом Маркова циклічно перебирає функцію, що містить випадкові змінні, так що кожен цикл представляє унікальну подію.

Цей метод називається Монте-Карло для ланцюгів Маркова, оскільки ланцюг Маркова генерується з використанням попередніх значень вибірки для створення випадкових значень наступної вибірки. Оскільки ймовірність переходу між вибірковими значеннями є лише функцією попереднього значення, тобто задовольняє умові відсутності післядії, то з урахуванням поточного стану майбутнє не залежить від минулого.

Метод Монте-Карло (МСМС) для ланцюгів Маркова — це набір алгоритмів, які реалізують ланцюги Маркова за певними правилами, включаючи алгоритми Метрополіса, Метрополіса-Хастінгса та Гіббса.

При застосуванні методів Монте-Карло виникає проблема отримання вибірок з деякого багатовимірного розподілу ймовірностей  $p(x)$ . Зокрема, були спроби інтегрувати дуже складні функції випадковим моделюванням. Результатом цих спроб є алгоритм Метрополіса–Хастінгса [9, 19].

Цей алгоритм потребує використання простого розподілу, який називається допоміжною функцією розподілу (або перехідною функцією)

$Q(\theta'/\theta)$ , щоб виділити підходящі значення з апостеріорного розподілу  $P(\theta = \theta / D)$ .

Алгоритм Метрополіса–Хастінгса витягує випадкову вибірку з розподілу, приймаючи або відкидаючи перехід до нового значення залежно від того, наскільки вірогідною є вибірка. Така випадкова прогулянка "без пам'яті" є частиною "марковських ланцюгів" МКМЛ.

Ймовірність кожного нового зразка визначається функцією  $f$ . Функція  $f$  має бути пропорційна заданому розподілу, з якого ми хочемо взяти зразки. Функція  $f$  зазвичай вибирається як функція щільності ймовірності, яка виражає цю пропорційність.

Щоб отримати нове значення зразка, необхідно запропонувати нову  $\theta'$ , яка є випадковою вибіркою з функції розподілу  $Q(\theta'/\theta)$ , яка зазвичай є симетричною, наприклад, нормальний розподіл із середнім значенням  $\theta$  та деяким стандартним відхиленням  $\sigma$ :  $Q\left(\frac{\theta'}{\theta}\right) = N(\theta, \sigma)$ . Критерієм прийняття зразка  $\theta'$  є формула (рис. 2.1):

$$P(\text{accept}) = \begin{cases} \frac{\prod_i^n f(d_i / \theta = \theta') P(\theta')}{\prod_i^n f(d_i / \theta = \theta) P(\theta)}, & \prod_i^n f(d_i / \theta = \theta) P(\theta) > \prod_i^n f(d_i / \theta = \theta') P(\theta') \\ 1, & \prod_i^n f(d_i / \theta = \theta) P(\theta) \leq \prod_i^n f(d_i / \theta = \theta') P(\theta') \end{cases}$$

Рисунок 2.1 – Формула критерію прийняття зразка за МКМЛ

(складено автором)

Це означає, що якщо  $\theta'$  має більшу ймовірність за  $\theta$ , то ми завжди приймаємо  $\theta'$  та навпаки.

Псевдокод алгоритму Метрополіса–Хастінгса:

Дано:

–  $f$  – функція розподілу, з якої робиться вибірка, у нашому випадку це буде функція рівномірного розподілу (рис. 2.2):

$$F_X(x) \equiv \mathbb{P}(X \leq x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & x \geq b \end{cases}.$$

Рисунок 2.2 – Функція рівномірного розподілу  
(складено автором)

- $q$  – функція переходу – формула (2.3):

$$q = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; \quad (2.3)$$

- $\sigma = 1, \mu = 0$ ;
- $N$  – кількість симуляцій = 1 000 000;
- $\theta_0 = q(0) = \frac{1}{\sqrt{2\pi}} e^0 = \frac{1}{\sqrt{2\pi}} \approx 0,399$ .

Для  $N$  ітерацій:

- $p = f(D / \theta = \theta')P(\theta)$ ;
- $\theta' = Q(\theta_i)$ ;
- $p' = f(D / \theta = \theta')P(\theta')$ ;
- $ratio = \frac{p'}{p}$ ;
- Генеруємо за рівновірним розподілом число  $r$  у проміжку  $[0, 1]$ ;
- Якщо  $r < ratio$ , то визначаємо  $\theta_i = \theta'$  [11].

Визначимо формулу розрахунку ринкової ціни активу як (формула 2.4):

$$S_t = S_0 e^{(r-0,5\sigma^2)\Delta t + \sigma\sqrt{\Delta t}z_t}, \quad (2.4)$$

де  $z_t$  – коефіцієнт зміни вартості базисного активу опціону на ринку, який генерується за допомогою алгоритму Метрополіса-Хастінгса.

Після виконання алгоритму отримуємо прогнозовані дані для подальших розрахунків [8–10, 21].

- ARIMA (метод Бокса-Дженкінса)

Однією з найпопулярніших і часто використовуваних моделей стохастичних часових рядів є модель (ARIMA). Ця модель показує, що розглянутий часовий ряд є лінійним і відповідає певному відомому статистичному розподілу, такому як нормальний розподіл. Моделі ARIMA мають підкласи інших моделей, наприклад моделі авторегресії (AR), моделі ковзного середнього (MA) і моделі авторегресії ковзного середнього (ARMA).

Для сезонного прогнозування часових рядів Бокс і Дженкінс запропонували досить вдалий варіант моделі ARIMA, а саме. Сезонна ARIMA (SARIMA). Популярність моделі ARIMA в основному пояснюється її здатністю легко представляти багато типів часових рядів у поєднанні з простотою її реалізації та відповідним методом Бокса-Дженкінса для оптимізації процесу побудови моделі. Але серйозним обмеженням цих моделей є попередньо прийнята лінійна форма відповідного часового ряду, яка стає недійсною в багатьох практичних ситуаціях. Для подолання цього недоліку в літературі запропоновано різні нелінійні стохастичні моделі, однак з точки зору реалізації вони не настільки прості, як моделі ARIMA [22].

Під час виконання експериментів ми подаємо у модель історичну вартість розглянуто фондового активу за останній рік (з 30/09/2021 по 30/09/2022), а прогнозована вартість на наступний часовий період розраховується як (формула 2.5):

$$\hat{Y}_t = \varphi_0 + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \varphi_3 Y_{t-3} + \varepsilon_t - \theta_1 \varepsilon_{t-1}. \quad (2.5)$$

Сама модель має такий вигляд (2.6):

$$Y_t = \varphi_0 + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \dots + \varphi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q}, \quad (2.6)$$

де  $Y_t$  – фактичне значення;

$\varepsilon_t$  – випадкова похибка при  $t$ ;

$\varphi_i$  та  $\theta_j$  – коефіцієнти моделі;

$p$  та  $q$  – показники авторегресії та ковзаної середньої відповідно [23].

– Модель нейронної мережі LSTM

Фондовий ринок - це платформа, де покупці та продавці торгують акціями, змінюючи ціни вгору та вниз. Ціни на акції щодня коливаються через ринкові принципи. Ці зміни викликані законами попиту та пропозиції. Якщо більше людей хочуть купити (вимагати) акцію, ніж продати (пропонувати) її, ціна зростає. І навпаки, якщо більше людей хочуть продати акції, ніж купити, пропозиція перевищить попит і ціни впадуть. Ці механізми легко зрозуміти. Важко зрозуміти, що спонукає людей купувати певні акції та продавати інші, а фондові ринки за своєю суттю дуже мінливі та непередбачувані. Тому інвестори завжди ризикують, щоб отримати прибуток. Люди інвестують у фондовий ринок відповідно до своєї особистої схильності до ризику та очікують повернення своїх інвестицій. Ціни на акції майже неможливо передбачити через волатильність, фактору, який значною мірою впливає на рух цін на акції. Однак цілком ймовірно можна оцінити майбутні ціни.

Фінанси створюють дані часових рядів. Часовий ряд – це набір точок даних, проіндексованих за часом. Часовий ряд зазвичай є послідовністю, взятою в послідовні рівновіддалені моменти часу, тобто, послідовністю даних у дискретному часі. Дані часових рядів тут стосуються індикаторів, які відстежують коливання цін на акції в певний день. Такими показниками ефективності є ціна відкриття («Open Price»), ціна закриття («Close Price»), щоденна мінімальна ціна («Low»), щоденна висока ціна («High») і загальний обсяг акцій, якими торгували протягом періоду (дня) («Volume») [15].

Рекурентна нейронна мережа (RNN) — це штучна нейронна мережа, у якій нейрони посилають один одному сигнали зворотного зв'язку. Ідея RNN полягає у використанні послідовної інформації. Традиційні нейронні мережі припускають, що всі входи (і виходи) незалежні один від одного. Але для багатьох завдань це не найкраща ідея. Краще знати слова, які постають перед

вами, особливо якщо ви хочете передбачити наступне слово в реченні. RNN називаються рекурсивними, оскільки вони виконують одне й те саме завдання для кожного елемента послідовності, а результат залежить від попередніх обчислень. Інший спосіб уявлення про RNN полягає в тому, що вони мають пам'ять, яка зберігає інформацію про те, що було обчислено до цього часу. Зверніть увагу, що кожна RNN теоретично може обробляти інформаційну послідовність будь-якої довжини, але на практиці може відображати лише кілька кроків. [44]

Такі нейронні мережі є найефективнішими системами для аналізу послідовних часових рядів даних, пошуку зв'язків між історичними даними та прогнозування нових даних. Історичні дані, у цьому контексті, є даними минулих часових рядів. Це одна з найважливіших і цінних частин для майбутніх цінових спекуляцій. Нейронна мережа — це взаємопов'язаний обчислювальний компонент, що складається з шарів штучних нейронів (вузлів мережі), які можуть обробляти вхідні дані та передавати виходи іншим вузлам мережі. Вузли з'єднані ребрами або вагами, які впливають на силу сигналу та кінцевий вихід мережі. Мережі довгострокової пам'яті, які зазвичай називають просто «LSTM», є особливим типом RNN зі спеціальним механізмом закривання, який контролює доступ до комірок пам'яті. З самого початку вентилі LSTM та їх варіанти показали перспективу у вирішенні різноманітних завдань моделювання послідовності машинного навчання, таких як обробка природної мови, підписи до зображень і розпізнавання мови. По суті, блок LSTM складається з вхідного вектора, вектора забувального шлюзу та вихідного вектора [16, 17].

LSTM добре підходить для класифікації, обробки та прогнозування часових рядів з урахуванням часових лагів невідомої тривалості та тренує модель за допомогою зворотного розповсюдження.

Припустимо, що  $x_t$  є входом, а  $h_{t-1}$  — прихованим результатом останнього часового кроку  $t-1$ , вхідний вектор вирішує, скільки нової

інформації буде додано до стану комірки  $c_t$ , і генерує кандидат стану  $\tilde{c}_t$  за допомогою (формули 2.7–2.8):

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad (2.7)$$

$$\tilde{c}_t = \varphi(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (2.8)$$

де  $i_t$  можна сприймати як ручку, яку LSTM навчається вибірково розглядати  $\tilde{c}_t$  для поточного кроку часу,  $\sigma$  – логістична сигмоїдна функція, а  $\varphi$  –  $\tanh$ .

Як правило, терміном  $W$  позначають вагові матриці (наприклад,  $W_{xi}$  – матриця ваг від входу до входного затвора), а  $b$  – вектори зсуву.

Вектор забувного вентиля вирішує, як попередня інформація зберігатиметься в новий крок часу і визначається як (формула 2.9):

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f). \quad (2.9)$$

Потім вектор стану комірки  $c_t$  оновлюється за допомогою (формула 2.10):

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t, \quad (2.10)$$

де  $*$  елементним добутком векторів. Потім вихідний вектор використовує вихідний  $o_t$  для управління тим, що потім зчитується з нового стану комірки  $c_t$  на прихований вектор  $h_t$  наступним чином (формули 2.11–2.12):

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \quad (2.11)$$

$$h_t = o_t * \varphi(c_t), \quad (2.12)$$

де  $W$  та  $b$  включають вагові матриці та вектори зміщення. Значення  $W$  і  $b$  визначаються на етапі тренування моделі.

В даній роботі функціональна модель LSTM використовується у вигляді (формула 2.13):

$$(h_t, c_t) = LSTM(x_t, h_{t-1}, c_{t-1}, W, b). \quad (2.13)$$

На виході моделі отримуємо ваги, за якими потім виконується прогноз даних [18].

– Модель генеративної нейронної мережі GAN

Генеративні змагальні мережі (GAN) – це інструменти для генеративного моделювання з використанням методів глибокого навчання, таких як глибокі згорткові нейронні мережі.

Генеративне моделювання – це неконтрольоване навчальне завдання в машинному навчанні, яке передбачає автоматичне виявлення та вивчення шаблонів у вхідних даних, щоб можна було використовувати модель для створення або отримання нових моделей, які можна отримати з вихідного набору даних. GAN здатні реалістично генерувати нові обличчя, а також змінювати зображення, розфарбовуючи фотографії, старіючи обличчя з високою роздільною здатністю.

GAN складається з дискримінатора та генератора. Дискримінатор намагається відрізнити справжні зразки від згенерованих зразків, тоді як генератор використовує інформацію від дискримінатора, щоб підробити справжні зразки. GAN унікальна з багатьох інших генеративних моделей. Замість явної вибірки з розподілу ймовірностей GAN використовує глибоку нейронну мережу як прямий генератор, який генерує зразки з випадкових шумів [39, 42–43].

Одночасно навчаються дві моделі: генеративна модель  $G$ , яка фіксує розподіл даних, генеруючи нові екземпляри даних, і дискримінаційна модель  $D$ , яка оцінює їх на автентичність, тобто оцінює ймовірність того, що вибірка походить із навчальних даних, а не була згенерована з  $G$ . Процедура навчання  $G$  полягає в тому, щоб максимізувати ймовірність помилки  $D$ . В оригінальній роботі  $D$  і  $G$  грають у *min-max* гру для двох гравців із функцією значення  $V(D, G)$  (формула 2.14):

$$\min_g \max_d V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} \left[ \log (1 - D(G(z))) \right], \quad (2.14)$$

де  $p_{data}(x)$  – отриманий розподіл за даними  $x$ , а  $p_z(z)$  – апріорна інформація про вхідний шум.

Використана архітектура GAN має мережу з 5 блоків LSTM як генератор і п'ятишарову CNN як дискримінатор. Дискримінатор був реалізований з 5 згортковими шарами, за якими слідують 3 сплюснені шари і два шари із лінійною та сигмоїдною активаціями відповідно [29–31].

Розглянемо структуру генератора, яка складається з блоків LSTM. Довгострокові мережі короткочасної пам'яті – зазвичай їх просто називають "LSTM" – це особливий тип RNN, оснащений спеціальним механізмом затвора, який контролює доступ до комірок пам'яті. Розробка блоку LSTM та його варіантів дало великі перспективи у вирішенні різних завдань моделювання послідовностей в машинному навчанні – наприклад, обробка природної мови, субтитри до зображень та розпізнавання мови. Схема блоку LSTM наведена на рис. 2.3:

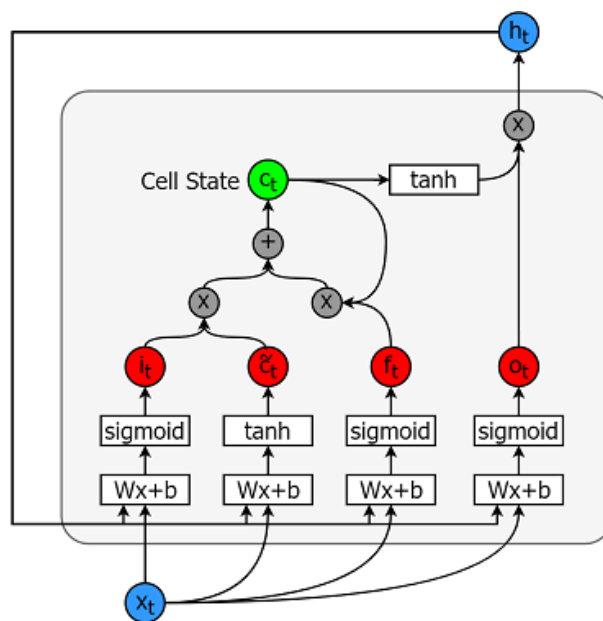


Рисунок 2.3 – Структура блоку LSTM  
(складено на основі [52])

В основному, блок LSTM складається з вхідного вектору, вектору забувального вентиля та вихідного вектору [51, 52].

Вихід генератора  $G(X)$  визначається наступним чином (формули 2.15–2.16):

$$h_t = g(X), \quad (2.15)$$

$$G(X) = \widehat{x_{t+1}} = \delta(W_h^T h_t + b_h), \quad (2.16)$$

де  $G(\cdot)$  позначає вихід LSTM, а  $h_t$  є вихід LSTM з  $X = \{x_1, \dots, x_t\}$  як вхід.  $\delta$  використовується для функції активації Leaky Rectified Linear Unit (ReLU).  $W_h$  і  $b_h$  позначають вагу і зсув у повному шарі моделі. Ми також використовуємо відсіювання як метод регуляризації, щоб уникнути переобладнання, після чого ми можемо продовжувати прогнозування  $\widehat{x_{t+2}}$  за допомогою  $\widehat{x_{t+1}}$  та  $X$ . Повна схема моделі генератора виглядає наступним чином (рис. 2.4):

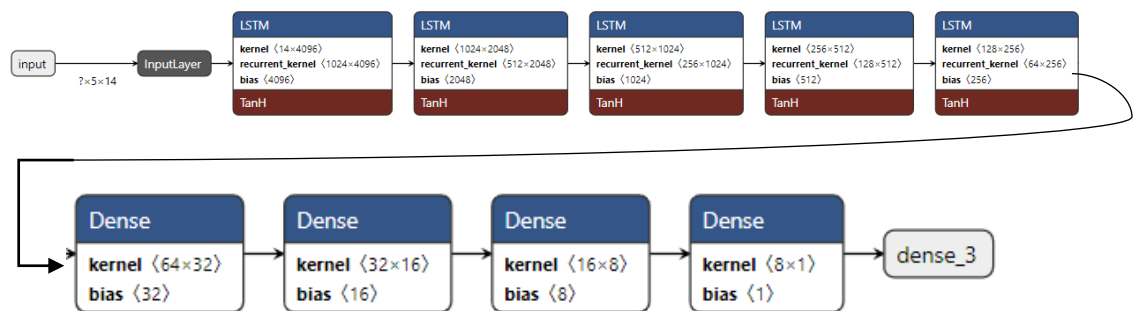


Рисунок 2.4 – Структура моделі генератора  
(складено автором)

Тепер перейдемо до структури моделі дискримінатора. Метою дискримінатора є створення диференційованої функції  $D$  для класифікації вхідних даних. Результатом дискримінатора є ймовірність, яка вказує на походження вибірки даних. Ймовірність 1 або число, дуже близьке до 1, означає, що вибірка даних є реальними даними. Ймовірність 0 або число, близьке до 0, означає підроблені дані. Коли ймовірність близька до 0,5, це означає, що дискримінатору важко визначити підроблені зразки. Генератор неодноразово навчається, щоб вихід  $D$  був рівним 1 для вибірок даних, згенерованих генератором. Модель навчається, доки не буде досягнуто рівноваги Неша, коли зміна стратегії більше не змінює гру. Рівновага Неша досягається, коли генератор отримав можливість генерувати дані, близькі до реальних даних. Дискримінатор не розрізняє реальні дані та дані генератора.

Вважається, що генератор тепер має вивчати розподіл реальних даних [32, 33].

CNN – це один з видів нейронних мереж для обробки даних які мають вигляд сітки, таких як зображення, натхненна організацією зорової кори тварин і розроблена для автоматичного та адаптивного навчання ієрархії просторових ознак, від шаблонів низького до високого рівня. CNN – це математична конструкція, яка зазвичай складається з трьох типів шарів (або блоків): шарів переплетення, зв'язувальних шарів і повністю зв'язаних шарів. Перші два шари, згортка та конкатенація, виконують виділення ознак, тоді як третій, повністю зв'язаний шар перетворює навчені характеристики в кінцевий результат, наприклад класифікацію (0 або 1). Оскільки один рівень передає свої результати на наступний рівень, витягнуті функції можуть стати ієрархічними та поступово складнішими (рис. 2.5) [45, 46].

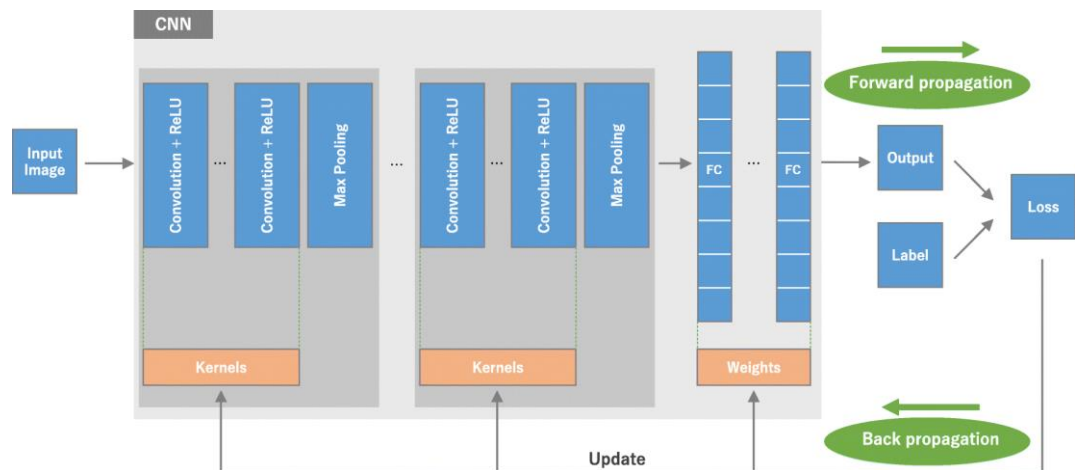


Рисунок 2.5 – Стандартна структура моделі CNN

(складено на основі [46])

Процес оптимізації таких параметрів, як ядро, називається навчанням, яке виконується для мінімізації різниці між результатом і реальними даними, тобто, похибку прогнозування за допомогою алгоритму оптимізації, що містить у собі зворотне поширення і градієнтний спуск [32].

Схема моделі-дискримінатора (рис. 2.6):

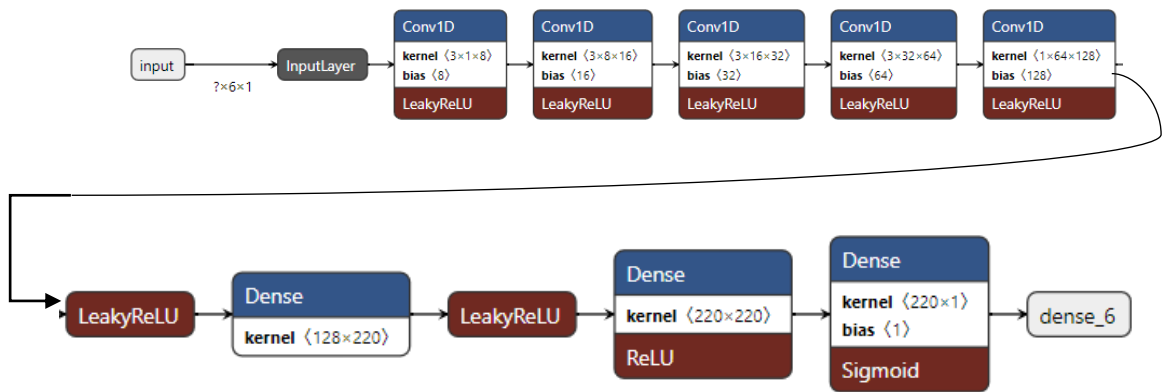


Рисунок 2.6 – Структура моделі-дискримінатора CNN  
(складено автором)

Загальна структура моделі GAN в поєднанні двох вище наведених моделей визначена на рис. 2.7:

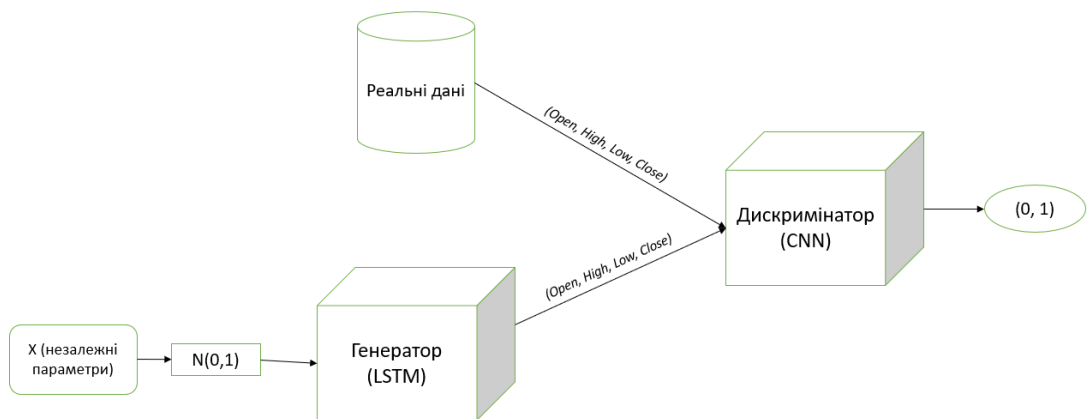


Рисунок 2.7 – Загальна структура моделі GAN  
(складено автором)

GAN – це новий фреймворк, який тренується на основі взаємодії двох моделей у вигляді гри з нульовою сумою. У змагальному процесі генератор можна розглядати як шахрая, який генерує дані, подібні до реальних даних, тоді як дискримінатор грає роль судді, щоб розрізнити реальні дані та згенеровані дані. Вони можуть досягти ідеальної точки, коли дискримінатор не зможе розрізнити два типи даних. На цьому етапі генератор може запам’ятати розподіл даних з цієї гри [34, 35].

## 2.3 Апробація методів прогнозування вартості акцій

На цьому етапі ми виконаємо прогнозування майбутніх цін акцій за допомогою методів, вказаних у минулому розділі.

Протестуємо запропоновані методи на акціях e-commerce компанії Amazon (\$AMZN) (рис. 2.8–2.11).

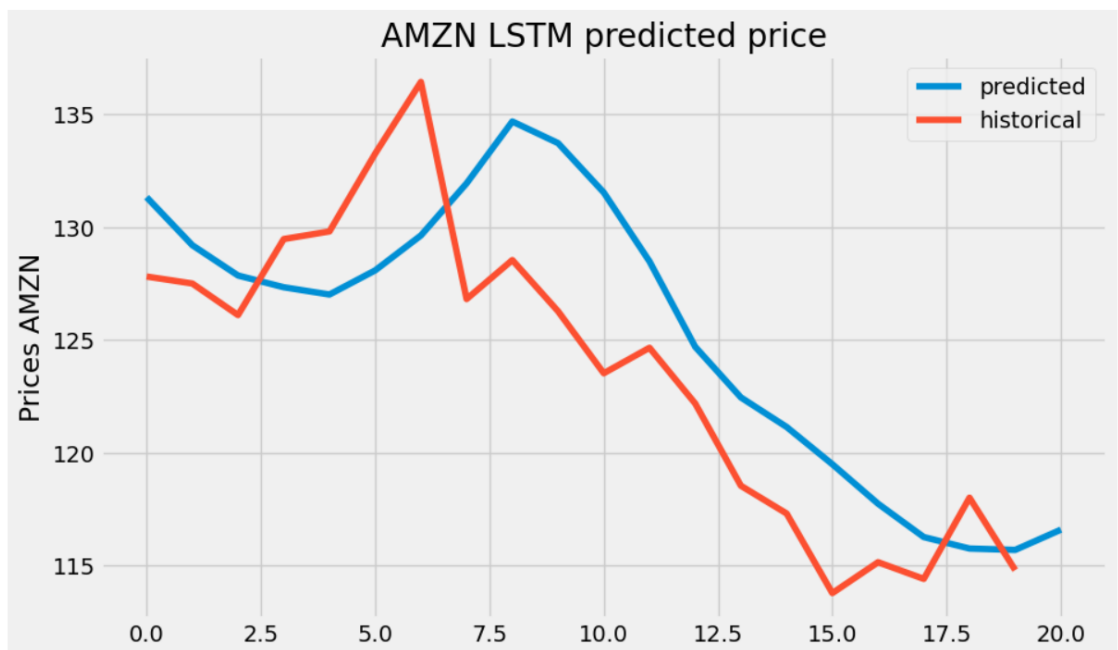


Рисунок 2.8 – Прогнозні дані вартості акцій AMZN методом LSTM  
(розраховано автором на основі використаних даних)

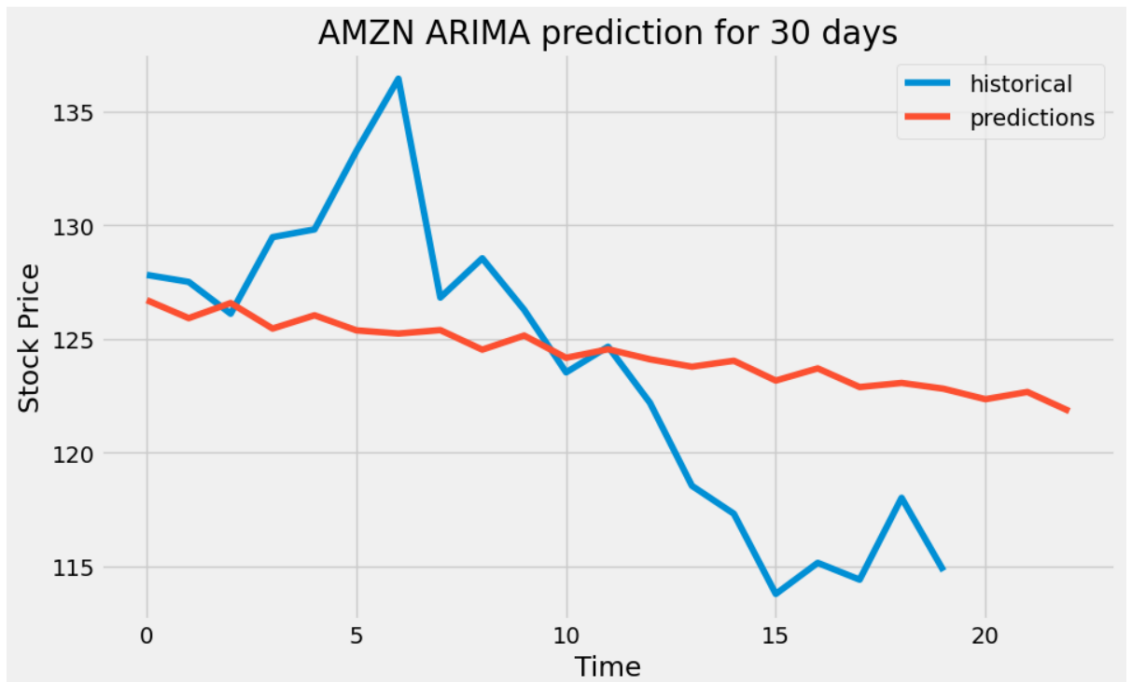


Рисунок 2.9 – Прогнозні дані вартості акцій AMZN методом ARIMA  
(розраховано автором на основі використаних даних)

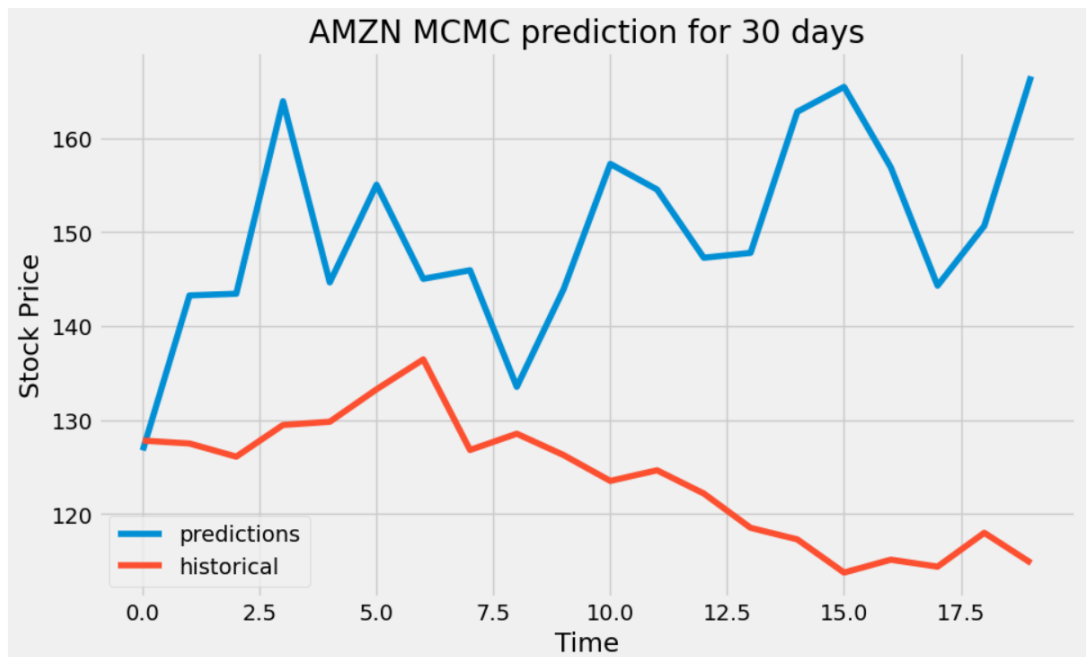


Рисунок 2.10 – Прогнозні дані вартості акцій AMZN методом MCMC  
(розраховано автором на основі використаних даних)

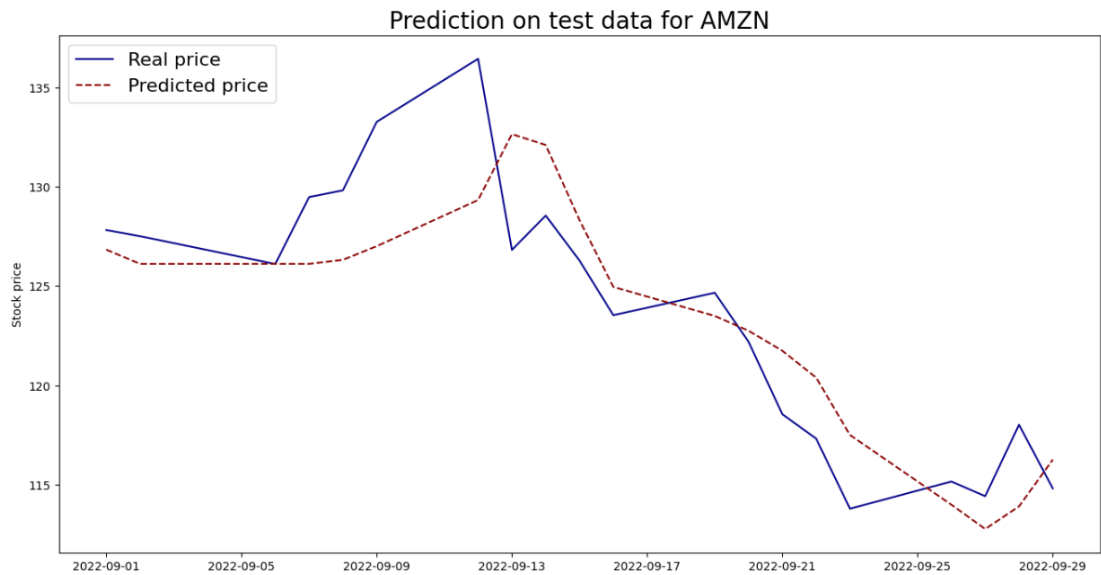


Рисунок 2.11 – Прогнозні дані вартості акцій AMZN методом GAN  
(розраховано автором на основі використаних даних)

Ефективність прогнозування цін акцій визначимо за допомогою RMSE – root mean squared error, тобто, середньої квадратичної похибки між прогнозними та реальними даними. Порівняння результатів роботи методів згаданих вище наведено на рис. 2.12:

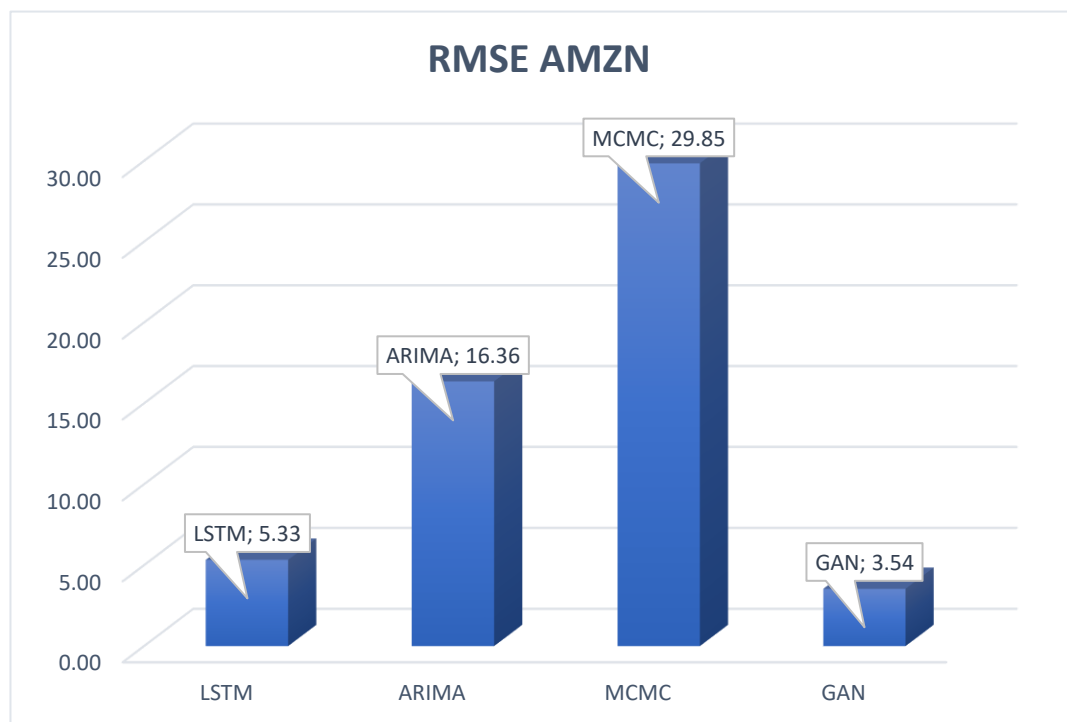


Рисунок 2.12 – Порівняння похибок прогнозів за різними методами  
(розраховано автором на основі використаних даних)

Тепер для того щоб переконатись в правильності вибору, проведемо прогнозування акцій компанії Tesla (\$TSLA) (рис. 2.13–2.16):

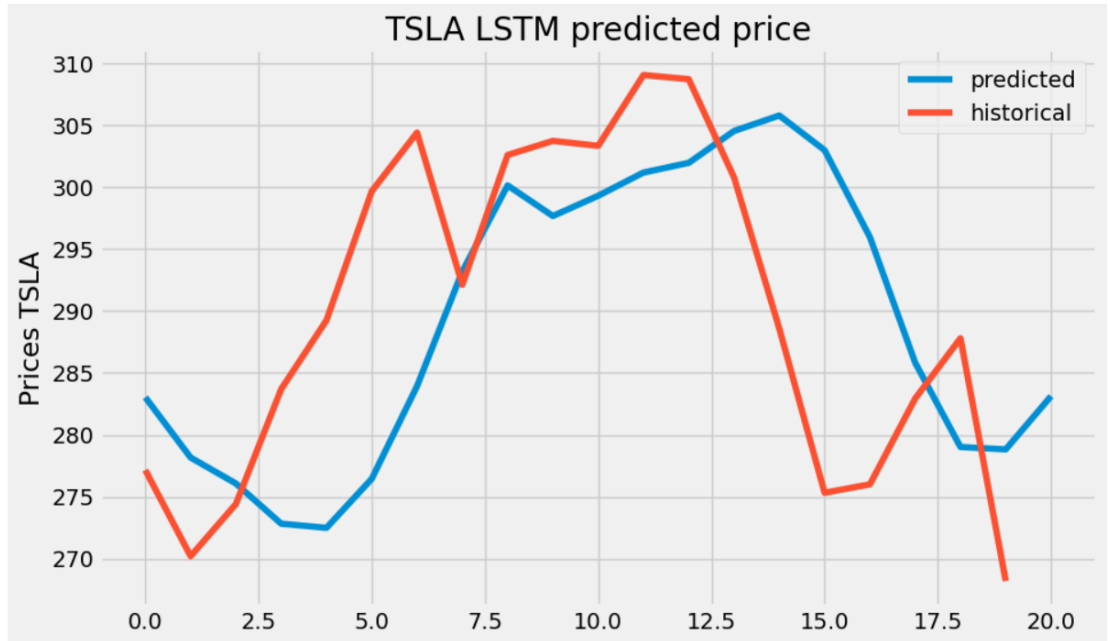


Рисунок 2.13 – Прогнозні дані вартості акцій TSLA методом LSTM (розраховано автором на основі використаних даних)

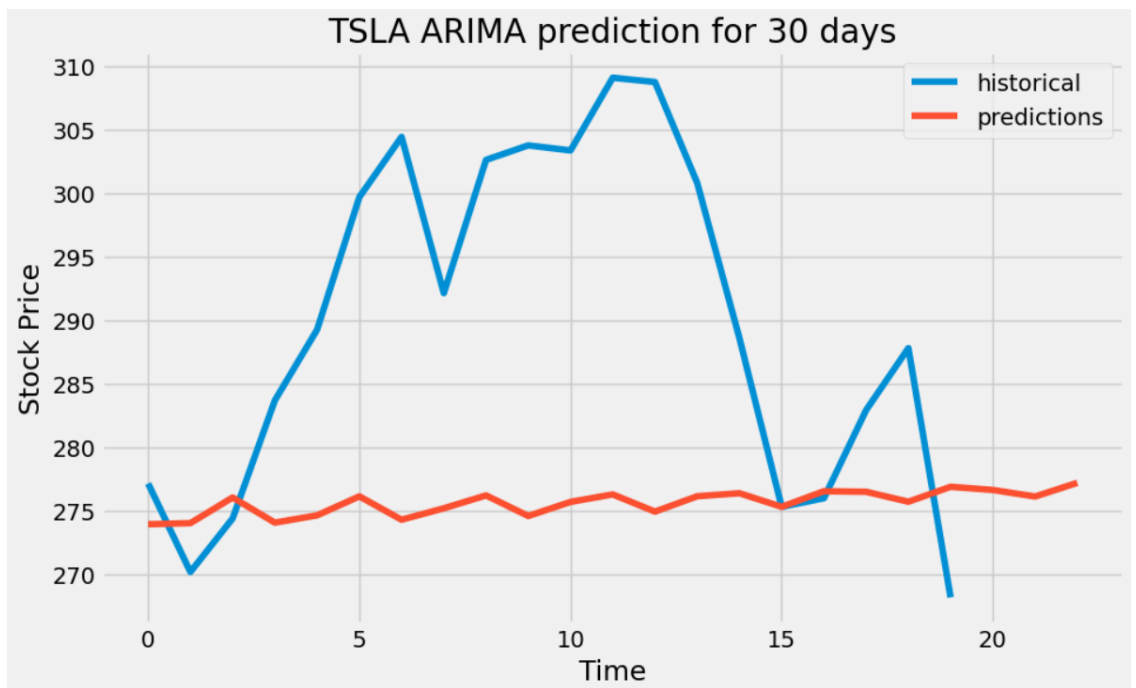


Рисунок 2.14 – Прогнозні дані вартості акцій TSLA методом ARIMA (розраховано автором на основі використаних даних)

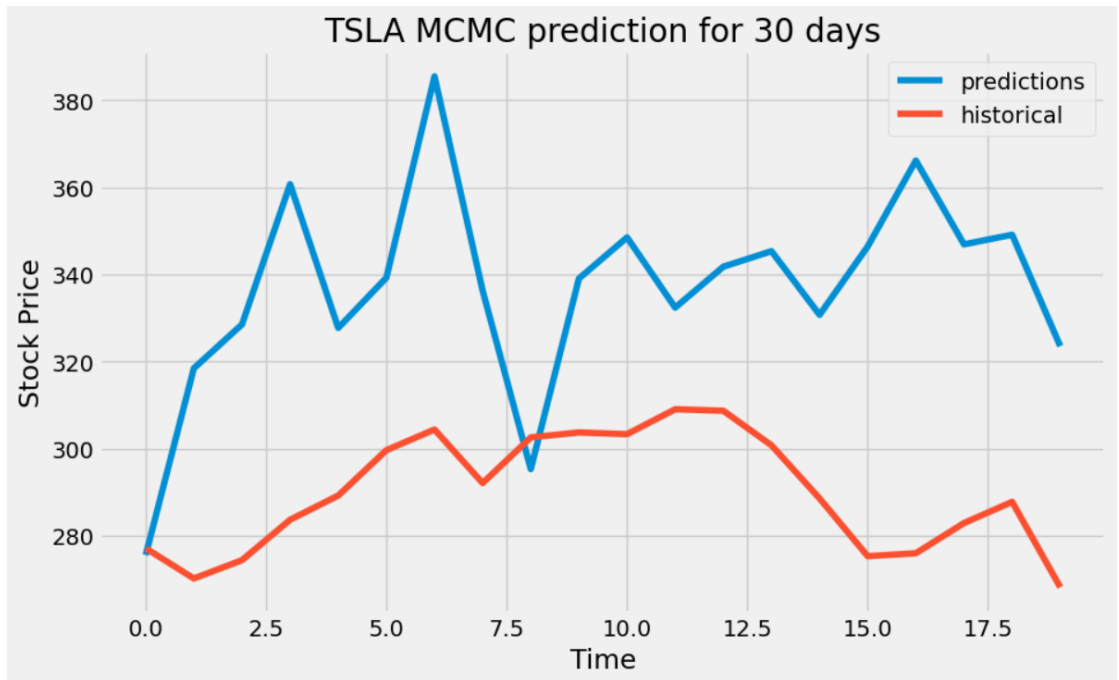


Рисунок 2.15 – Прогнозні дані вартості акцій TSLA методом MCMC  
(розраховано автором на основі використаних даних)

Test RMSE: 9.46412361130475

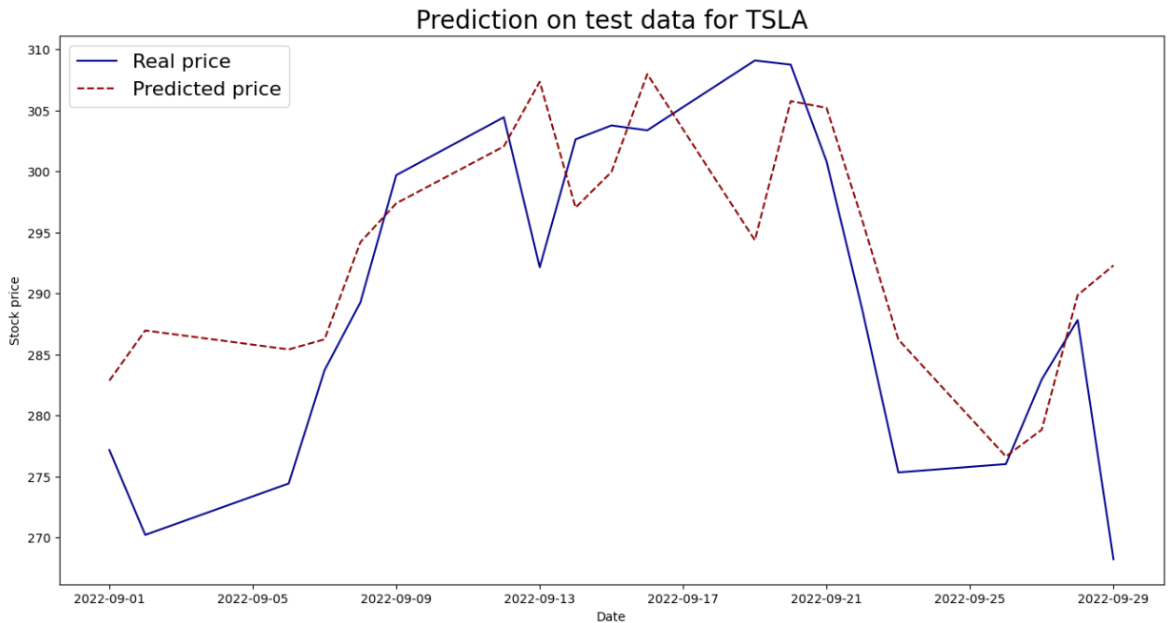


Рисунок 2.16 – Прогнозні дані вартості акцій TSLA методом GAN  
(розраховано автором на основі використаних даних)

Порівняння результатів роботи методів згаданих вище наведено на  
рис. 2.17:

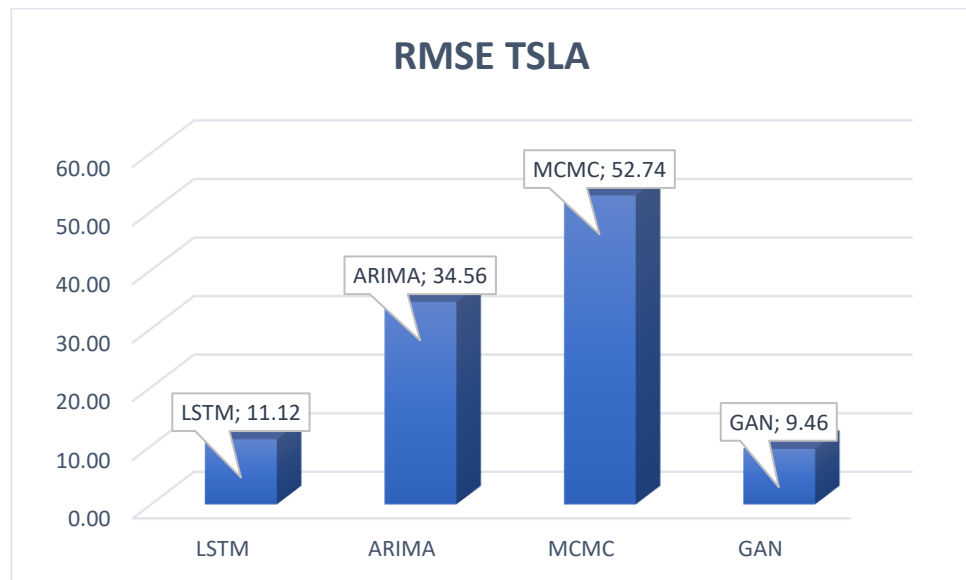


Рисунок 2.17 – Порівняння похибок прогнозів за різними методами  
(розраховано автором на основі використаних даних)

З даних розрахунків можемо побачити що генеративні нейронні мережі є найбільш ефективними, оскільки мають найнижчу похибку від реальних даних, тому вони будуть використані далі під час формування та керування інвестиційним портфелем.

## Висновки до розділу 2.

В даному розділі було сформульовано економічну задачу, ціллю якої є розробка програмного продукту, який розраховує інвестиційну стратегію на наступний місяць, беручи за основу прогнозні дані, які були отримані з моделі, яку ми створили для цієї задачі. Для створення даної моделі було розглянуто різні моделі, які зазвичай використовуються для моделювання руху часових рядів.

Часовий ряд – це послідовний набір точок даних, що вимірюється, як правило, протягом послідовних часів. Це математично визначається як набір векторів  $x(t)$ ,  $t = 0, 1, \dots$  де  $t$  являє собою час, що минув. Змінна  $x(t)$

розглядається як випадкова величина. Вимірювання, проведені під час події в часовому ряді, розташовані у відповідному хронологічному порядку.

Основними методами прогнозування часових рядів і цін акцій є наступні: метод Бокса-Дженкінса (ARIMA), метод Метрополіса-Хастінгса (Монте-Карло з марківськими ланцюгами), нейронні мережі (рекурентні, серед яких LSTM) та багато інших. Також було розглянуто генеративні змагальні мережі GAN, які містять в свої структурі LSTM і тільки набирають популярність для прогнозування часових рядів. Дані методи було детально розглянуто та апробовано на історичних даних фондових активів і у висновку можемо побачити, що мережі GAN є найбільш успішними для даного завдання, тому в фінальній моделі будуть використані саме вони. Причиною цього є той факт, що структура GAN є більш складною ніж за звичайні нейронні мережі, а змагальна ідея в цій схемі дає більше покращення в результатах.

## 3 ОЦІНКА РЕЗУЛЬТАТІВ АЛГОРИТМУ МОДЕЛЮВАННЯ ІНВЕСТИЦІЙНОГО ПОРТФЕЛЮ

### 3.1 Підготовка даних для тренування моделі

Спочатку протестуємо модель прогнозування на оригінальних даних без попередньої їх обробки. Після 200 пробних ітерацій тренування отримуємо такі результати (рис. 3.1):

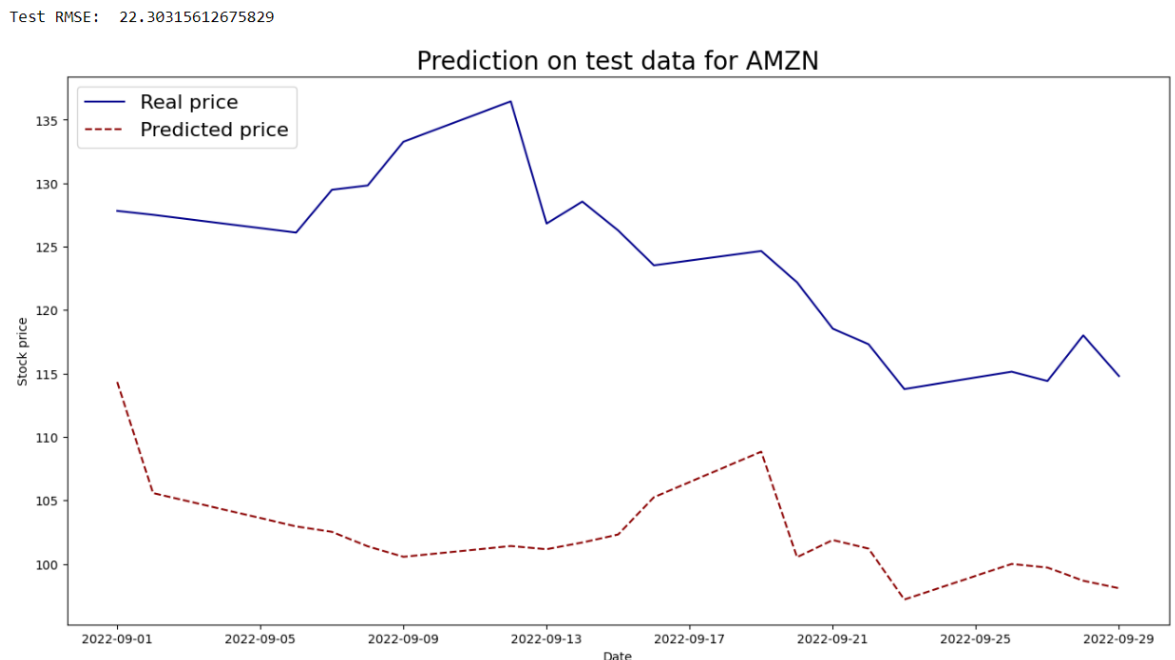


Рисунок 3.1 – Прогноз ціни AMZN на оригінальних даних  
(розраховано автором на основі використаних даних)

Можна зробити висновок, що похибка прогнозування досить висока. Це може бути спричинено багатьма причинами, але варто виділити дві основні: недостатня кількість ітерацій тренування; недостатній обсяг незалежних показників – оригінальні дані містять інформацію про вартість акцій в різні періоди та обсяг їх продажу, що обмежує картинку подій в межах одного трейдового дня.

Для того, щоб покращити якість прогнозування моделі часто використовується метод *feature engineering* (керування змінними), який містить в собі обробку, модифікацію існуючих змінних та створення нових.

Найпоширенішими показниками у фондовому ринку є такі технічні індикатори як ковзна середня за останній тиждень або 20 днів, експоненційна ковзна середня, а також лінії Боллінджера.

Перед запуском процесу тренування моделі необхідно попередньо підготувати дані, щоб прискорити та облегшити роботу алгоритму. Для цього ми нормалізуємо дані, тобто переводимо їх у діапазон від 0 до 1. Далі розглянемо незалежні параметри в даних – оригінальні дані самі по собі не дають повну картину розвитку ціни закриття, оскільки в процесі формування *close price* існує багато зовнішніх факторів. Але ми можемо доповнити їх список, додавши фінансові технічні індикатори за ціною активу *Close*, тобто додати більше характеристик самої вартості закриття для кращого визначення тренду [40–41].

В даному випадку ми розглянемо чотири індикатори –  $MA(7)$ ,  $MA(20)$ ,  $EMA$  та лінії Боллінджера.  $MA(n)$ , або проста ковзна середня ( $SMA$ ) визначається як сума цін закриття активу (*close*) за кожний день на протязі певного періоду  $n$  (наприклад, за 7 або 20 днів, як ми беремо в наших параметрах), поділена на число періодів (формула 3.1):

$$MA(n) = \frac{\sum(close, n)}{n}. \quad (3.1)$$

Смуги Боллінджера ( $BB$ ) були створені на початку 1980-х років фінансовим аналітиком і трейдером Джоном Боллінджером. Вони широко використовуються як інструмент для технічного аналізу ( $TA$ ). По суті, смуги Боллінджера працюють як вимірник осцилятора. Він вказує на високу чи низьку волатильність ринку, а також умови перекупленості чи перепроданості.

Основна ідея індикатора ВВ полягає в тому, щоб підкреслити, як ціни розсіюються навколо середнього значення. Точніше, він складається з верхньої смуги, нижньої смуги та середньої лінії ковзного середнього (також відомої як середня смуга). Дві бічні смуги реагують на ринкову цінову дію, розширюючись, коли волатильність висока (віддаляючись від середньої лінії), і стискаючись, коли волатильність низька (рухається до середньої лінії) [52–54].

Стандартна формула смуг Боллінджера встановлює середню лінію як 20-денне просте ковзне середнє (*SMA*), тоді як верхня та нижня смуги розраховуються на основі волатильності ринку по відношенню до *MA* (яка називається стандартним відхиленням). Стандартні налаштування індикатора смуг Боллінджера виглядатимуть так:

- Середня лінія:  $MA(20)$ ;
- Верхня смуга:  $MA(20) + (2 * \text{стандартне відхилення}(MA(20)))$ ;
- Нижня смуга:  $MA(20) - (2 * \text{стандартне відхилення}(MA(20)))$ .

Індикатор враховує 20-денний період і визначає верхню та нижню межі на два стандартних відхилення від середньої лінії. Це робиться для того, щоб принаймні 85% цінових даних переміщалися між цими двома діапазонами, але параметри можна регулювати відповідно до різних потреб і трейдингових стратегій [48-49].

Експоненційна ковзна середня (*EMA*) розраховується шляхом додавання до попереднього значення середньої частки ціни закриття на момент розрахунку. Таким чином, найбільш питома вага при розрахунку цього показника надається останньому значенню ціни на графіку (формула 3.2):

$$EMA_t = (close * P) + (EMA_{t-1} * (100 - P)), \quad (3.2)$$

де *close* – ціна закриття в момент *t*;

$EMA_{t-1}$  – значення *EMA* за попередній період;

*P* – частка від значення цін.

Нові розраховані показники додаються до списку параметрів, які подаються в модель на тренування як незалежні ( $X$ ) [50].

Отримані дані та сам рух ціни закриття активу виглядають так (рис. 3.2):

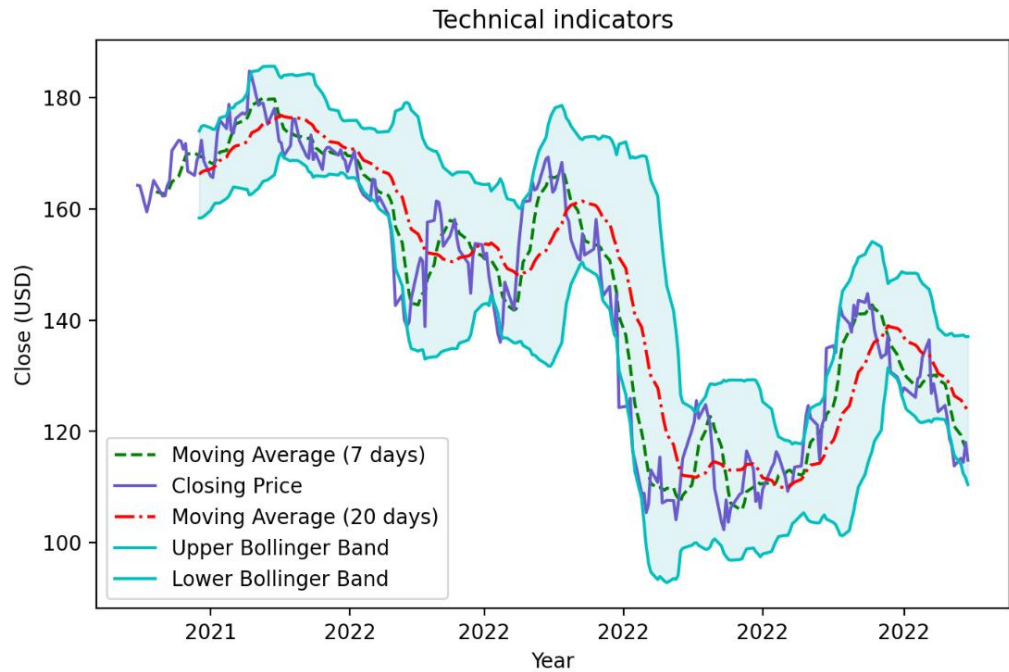


Рисунок 3.2 – Історичні дані ціни AMZN та їх технічні індикатори  
(розраховано автором на основі використаних даних)

Тепер подамо модифіковані дані на тренування. Параметри моделі та її тренування не змінені. Отримані прогнозні дані наведені на рис. 3.3:

Test RMSE: 8.497461431854552

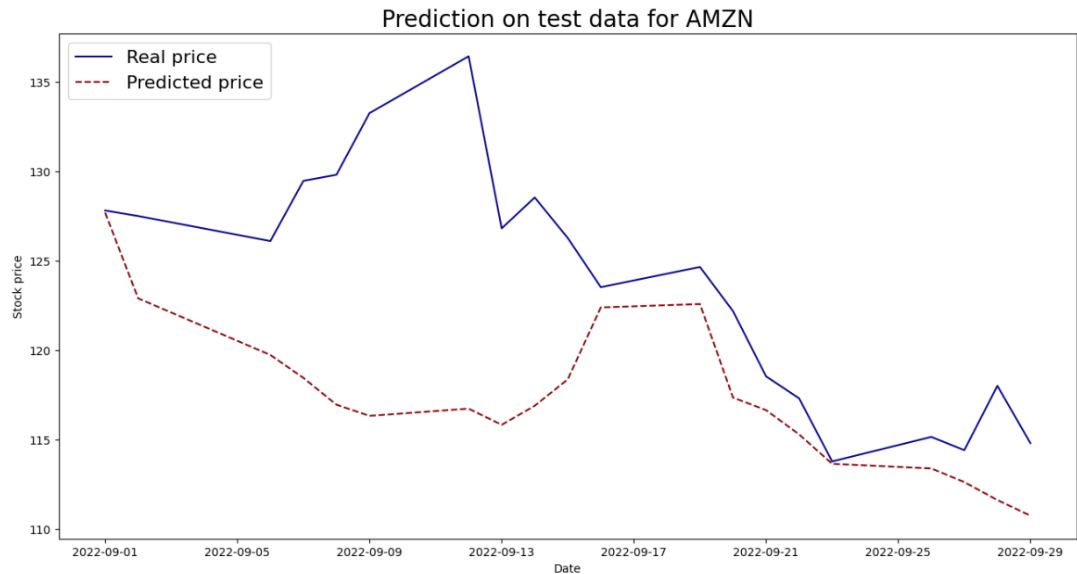


Рисунок 3.3 – Прогноз ціни AMZN на нових даних  
(розраховано автором на основі використаних даних)

З цього можна зробити висновок, що додавання нових змінних допомогло моделі покращити якість прогнозування даних – середня квадратична похибка зменшилась з 22.3 до 8.4, майже втричі. Це спричинено тим, що модель тепер має інформацію не тільки про чисту вартість акцій та обсяг продажу за один день, а також про зміну більш глобальних показників, які дають картину руху вартості фондового активу за певний минулий проміжок часу.

Тепер постає питання, як можливо далі покращити якість прогнозування руху фондового активу. Найбільшою поширеною проблемою у подібних дослідженнях полягає у тому, що технічні індикатори та чиста вартість активу все ще не дають повну картину розвитку фондового ринку. Із розвитком технологій та збільшенням швидкості поширення інформації ринок стає більш залежним від таких зовнішніх факторів, як новини, соціальний імідж, репутація. За останній рік така тенденція особливо виражена – в день початку повномасштабного вторгнення країни-терориста росії в Україну в лютому 2022 року весь фондовий ринок «обвалився», тобто, ціни на всі фондові акції різко впали.

Останні роки все більше і більше бачимо тенденцію, що ринок реагує на новину про можливість певної події, а не саму фактичну її реалізацію. Це спричинено тим, що стан ринку сильно залежить від її учасників, точніше, загального їх настрою, тобто, сентименту.

Деякі задокументовані випадки показали, як негативні неправдиві повідомлення щодо компаній сильно вплинули на ціни їхніх акцій протягом короткого періоду часу та як ціна акцій відновила частину своєї вартості, коли виявилось, що новини були фальшивими.

Перший приклад: 14 червня 2021 року Кріштіану Роналду на прес-конференції заявив: «Agua, no Sosa»; ця заява була піддана жорсткій критиці, оскільки того дня акції Соса-Сола впали на 1,06% порівняно з попередньою ціною закриття. Аналіз часових рамок поведінки акцій показав, що сама по собі декларація не вплинула на результати діяльності компанії. Кріштіану Роналду зробив заяву о 9:43 EST, і ціна акцій впала до \$55,26 до 9:40 EST, за 3 хвилини до заяви Роналду. Більше того, акції закрилися на 0,30 долара вище 55,26 долара до кінця торгового дня. Що більше вплинуло на ціну, так це заголовок «Кріштіану Роналду вилучає пляшки з кока-колою та падіння цін на акції Соса-Сола» CNN Іспанія 16 червня, який вплинув на ціну 18 і 23 числа того ж місяця [55, 56].

Дослідники численних публікацій прагнули побудувати моделі для прогнозування цін на акції, оскільки традиційні моделі не повністю успішні для цього завдання. На сучасних ринках думки акціонерів вважаються вірними показниками майбутньої вартості їхніх інвестиційних пакетів. З поширеним використанням соціальних мереж думка акціонерів стала більш актуальною, ніж будь-коли раніше. Оцінка думок соціальних мереж у поєднанні з традиційними моделями прогнозування значно підвищив рівень успішності методологій прогнозування.

Кілька опублікованих досліджень описують нові моделі для прогнозування цін на акції, переважно з використанням суспільних думок. Хоча автори деяких досліджень досягли певної здатності прогнозувати, деякі

інші дійшли висновку, що соціальні настрої не є корисними для прогнозування курсу акцій. Аналіз настроїв у соціальних мережах також використовувався для вивчення впливу потоків новин на ціну криптовалют. З огляду на те, що ця категорія активів ще не завоювала довіри інвесторів, її ціна більш сприйнятлива до волатильності, а кореляція між новинними релізами та ціною поведінкою більш виражена. Проте всі ці дослідження були зосереджені на прогнозуванні руху фондового ринку чи цін на окремі акції, а не на визначенні величини впливу поганих новин на курс акцій порівняно з хорошими новинами для конкретних компаній [37, 57].

Для того, щоб оцінити загальний настрій ринку, ми візьмемо за основу соціальну мережу Twitter, оскільки в ній найшвидше розповсюджуються новини.

Twitter спостерігав вибухове зростання за останні кілька років. Зараз зареєстровано до 300 мільйонів користувачів, Twitter відвідує 50 мільйонів користувачів на день і 400 мільйонів відвідувачів на місяць. Twitter є багатим джерелом інформації в реальному часі про поточні суспільні тенденції та думки.

Дані Twitter є рішенням проблеми, яка існує у прогнозуванні цін. Фактори, які обговорювалися вище, такі як економічні обставини, політичні події та інші екологічні фактори, які можуть вплинути на курс акцій, доступні в даних Twitter.

Поведінкова економіка говорить нам, що люди не є раціональними споживачами, а на індивідуальну поведінку та рішення значною мірою впливають емоції – і навіть думка інших. Сентиментальний аналіз Twitter може бути надзвичайно корисним для прогнозування емоцій або думки про певний продукт. Таким чином, досліджуючи прогнозний потенціал Twitter щодо купівлі та думки споживачів, спостерігаючи за зв'язком між суспільними тенденціями Twitter у технологічному секторі та погодинними курсами акцій компаній, що найбільше здобули та програли, компаній у технологічному секторі [38, 58].

З сайту ми зберемо сет даних з твітами(дописами) стосовно акцій, що розглядається, та використаємо метод sentiment analysis (оцінка настрою) для відношення людей до даного активу в розглядаємий проміжок часу.

Аналіз настрою – це метод аналізу тексту, який виявляє полярність (наприклад, позитивну чи негативну думку) у тексті, будь то весь документ, абзац, речення чи пункт. Аналіз настроїв має на меті виміряти ставлення, почуття, оцінки, ставлення та емоції оратора/письма на основі обчислювальної обробки суб'єктивності в тексті.

В даній роботі для оцінки настрою тексту ми використаємо модель VADER. VADER (Valence Aware Dictionary for Sentiment Reasoning) – це модель, що використовується для аналізу настроїв тексту, який чутливий як до полярності (позитивний/негативний), так і до інтенсивності (сили) емоцій. Він доступний у пакеті NLTK і може бути застосований безпосередньо до текстових даних без міток.

Сентиментальний аналіз VADER покладається на словник, який зіставляє лексичні особливості з інтенсивністю емоцій, відомі як оцінки сентиментів. Оцінку настрою тексту можна отримати шляхом підсумовування інтенсивності кожного слова в тексті.

Наприклад, такі слова, як «люблю», «насолоджуюся», «щасливий», «подобається», передають позитивні настрої. Крім того, VADER достатньо розумний, щоб зрозуміти основний контекст цих слів, таких як «не любив» як негативне твердження. Він також розуміє наголошення великих літер і пунктуації, як-от «ЛЮБЛЮ».

Аналіз настрою VADER повертає оцінку настрою в діапазоні від -1 до 1, від найбільш негативного до найбільш позитивного. Оцінка настрою речення обчислюється шляхом підсумовування оцінок настрою кожного слова в реченні, зазначеного у словнику VADER. Звичайно, існує протиріччя: окремі слова мають оцінку настрою від -4 до 4, але повернута оцінка настрою речення становить від -1 до 1. Даний показник називається в наукових роботах як полярність (polarity score). Якщо показник полярності становить

«1», значить текст має позитивний настрій, «-1» – негативний, «0» – нейтральний [36, 47].

Для прикладу зберемо усі дописи в мережі Twitter про акцію AMZN за останній рік (30.09.2021–30.09.2022). Оскільки в соціальних мережах є багато повторених дописів, або дописів без сенсу, то ми відфільтруємо тільки ті, які мають більше 100 вподобань. В результаті ми отримаємо більше 4000 твітів за розглянутий період часу, що буде в середньому 10 твітів в день. Це не є найбільшим показником, оскільки Amazon не є найбільш обговорюваною компанією в даній соціальній мережі як, наприклад, Tesla, власник якої нещодавно придбав Twitter. Отримаємо датасет такого вигляду (рис.3.4):

	Date	Tweet
0	2022-09-29 22:40:47+00:00	A group of lawmakers led by Sen. Elizabeth War...
1	2022-09-29 22:23:54+00:00	\$NIO just because I'm down money doesn't mean ...
2	2022-09-29 18:34:51+00:00	Today's drop in \$SPX is a perfect example of w...
3	2022-09-29 15:57:59+00:00	Druckenmiller owned \$CVNA this year \nMunger b...
4	2022-09-29 15:10:30+00:00	Top 10 \$QQQ Holdings \n\nAnd Credit Rating\n\n...
...	...	...
4084	2021-09-30 20:50:16+00:00	My top 10 holdings make up 66% of my portfolio...
4085	2021-09-30 19:45:13+00:00	Amazon \$AMZN has 4X the market cap of Alibaba ...
4086	2021-09-30 18:01:19+00:00	Returns over the last 10 Years:\n\nBitcoin \$BT...
4087	2021-09-30 17:56:14+00:00	Great snapshot of what it looks like when an i...
4088	2021-09-30 01:59:22+00:00	Shopify is now 9% of US e-commerce. It really ...

4089 rows × 2 columns

Рисунок 3.4 – дані з дописами про акцію AMZN за минулий рік  
(розраховано автором на основі використаних даних)

Далі додамо 4 показники в дані – Positive, Negative, Neutral та Sentiment Score. Після використання VADER в ці колонки буде додано показник відповідного настрою за шкалою від -1 до 1, де 1 – дуже ймовірний настрій, а -1 – дуже неймовірний настрій.

Далі розраховується фінальна комплексна оцінка (Compound Score). Комплексна оцінка – це сума позитивних, негативних та нейтральних оцінок, яка потім нормалізується в діапазоні від -1 (крайня негативна оцінка) до +1 (крайня позитивна оцінка). Чим більше комплексна оцінка наближається до +1, тим вищий рівень позитивності тексту. Цей фінальний показник подається в колонку «Sentiment Score» [47, 59].

### **3.2 Створення програмного додатку для оцінки адекватності моделі та розв’язку економічної задачі**

Розрахунки були виконані у розробленому програмному додатку на мові програмування Python у середовищі Jupyter Notebook (точні результати прогнозування даних наведені у Додатку 1). На вхід подаються дані про початковий обсяг грошового фонду портфелю та типи акцій, що розглядаються. На вихід алгоритм подає сформований початковий портфель та подальшу стратегію керування ним.

Розглянемо типи фондових акцій, що були використані у даній роботі:

– AMZN

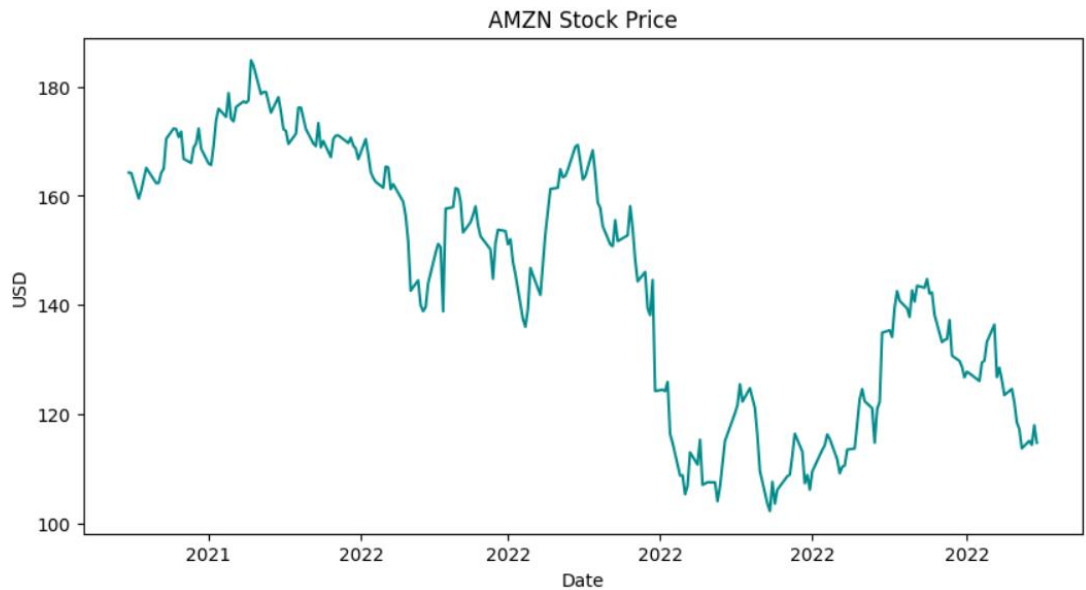


Рисунок 3.5 – Історичні дані вартості акцій AMZN за минулий рік  
(розраховано автором на основі використаних даних)

Після розрахунку технічних індикаторів отримали такі дані AMZN  
(рис. 3.6):

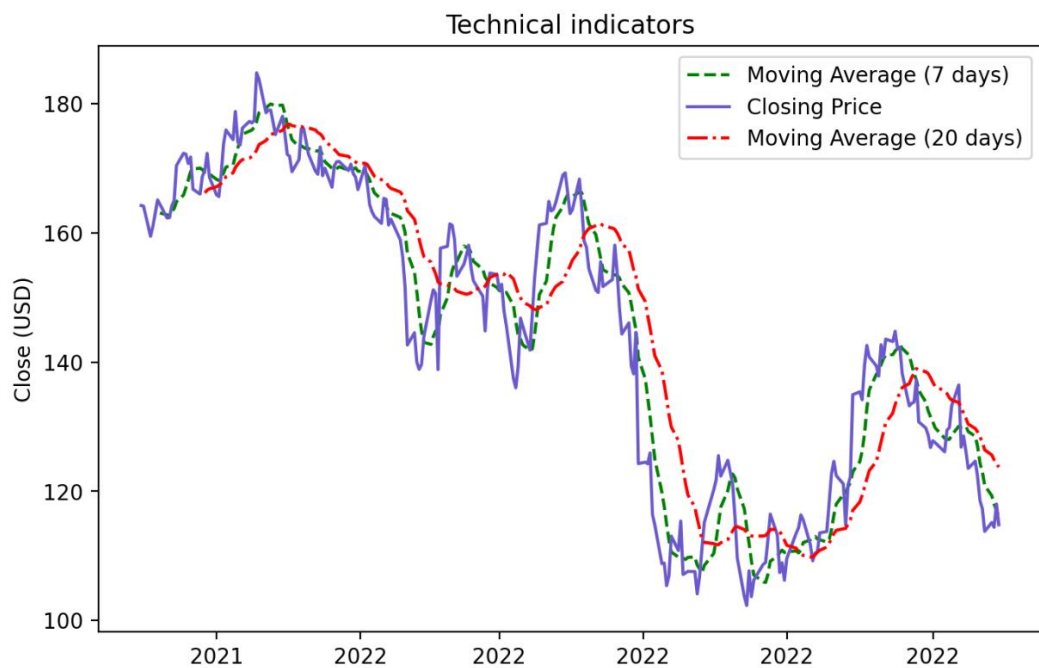


Рисунок 3.6 – Технічні індикатори AMZN  
(розраховано автором на основі використаних даних)

Після тренування моделі отримали такі прогнози від GAN (рис. 3.7):

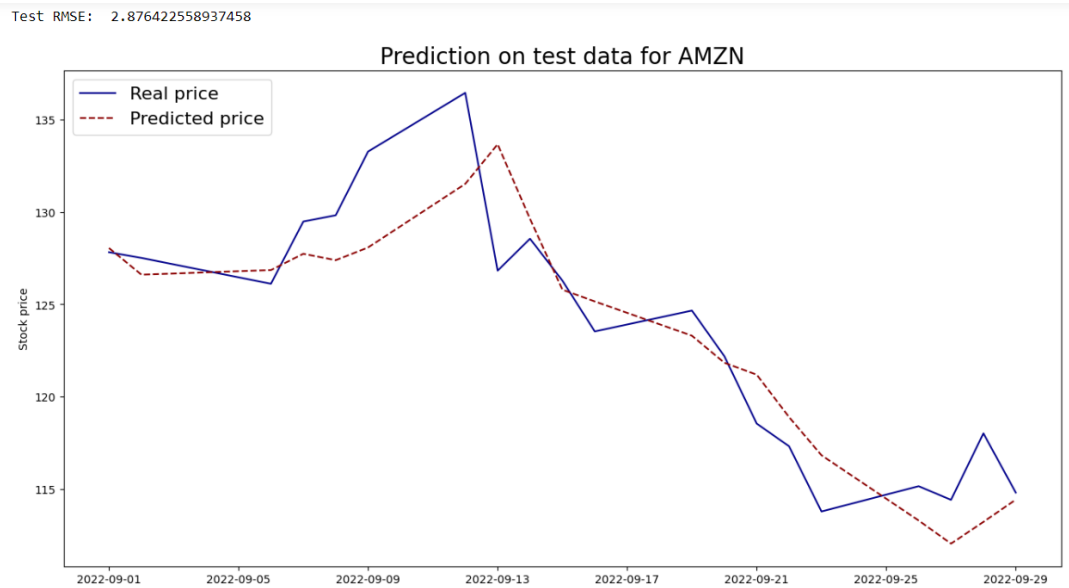


Рисунок 3.7 – Фінальні прогнозні дані акцій AMZN  
(розраховано автором на основі використаних даних)

– FB

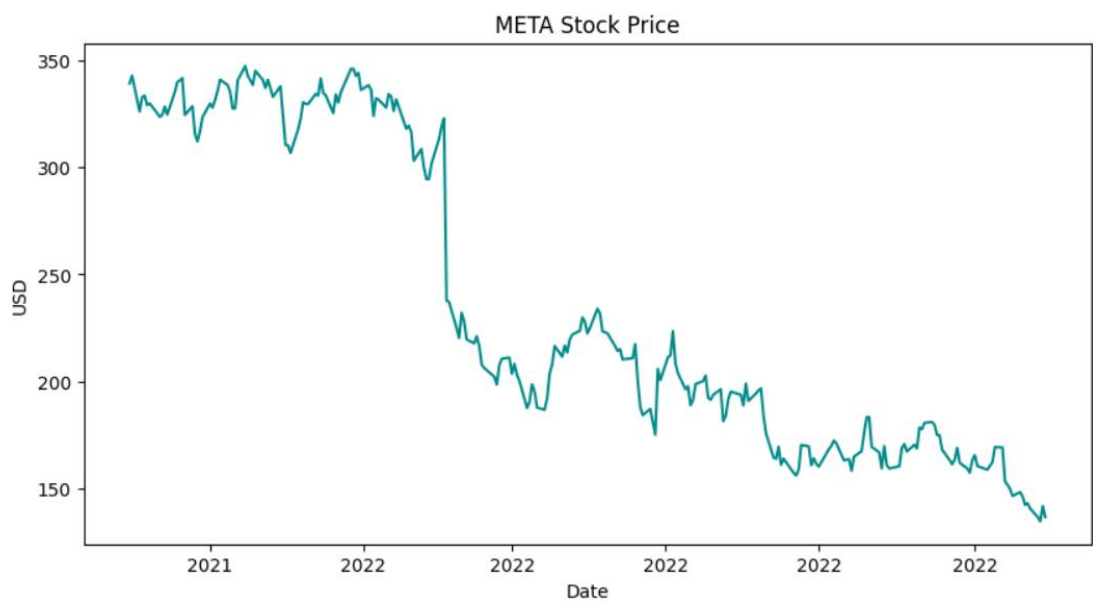


Рисунок 3.8 – Історичні дані вартості акцій FB (META) за минулий рік  
(розраховано автором на основі використаних даних)

Після розрахунку технічних індикаторів отримали такі дані META  
(рис. 3.9):

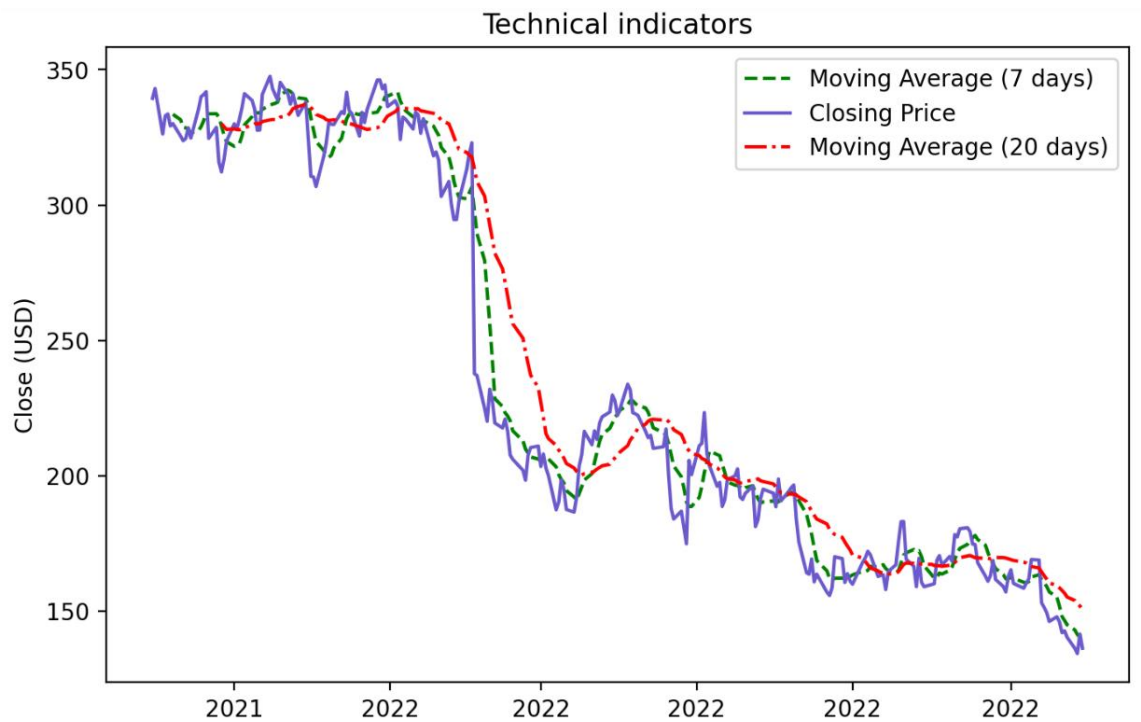


Рисунок 3.9 – Технічні індикатори FB (META)

(розраховано автором на основі використаних даних)

Після тренування моделі отримали такі прогнози від GAN (рис. 3.10):

Test RMSE: 20.946868463825147

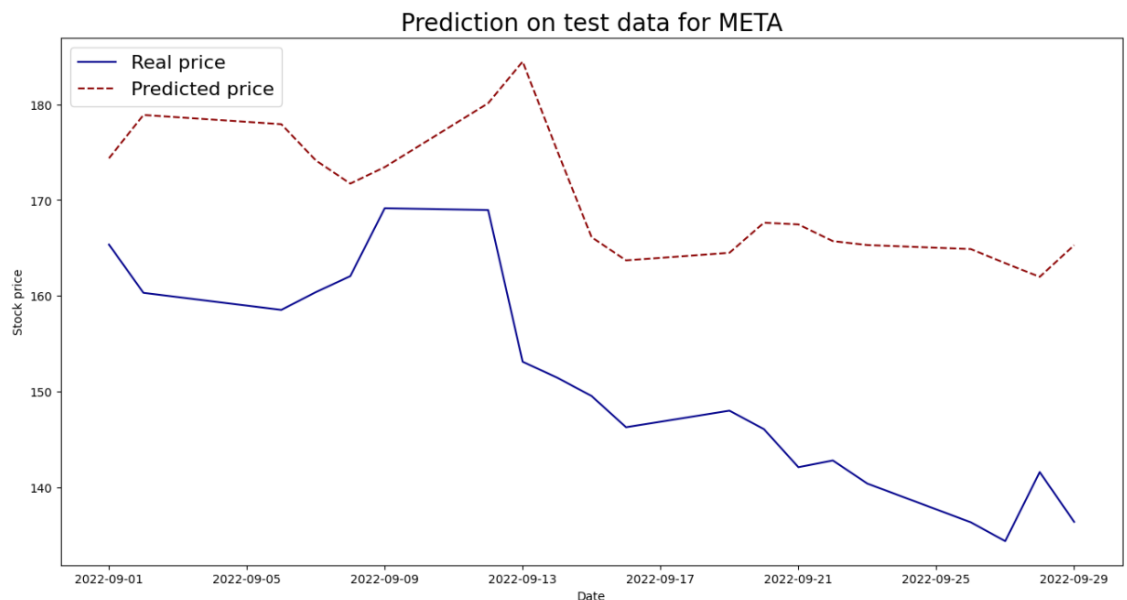


Рисунок 3.10 – Фінальні прогнозні дані акцій FB(META) на 30 днів

(розраховано автором на основі використаних даних)

– MSFT

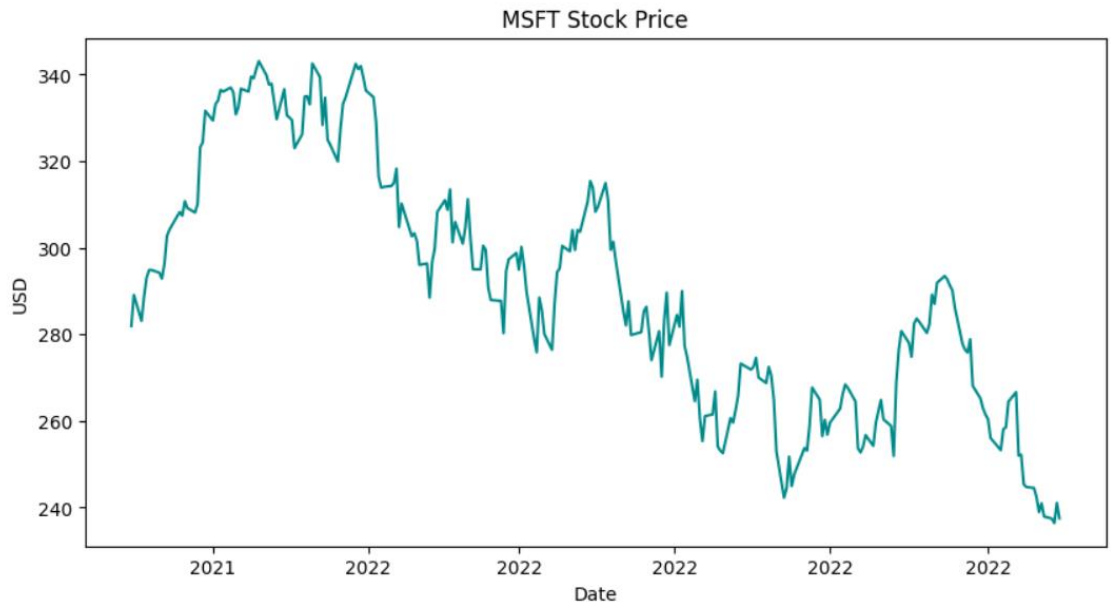


Рисунок 3.11 – Історичні дані вартості акцій MSFT за минулий рік  
(розраховано автором на основі використаних даних)

Після розрахунку технічних індикаторів отримали такі дані MSFT  
(рис. 3.12):

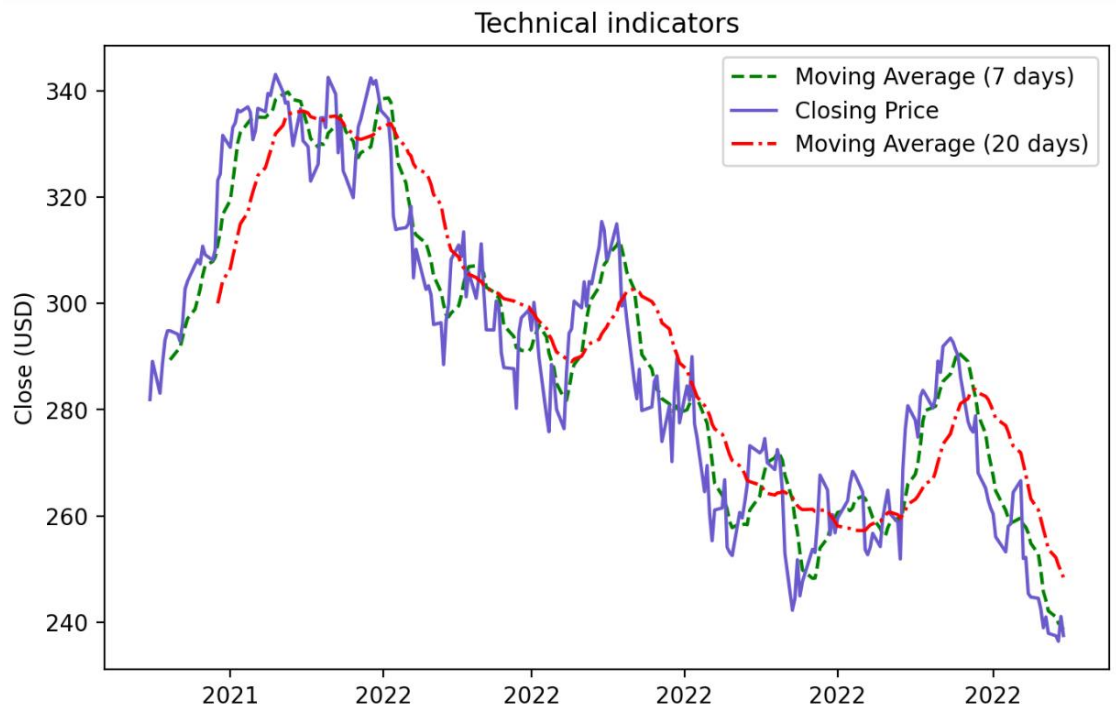


Рисунок 3.12 – Технічні індикатори MSFT  
(розраховано автором на основі використаних даних)

Після тренування моделі отримали такі прогнози від GAN (рис. 3.13):

Test RMSE: 8.760897473839607

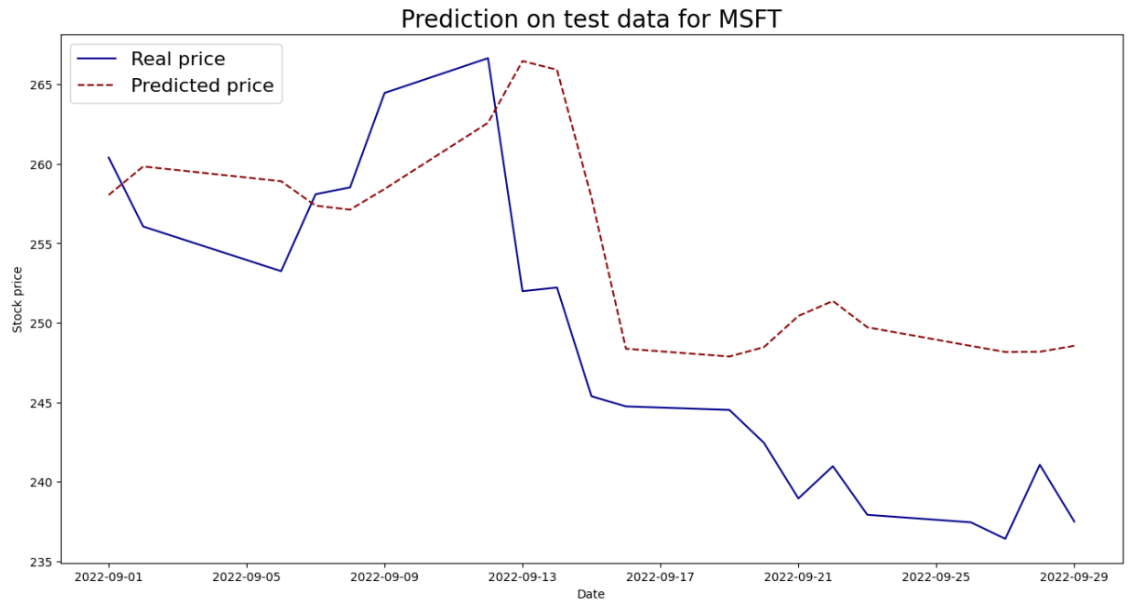


Рисунок 3.13 – Фінальні прогнозні дані акцій MSFT на 30 днів  
(розраховано автором на основі використаних даних)

В результаті виконання алгоритму отримуємо значення вагів для початкового портфелю, за якими можна розрахувати скільки активів необхідно спочатку закупити, а коефіцієнт Шарпа=3,32% (рис. 3.14–3.15):

```

AMZN bought: 57600.0$
META bought: 60000.0$
MSFT bought: 2400.0$
weights by markowitz: [0.48 0.5 0.02]
sharpe ratio by markowitz = 0.0921910304094797%

```

Рисунок 3.14 – Результати алгоритму з методом Марковіца  
(розраховано автором на основі використаних даних)

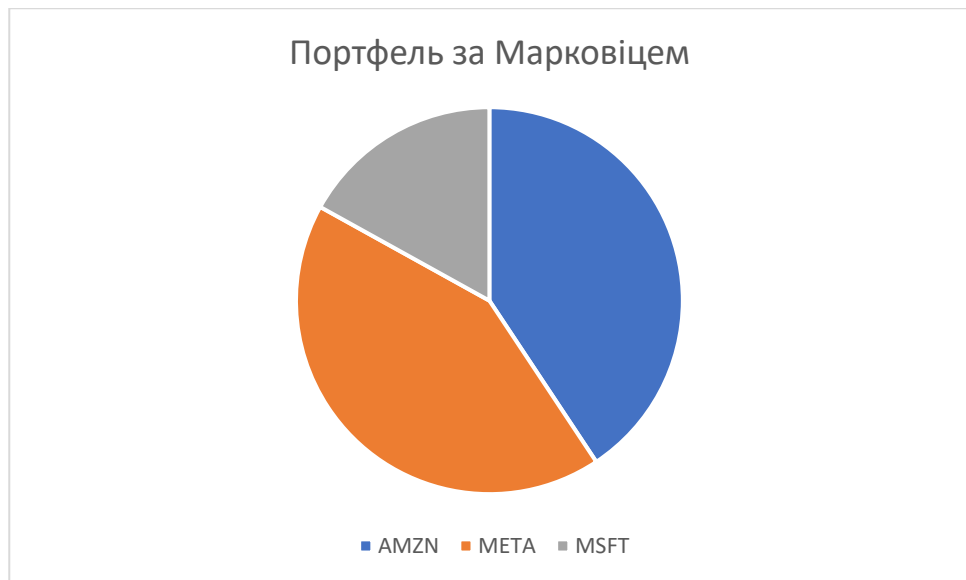


Рисунок 3.15 – Ваги за методом Марковіца  
(розраховано автором на основі використаних даних)

Далі після калібрування вагів на основі прогнозованих даних в останньому часовому періоді отримали такий ваговий вектор з очікуваною дохідністю = 3% (табл. 3.1, рис. 3.16):

Таблиця 3.1 – Ваги активів за методом Хуанга-Літценбергера

Акція	Частка
AMZN	40%
FB	12,5%
MSFT	47,5%

*Розраховано автором*

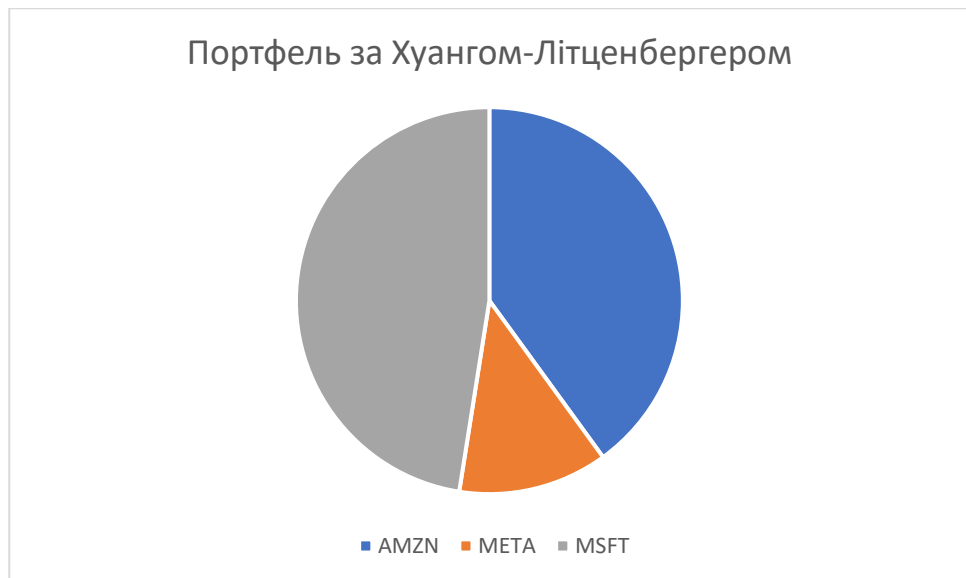


Рисунок 3.16 – Ваги за методом Хуанга-Літценбергера  
(розраховано автором на основі використаних даних)

```

final real profit: 3627
real profit %: 1.9344000000000001%
avg h-l risk = 0.18769817223221757%
expected return = 1.5%

```

Рисунок 3.17 – Результати алгоритму Хуанга-Літценбергера  
(розраховано автором на основі використаних даних)

В результаті проведеної роботи була розроблена зведена модель управління портфелем. Він прогнозує розвиток використовуваних активів фонду та створює стратегії управління ними на основі отриманих ключових прогнозних даних. Модель також розраховує очікувані прибутки та інвестиційні стратегії у разі неочікуваних змін на ринку. Достовірність моделі перевіряється за прогнозними та фактичними історичними даними за період, що розглядається.

### 3.3. Програмна реалізація моделі прогнозування вартості акцій за допомогою мережі GAN

*Код програми моделі зроблено в середовищі Python:*

```
def gan_pred(stock_name):

    df = pd.read_csv(f'{stock_name}_tweets.csv')

    sent_df = df.copy()
    sent_df["sentiment_score"] = ''
    sent_df["Negative"] = ''
    sent_df["Neutral"] = ''
    sent_df["Positive"] = ''
    sent_df.head()

    sentiment_i_a = SentimentIntensityAnalyzer()
    for indx, row in sent_df.T.iteritems():
        try:
            sentence_i = unicodedata.normalize('NFKD',
sent_df.loc[indx, 'Tweet'])
            sentence_sentiment =
sentiment_i_a.polarity_scores(sentence_i)
            sent_df.at[indx, 'sentiment_score'] =
sentence_sentiment['compound']
            sent_df.at[indx, 'Negative'] = sentence_sentiment['neg']
            sent_df.at[indx, 'Neutral'] = sentence_sentiment['neu']
            sent_df.at[indx, 'Positive'] = sentence_sentiment['pos']
        except TypeError:
            print (sent_df.loc[indexx, 'Tweet'])
            print (indx)
            break
```

```

sent_df['Date'] = pd.to_datetime(sent_df['Date'])
sent_df['Date'] = sent_df['Date'].dt.date
sent_df = sent_df.drop(columns=['Negative', 'Positive',
'Neutral'])

twitter_df = sent_df.groupby([sent_df['Date']]).mean()

# ## Get final dataset for training

# getting data from Yahoo Finance
#stock_name = 'AMZN' # here you can change the name of stock
ticker, for example we will take AMD ticker
stock_df = yf.download(stock_name, start="2021-09-30", end="2022-
09-30")
stock_df = stock_df.reset_index(level=0)
stock_df['Date'] = stock_df['Date'].dt.date

final_df = stock_df.join(twitter_df, how="left", on="Date")

fig, ax = plt.subplots(figsize=(10,5))
ax.plot(final_df['Date'], final_df['Close'], color='#008B8B')
ax.set(xlabel="Date", ylabel="USD", title=f"{stock_name} Stock
Price")
ax.xaxis.set_major_formatter(DateFormatter("%Y"))
plt.show()

def get_tech_ind(data):
    data['MA7'] = data.iloc[:,4].rolling(window=7).mean() #Close
column
    data['MA20'] = data.iloc[:,4].rolling(window=20).mean() #Close
Column

    data['MACD'] = data.iloc[:,4].ewm(span=26).mean() -
data.iloc[:,1].ewm(span=12,adjust=False).mean()

```

```

#This is the difference of Closing price and Opening Price
# Create Bollinger Bands
data['20SD'] = data.iloc[:, 4].rolling(20).std()
data['upper_band'] = data['MA20'] + (data['20SD'] * 2)
data['lower_band'] = data['MA20'] - (data['20SD'] * 2)

# Create Exponential moving average
data['EMA'] = data.iloc[:,4].ewm(com=0.5).mean()

# Create LogMomentum
data['logmomentum'] = np.log(data.iloc[:,4] - 1)

return data

tech_df = get_tech_ind(final_df)
dataset = tech_df.iloc[20:,:].reset_index(drop=True)

def tech_ind(dataset):
    fig,ax = plt.subplots(figsize=(8, 5), dpi = 200)
    x_ = range(3, dataset.shape[0])
    x_ = list(dataset.index)

    ax.plot(dataset['Date'], dataset['MA7'], label='Moving Average
(7 days)', color='g', linestyle='--')
    ax.plot(dataset['Date'], dataset['Close'], label='Closing
Price', color='#6A5ACD')
    ax.plot(dataset['Date'], dataset['MA20'], label='Moving
Average (20 days)', color='r', linestyle='-.')
    #ax.plot(dataset['Date'], dataset['upper_band'], label='Upper
Bollinger Band', color='c')
    #ax.plot(dataset['Date'], dataset['lower_band'], label='Lower
Bollinger Band', color='c')
    #ax.fill_between(dataset['Date'], dataset['lower_band'],
dataset['upper_band'], alpha=0.35, color = '#B0E0E6')

```

```

ax.xaxis.set_major_formatter(DateFormatter("%Y"))
plt.title('Technical indicators')
plt.ylabel('Close (USD)')
plt.xlabel("Year")
plt.legend()
plt.show()

```

```
tech_ind(tech_df)
```

```
dataset.iloc[:, 1:] = pd.concat([dataset.iloc[:, 1:].ffill()])
```

```

datetime_series = pd.to_datetime(dataset['Date'])
datetime_index = pd.DatetimeIndex(datetime_series.values)
dataset = dataset.set_index(datetime_index)
dataset = dataset.sort_values(by='Date')
dataset = dataset.drop(columns='Date')

```

```
def normalize_data(df, range, target_column):
```

```
    ...
```

```
    df: dataframe object
```

```
    range: type tuple -> (lower_bound, upper_bound)
```

```
        lower_bound: int
```

```
        upper_bound: int
```

```
    target_column: type str -> should reflect closing price of
```

```
stock
```

```
    ...
```

```
    target_df_series = pd.DataFrame(df[target_column])
```

```
    data = pd.DataFrame(df.iloc[:, :])
```

```
    X_scaler = MinMaxScaler(feature_range=range)
```

```
    y_scaler = MinMaxScaler(feature_range=range)
```

```
    X_scaler.fit(data)
```

```

y_scaler.fit(target_df_series)
X_scale_dataset = X_scaler.fit_transform(data)
y_scale_dataset = y_scaler.fit_transform(target_df_series)

dump(X_scaler, open('X_scaler.pkl', 'wb'))
dump(y_scaler, open('y_scaler.pkl', 'wb'))

return (X_scale_dataset,y_scale_dataset)

def batch_data(x_data,y_data, batch_size, predict_period):
    X_batched, y_batched, yc = list(), list(), list()

    for i in range(0,len(x_data),1):
        x_value = x_data[i: i + batch_size][:, :]
        y_value = y_data[i + batch_size: i + batch_size +
predict_period][:, 0]
        yc_value = y_data[i: i + batch_size][:, :]
        if len(x_value) == batch_size and len(y_value) ==
predict_period:
            X_batched.append(x_value)
            y_batched.append(y_value)
            yc.append(yc_value)

    return np.array(X_batched), np.array(y_batched), np.array(yc)

def split_train_test(data, split_ratio = 0.8):
    #train_size = round(len(data) * split_ratio)
    train_size = len(data) - 20
    data_train = data[0:train_size]
    data_test = data[train_size:]
    return data_train, data_test

def predict_index(dataset, X_train, batch_size,
prediction_period):

```

```

        # get the predict data (remove the in_steps days)
        train_predict_index = dataset.iloc[batch_size:
X_train.shape[0] + batch_size + prediction_period, :].index
        test_predict_index = dataset.iloc[X_train.shape[0] +
batch_size:, :].index

    return train_predict_index, test_predict_index

X_scale_dataset,y_scale_dataset = normalize_data(dataset, (-1,1),
"Close")
X_batched, y_batched, yc = batch_data(X_scale_dataset,
y_scale_dataset, batch_size = 5, predict_period = 1)
print("X shape:", X_batched.shape)
print("y shape:", y_batched.shape)
print("yc shape:", yc.shape)

X_train, X_test, = split_train_test(X_batched)
y_train, y_test, = split_train_test(y_batched)
yc_train, yc_test, = split_train_test(yc)
index_train, index_test, = predict_index(dataset, X_train, 5, 1)

input_dim = X_train.shape[1]
feature_size = X_train.shape[2]
output_dim = y_train.shape[1]

# ## Build GAN model

def make_generator_model(input_dim, output_dim, feature_size):
    model = tf.keras.Sequential([LSTM(units = 1024,
return_sequences = True,
                                     input_shape=(input_dim,
feature_size),recurrent_dropout = 0.3),

```

```

        LSTM(units = 512, return_sequences
= True, recurrent_dropout = 0.3),
        LSTM(units = 256, return_sequences
= True, recurrent_dropout = 0.3),
        LSTM(units = 128, return_sequences
= True, recurrent_dropout = 0.3),
        LSTM(units = 64, recurrent_dropout
= 0.3),

        Dense(32),
        Dense(16),
        Dense(8),
        Dense(units=output_dim)])

    return model

```

```

def make_discriminator_model(input_dim):
    cnn_net = tf.keras.Sequential()
    cnn_net.add(Conv1D(8, input_shape=(input_dim+1, 1),
kernel_size=3, strides=2, padding='same',
activation=LeakyReLU(alpha=0.01)))
    cnn_net.add(Conv1D(16, kernel_size=3, strides=2,
padding='same', activation=LeakyReLU(alpha=0.01)))
    cnn_net.add(Conv1D(32, kernel_size=3, strides=2,
padding='same', activation=LeakyReLU(alpha=0.01)))
    cnn_net.add(Conv1D(64, kernel_size=3, strides=2,
padding='same', activation=LeakyReLU(alpha=0.01)))
    cnn_net.add(Conv1D(128, kernel_size=1, strides=2,
padding='same', activation=LeakyReLU(alpha=0.01)))
    #cnn_net.add(Flatten())
    cnn_net.add(LeakyReLU())
    cnn_net.add(Dense(220, use_bias=False))
    cnn_net.add(LeakyReLU())
    cnn_net.add(Dense(220, use_bias=False, activation='relu'))
    cnn_net.add(Dense(1, activation='sigmoid'))
    return cnn_net

```

```

def discriminator_loss(real_output, fake_output):
    loss_f = tf.keras.losses.BinaryCrossentropy(from_logits=True)
    real_loss = loss_f(tf.ones_like(real_output), real_output)
    fake_loss = loss_f(tf.zeros_like(fake_output), fake_output)
    total_loss = real_loss + fake_loss
    return total_loss

def generator_loss(fake_output):
    loss_f = tf.keras.losses.BinaryCrossentropy(from_logits=True)
    loss = loss_f(tf.ones_like(fake_output), fake_output)
    return loss

@tf.function

def train_step(real_x, real_y, yc, generator, discriminator,
g_optimizer, d_optimizer):
    #@tf.function
    with tf.GradientTape() as gen_tape, tf.GradientTape() as
disc_tape:
        generated_data = generator(real_x, training=True)
        generated_data_reshape = tf.reshape(generated_data,
[generated_data.shape[0], generated_data.shape[1], 1])
        d_fake_input = tf.concat([tf.cast(generated_data_reshape,
tf.float64), yc], axis=1)
        real_y_reshape = tf.reshape(real_y, [real_y.shape[0],
real_y.shape[1], 1])
        d_real_input = tf.concat([real_y_reshape, yc], axis=1)

        real_output = discriminator(d_real_input, training=True)
        fake_output = discriminator(d_fake_input, training=True)

        g_loss = generator_loss(fake_output)
        disc_loss = discriminator_loss(real_output, fake_output)

```

```

        gradients_of_generator = gen_tape.gradient(g_loss,
generator.trainable_variables)
        gradients_of_discriminator = disc_tape.gradient(disc_loss,
discriminator.trainable_variables)

        g_optimizer.apply_gradients(zip(gradients_of_generator,
generator.trainable_variables))
        d_optimizer.apply_gradients(zip(gradients_of_discriminator,
discriminator.trainable_variables))

    return real_y, generated_data, {'d_loss': disc_loss, 'g_loss':
g_loss}

def train(real_x, real_y, yc, Epochs, generator, discriminator,
g_optimizer, d_optimizer, checkpoint = 50):
    train_info = {}
    train_info["discriminator_loss"] = []
    train_info["generator_loss"] = []

    for epoch in tqdm(range(Epochs)):
        real_price, fake_price, loss = train_step(real_x, real_y,
yc, generator, discriminator, g_optimizer, d_optimizer)
        G_losses = []
        D_losses = []
        Real_price = []
        Predicted_price = []
        D_losses.append(loss['d_loss'].numpy())
        G_losses.append(loss['g_loss'].numpy())
        Predicted_price.append(fake_price.numpy())
        Real_price.append(real_price.numpy())

        #Save model every X checkpoints
        if (epoch + 1) % checkpoint == 0:

```

```

        tf.keras.models.save_model(generator,
f'./models_gan/{stock_name}/generator_V_%d.h5' % epoch)
        tf.keras.models.save_model(discriminator,
f'./models_gan/{stock_name}/discriminator_V_%d.h5' % epoch)
        print('epoch', epoch + 1, 'discriminator_loss',
loss['d_loss'].numpy(), 'generator_loss', loss['g_loss'].numpy())

        train_info["discriminator_loss"].append(D_losses)
        train_info["generator_loss"].append(G_losses)

        Predicted_price = np.array(Predicted_price)
        Predicted_price =
Predicted_price.reshape(Predicted_price.shape[1],
Predicted_price.shape[2])
        Real_price = np.array(Real_price)
        Real_price = Real_price.reshape(Real_price.shape[1],
Real_price.shape[2])

        plt.subplot(2,1,1)
        plt.plot(train_info["discriminator_loss"], label='Disc_loss',
color='#000000')
        plt.xlabel('Epoch')
        plt.ylabel('Discriminator Loss')
        plt.legend()

        plt.subplot(2,1,2)
        plt.plot(train_info["generator_loss"], label='Gen_loss',
color='#000000')
        plt.xlabel('Epoch')
        plt.ylabel('Generator Loss')
        plt.legend()

        plt.show()

```

```

    return Predicted_price, Real_price,
np.sqrt(mean_squared_error(Real_price, Predicted_price)) /
np.mean(Real_price)

def plot_results(Real_price, Predicted_price, index_train):
    X_scaler = load(open('/content/X_scaler.pkl', 'rb'))
    y_scaler = load(open('/content/y_scaler.pkl', 'rb'))
    train_predict_index = index_train

    rescaled_Real_price = y_scaler.inverse_transform(Real_price)
    rescaled_Predicted_price =
y_scaler.inverse_transform(Predicted_price)

    predict_result = pd.DataFrame()
    for i in range(rescaled_Predicted_price.shape[0]):
        y_predict = pd.DataFrame(rescaled_Predicted_price[i],
columns=["predicted_price"],
index=train_predict_index[i:i+output_dim])
        predict_result = pd.concat([predict_result, y_predict],
axis=1, sort=False)

    real_price = pd.DataFrame()
    for i in range(rescaled_Real_price.shape[0]):
        y_train = pd.DataFrame(rescaled_Real_price[i],
columns=["real_price"], index=train_predict_index[i:i+output_dim])
        real_price = pd.concat([real_price, y_train], axis=1,
sort=False)

    predict_result['predicted_mean'] = predict_result.mean(axis=1)
    real_price['real_mean'] = real_price.mean(axis=1)

    plt.figure(figsize=(16, 8))
    plt.plot(real_price["real_mean"])
    plt.plot(predict_result["predicted_mean"], color = 'r')

```

```

plt.xlabel("Date")
plt.ylabel("Stock price")
plt.legend(("Real price", "Predicted price"), loc="upper
left", fontsize=16)
plt.title("The result of Training", fontsize=20)
plt.show()

```

```

predicted = predict_result["predicted_mean"]
real = real_price["real_mean"]
For_MSE = pd.concat([predicted, real], axis = 1)
RMSE = np.sqrt(mean_squared_error(predicted, real))
print('-- Train RMSE -- ', RMSE)

```

```
## Test Code
```

```
@tf.function
```

```

def eval_op(generator, real_x):
    generated_data = generator(real_x, training = False)

    return generated_data

```

```

def plot_test_data(Real_test_price, Predicted_test_price,
index_test):
    X_scaler = load(open('X_scaler.pkl', 'rb'))
    y_scaler = load(open('y_scaler.pkl', 'rb'))
    test_predict_index = index_test

    rescaled_Real_price =
y_scaler.inverse_transform(Real_test_price)
    rescaled_Predicted_price =
y_scaler.inverse_transform(Predicted_test_price)

```

```

predict_result = pd.DataFrame()
for i in range(rescaled_Predicted_price.shape[0]):
    y_predict = pd.DataFrame(rescaled_Predicted_price[i],
columns=["predicted_price"], index=test_predict_index[i:i+output_dim])
    predict_result = pd.concat([predict_result, y_predict],
axis=1, sort=False)

real_price = pd.DataFrame()
for i in range(rescaled_Real_price.shape[0]):
    y_train = pd.DataFrame(rescaled_Real_price[i],
columns=["real_price"], index=test_predict_index[i:i+output_dim])
    real_price = pd.concat([real_price, y_train], axis=1,
sort=False)

predict_result['predicted_mean'] = predict_result.mean(axis=1)
real_price['real_mean'] = real_price.mean(axis=1)

predicted = predict_result["predicted_mean"]
real = real_price["real_mean"]
For_MSE = pd.concat([predicted, real], axis = 1)
RMSE = np.sqrt(mean_squared_error(predicted, real))
print('Test RMSE: ', RMSE)

plt.figure(figsize=(16, 8))
plt.plot(real_price["real_mean"], color='#00008B')
plt.plot(predict_result["predicted_mean"], color = '#8B0000',
linestyle='--')
plt.xlabel("Date")
plt.ylabel("Stock price")
plt.legend(("Real price", "Predicted price"), loc="upper
left", fontsize=16)
plt.title(f"Prediction on test data for {stock_name}",
fontsize=20)
plt.show()

```

```

learning_rate = 5e-4
epochs = 400

g_optimizer = tf.keras.optimizers.Adam(lr = learning_rate)
d_optimizer = tf.keras.optimizers.Adam(lr = learning_rate)

generator = make_generator_model(X_train.shape[1], output_dim,
X_train.shape[2])
discriminator = make_discriminator_model(X_train.shape[1])

from keras.utils import plot_model
plot_model(generator, to_file='generator_keras_model.png',
show_shapes=True)

tf.keras.utils.plot_model(discriminator,
to_file='discriminator_keras_model.png', show_shapes=True)

# ## Train and test model

Predicted_price, Real_price, RMSPE = train(X_train, y_train,
yc_train, epochs, generator, discriminator, g_optimizer, d_optimizer)

test_generator =
tf.keras.models.load_model(f'./models_gan/{stock_name}/generator_V_{ep
ochs-1}.h5')

predicted_test_data = eval_op(test_generator, X_test)
plot_test_data(y_test, predicted_test_data, index_test)

return predicted_test_data

gan_preds = gan_pred(stock_name)

```

```

%%time
import snsrape.modules.twitter as sntwitter
import pandas as pd

# Query and creating an empty list for storing tweets
stock_name = 'PG'
query = stock_name + " lang:en until:2022-09-30 since:2021-09-30"
tweets = []

# A simple for loop
for tweet in sntwitter.TwitterSearchScrapper(query).get_items():
    if tweet.likeCount > 100:
        tweets.append([tweet.date, tweet.content])

# Converting it to a pandas dataframe
df = pd.DataFrame(tweets, columns = ['Date', 'Tweet'])
print(len(df))

df.to_csv(path_or_buf=f'{stock_name}_tweets.csv', index=False)

ccdata = df.copy()
ccdata["sentiment_score"] = ''
ccdata["Negative"] = ''
ccdata["Neutral"] = ''
ccdata["Positive"] = ''
ccdata

from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import unicodedata
sentiment_i_a = SentimentIntensityAnalyzer()
for indexx, row in ccdata.T.iteritems():
    try:

```

```

        sentence_i = unicodedata.normalize('NFKD', ccdata.loc[indexx,
'Tweet'])
        sentence_sentiment = sentiment_i_a.polarity_scores(sentence_i)
        ccdata.at[indexx, 'sentiment_score'] =
sentence_sentiment['compound']
        ccdata.at[indexx, 'Negative'] = sentence_sentiment['neg']
        ccdata.at[indexx, 'Neutral'] = sentence_sentiment['neu']
        ccdata.at[indexx, 'Positive'] = sentence_sentiment['pos']
    except TypeError:
        print (ccdata.loc[indexx, 'Tweet'])
        print (indexx)
        break

```

```

ccdata['Date'] = ccdata['Date'].dt.date
ccdata = ccdata.drop(columns=['Negative', 'Positive', 'Neutral'])

```

```
ccdata.head()
```

```
twitter_df = ccdata.groupby([df['Date'].dt.date]).mean()
```

```
import yfinance as yf
```

```
# getting data from Yahoo Finance
```

```
stock_name = 'AMZN' # here you can change the name of stock ticker,
for example we will take AMD ticker
```

```
stock_df = yf.download(stock_name, start="2021-09-30", end="2022-09-
30")
```

```
stock_df = stock_df.reset_index(level=0)
```

```
stock_df['Date'] = stock_df['Date'].dt.date
```

```
final_df = stock_df.join(twitter_df, how="left", on="Date")
```

### **Висновки до розділу 3.**

В останньому розділі було імплементовано моделі, обрані в розділах 1 та 2, у фінальний програмний продукт, який є результатом економічної задачі цієї роботи.

В результаті було розроблено комбіновану модель керування портфелем, яка прогнозує розвиток використаних фондових активів та створює стратегію керування ними на основні отриманих прогнозованих даних. На вхід подаються історичні дані за минулий рік, які потім розроблена модель використовує для тренування і прогнозування. Крім чистих історичних даних також розраховуються такі технічні індикатори, як ковзні середні за минулі 7 та 20 днів, межі Боллінджера тощо.

Але технічні індикатори та чиста вартість активу все ще не дають повну картину розвитку фондового ринку. Із розвитком технологій та збільшенням швидкості поширення інформації ринок стає більш залежним від таких зовнішніх факторів, як новини, соціальний імідж, репутація. З поширеним використанням соціальних мереж думка акціонерів стала більш актуальною, ніж будь-коли раніше. Оцінка думок соціальних мереж у поєднанні з традиційними моделями прогнозування значно підвищив рівень успішності методологій прогнозування.

Для цього ми використовуємо метод сентиментального аналізу – це метод аналізу тексту, який виявляє полярність (наприклад, позитивну чи негативну думку) у тексті, будь то весь документ, абзац, речення чи пункт. Аналіз настроїв має на меті виміряти ставлення, почуття, оцінки, ставлення та емоції оратора/письма на основі обчислювальної обробки суб'єктивності в тексті. В даній роботі для оцінки настрою тексту ми використаємо модель VADER. На вихід ми отримуємо середній показник настрою для всіх дописів в соціальній мережі Twitter про розглянутий фондовий актив в певний день.

Нові дані подаються в мережу GAN – генеративну змагальну мережу, яка базується на основі рівноваги Неша між двома моделями – дискримінатора та генератора. Генератор вчиться створювати нові зразки даних з вивченого розподілу тренувальних даних, а дискримінатор вчиться розрізняти згенеровані дані від справжніх. Метою даної структури є отримання якомога меншої похибки генератора при максимальній похибці дискримінатора, тобто, генерування нових даних, які якомога наближені до тренувальних, але їх важко відрізнити від справжніх.

Використана архітектура GAN має мережу з 5 блоків LSTM як генератор і п'ятишарову CNN як дискримінатор. Дискримінатор був реалізований з 5 згортковими шарами, за якими слідують 3 сплюснені шари і два шари із лінійною та сигмоїдною активаціями відповідно.

Після прогнозування вартості фондових активів, прогнозні дані подаються в модель Марковіца для розрахунку початкової інвестиційної стратегії. Далі прораховується інвестиційна діяльність на наступний місяць за допомогою моделі Хуанга-Літценбергера, де на вхід ми подаємо прогноз цін акцій на певний часовий період (день), а на вихід отримаємо ваги портфеля.

Ефективність моделі перевіряється на прогнозованих та фактичних історичних даних за розглянутий часовий період. Сама модель дає позитивні результати, оскільки гнучкість стратегії стає помітною при зміні вартості базового активу, що приносить прибуток після коротких угод. У цьому випадку ми отримали портфоліо з очікуваною прибутковістю 1% та 1,9%, в той же час чиста модель Марковіца без коригування вагів дає на виході негативні прибутки.

## ВИСНОВКИ

Основною метою створення інвестиційного портфеля є реалізація політики фінансового інвестування компанії шляхом вибору найбільш прибуткових і безпечних фінансових інструментів.

Перед створенням інвестиційного портфеля необхідно визначити конкретні цілі та завдання стратегії управління ними та пріоритети у виборі активів фонду для інвестування, а також визначити частку інвестиційних ресурсів для різних типів портфелів.

Кожен інвестор створює портфель цінних паперів відповідно до власних уявлень та уподобань. Однак існують основні принципи, за якими можна побудувати інвестиційний портфель. Одним із головних принципів портфелю є рентабельність, тобто здатність інвестора отримувати визначені майбутні доходи. Це включає курсові різниці, а також дивіденди або відсотки за цими цінними паперами. Тому при розробці моделі керування інвестиціями було приділено увагу максимізації прибутку оптимального інвестиційного портфеля.

В ході виконання дисертації було досліджено різні методи прогнозування ринкових цін акцій, було розроблено модель для визначення оптимального інвестиційного портфелю та його вартість.

Основними методами прогнозування часових рядів і цін акцій є наступні: метод Бокса-Дженкінса (ARIMA), метод Метрополіса-Хастінгса (Монте-Карло з марківськими ланцюгами), нейронні мережі (рекурентні, серед яких LSTM) та багато інших. Також було розглянуто генеративні змагальні мережі GAN, які містять в свої структурі LSTM і тільки набирають популярність для прогнозування часових рядів. Дані методи було детально розглянуто та апробовано на історичних даних фондових активів, можемо побачити, що мережі GAN є найбільш успішними для даного завдання, тому в фінальній моделі були використані саме вони. Причиною цього є той факт,

що структура GAN є більш складною ніж за звичайні нейронні мережі, а змагальна ідея в цій схемі дає більше покращення в результатах.

В результаті дослідження було розроблено комбіновану модель керування портфелем, яка прогнозує розвиток використаних фондових активів та створює стратегію керування ними на основні отриманих прогнозованих даних. На вхід подаються історичні дані за минулий рік, які потім розроблена модель використовує для тренування і прогнозування. Крім чистих історичних даних також розраховуються такі технічні індикатори, як ковзні середні за минулі 7 та 20 днів, межі Боллінджера тощо.

Проводився сентимент аналіз дописів в соціальній мережі Twitter для оцінки настроїв учасників фондового ринку за допомогою заздалегідь натренованої моделі VADER. Отримані дані подаються в модель GAN для тренування та прогнозування вартостей акцій на наступний місяць.

Після прогнозування вартості фондових активів, прогнозні дані подаються в модель Марковіца для розрахунку початкової інвестиційної стратегії. Прораховується інвестиційна діяльність на наступний місяць за допомогою моделі Хуанга-Літценбергера, де на вхід подається прогноз цін акцій на певний часовий період (день), а на виході отримаємо ваги портфеля.

Алгоритм був автоматизований у вигляді розробленого програмного додатку. Сама модель дає позитивні результати, оскільки гнучкість стратегії стає помітною при зміні вартості базового активу, що приносить прибуток після коротких угод. У цьому випадку ми отримали портфоліо з очікуваною прибутковістю 1% та 1,9%, в той же час чиста модель Марковіца без коригування вагів дає на виході негативні прибутки.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Замковий О. Портфельні теорії інвестування. Методичні рекомендації для самостійної підготовки до практичних занять з дисципліни магістрів спеціальності 072 Фінанси, банківська справа та страхування. Дніпро : НТУ«ДП», 2020. 70 с.
2. Коновалюк М. Байєсівський аналіз моделі стохастичної волатильності в середовищі openbugs. НТУУ "КПІ". 2011. С. 8.
3. Гальперін В. Динамічне управління інвестиційним портфелем з урахуванням стрибкоподібної зміни цін фінансових активів. Вісник ТГУ. 2003. № 280. С. 112–117.
4. Гонтарь Н. Оптимізація інвестиційного портфеля методом Марковіца. 2017.
5. Єфімова К. Статистичний аналіз ризику та доходності портфельних інвестицій. 2016.
6. Xiang M. Dynamic mean-variance portfolio optimisation. 2018.
7. Ayodele A. Stock price prediction using the arima model. 2014.
8. Hernandez A. Monte-carlo simulation with black-scholes. Malardalen university. 2010. P. 13.
9. Monte carlo markov chain algorithm. URL: <https://github.com/Joseph94m/MCMC/blob/master/MCMC.ipynb>.
10. Understanding delta hedging options using monte carlo simulation. Finance training course. URL: <https://financetrainingcourse.com/education/2012/10/understanding-delta-hedging-for-options-using-monte-carlo-simulation/>.
11. Computational finance: monte carlo (mc) simulation method: understanding drift, diffusion and volatility drag. Finance training course. URL: <https://financetrainingcourse.com/education/2010/08/computational-finance-monte-carlo-mc-simulation-method-understanding-drift-diffusion-and-volatility-drag/>.

12. Asmama O. Investment strategies: portfolio optimization & risk management with r language. 2019. P. 56.
13. Michaud R. Efficient asset management. New york: oxford university press. 2008. P. 145.
14. Adhikari R. An introductory study on time series modeling and forecasting. 2010. P. 67.
15. Agarwal H. Analysis and prediction of stock market trends using deep learning. 2020. P. 11.
16. Ganegedara T. Stock market predictions with lstm in python. 2020.
17. Zexin H. A survey of forex and stock price prediction using deep learning. Mdpi. 2021.
18. Moghar A. Stock market prediction using lstm recurrent neural network. 2020. P. 5.
19. Brood T. Monte carlo simulations of stock prices. 2018. P. 33.
20. Sonono M. Prediction of stock price movement using continuous time models. 2015. P. 15.
21. Arnaut-Berilo A. Effectiveness of monte carlo simulation and arima model in predicting stock prices. 2015. P. 12.
22. Stock prediction using hybrid arima and gru models / R. Mangalampalli et al. 2020. P. 5.
23. Khan S., Alghulaiakh H. Arima model for accurate time series stocks forecasting. 2020. P. 5.
24. Zou Z. Using lstm in stock prediction and quantitative trading. 2020. P. 6.
25. Pav S. Notes on the sharpe ratio. 2020. P. 50.
26. Duran F., Cotta C., Fernandez A. On the use of sharpe's index in evolutionary portfolio optimization under markowitz's model. 2009. P. 6.
27. Sharpe W. Capital asset price: a theory of market equilibrium under conditions of risk. *Journal of finance*. 1964. Vol. 19, no. 3.

28. Huang C., Litzenberger R. Foundations for financial economics. 1988. 365 p.
29. Romero R. Generative adversarial network for stock market price prediction. *CD230: Deep Learning, Stanford University*. 2018. P. 5.
30. Guoqiang Z. Stock market prediction based on generative adversarial network. 2019. URL: [https://www.researchgate.net/publication/331002527\\_Stock\\_Market\\_Prediction\\_Based\\_on\\_Generative\\_Adversarial\\_Network/](https://www.researchgate.net/publication/331002527_Stock_Market_Prediction_Based_on_Generative_Adversarial_Network/) (дата звернення: 29.10.2022).
31. Stock price prediction using generative adversarial networks / H. Lin et al. 2021. URL: <https://thescipub.com/abstract/jcssp.2021.188.196/> (дата звернення: 29.10.2022).
32. Staffini A. Stock price forecasting by a deep convolutional generative adversarial network. 2022. URL: <https://www.frontiersin.org/articles/10.3389/frai.2022.837596/full> (дата звернення: 29.10.2022).
33. Zhang K., Zhong G., Dong J. Stock market prediction based on generative adversarial network. *International Conference on Identification, Information and Knowledge in the Internet of Things*. 2018. URL: <https://www.sciencedirect.com/science/article/pii/S1877050919302789> (дата звернення: 29.10.2022).
34. Seranto M. L. A timegan application for generating time series related to climate prediction. *Ca' Foscari University of Venice*. 2021. P. 49.
35. Twitter sentiment analysis and influence on stock performance using transfer entropy and egarch methods / R. Mendoza-Urdiales et al. *MDP: Entropy*. 2022. No. 24. P. 17.
36. Mittal A., Goel A. Stock prediction using twitter sentiment analysis. *CS 229*. 2011. P. 5.
37. Sentiment analysis of twitter data for predicting stock market movements / V. Pagolu et al. *International conference on Signal Processing*,

*Communication, Power and Embedded System (SCOPE)-2016*. 2016. URL: <https://arxiv.org/pdf/1610.09225.pdf>. (дата звернення: 29.10.2022).

38. Patel U. Twitter data predicting stock price using data mining techniques. *University of Bridgeport*. 2016. URL: [https://www.researchgate.net/publication/315562673\\_Twitter\\_Data\\_Predicting\\_Stock\\_Price\\_Using\\_Data\\_Mining\\_Techniques](https://www.researchgate.net/publication/315562673_Twitter_Data_Predicting_Stock_Price_Using_Data_Mining_Techniques) (дата звернення: 29.10.2022).

39. Goodfellow I. Generative adversarial networks. *CoRR, abs/1701.00160*. 2017. P. 57.

40. Yoon J., Jarett D., van der Schaar M. Time-series generative adversarial networks. *NeurIPS 2019*. 2019. P. 11.

41. Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ICLR 2016*. 2016. P. 16.

42. Takahashi S., Chen Y., Tanaka-Ishii K. Modeling financial time-series with generative adversarial networks. *Statistical Mechanics and its Applications*. 2019. P. 14.

43. Sawada A., Miyagawa T., Ebihara A. Convolutional neural networks for time-dependent classification of variable-length time series. *IEEE 2022*. 2022. P. 9.

44. Fawaz H., Forestier G., Weber J. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*. 2019. P. 44.

45. Albawi S., Mohammed T., Al-Zawi S. Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*. 2017.

46. Indolia S. Conceptual understanding of convolutional neural network - a deep learning approach. 2018. P. 10.

47. Hutto C., Eric G. Vader: a parsimonious rule-based model for sentiment analysis of social media text. *Advancement of Artificial Intelligence*. 2014. P. 10.

48. Williams O. Empirical optimization of Bollinger bands for profitability. *Simon Fraser University*. 2013. P. 72.
49. Shah N. A comparative study on technical analysis by Bollinger band and RSI. *International Journal in Management and Social Science*. 2015. P. 19.
50. Dinesh S. Prediction of trends in stock market using moving averages and machine learning. *2021 6th International Conference for Convergence in Technology (I2CT)*. 2021.
51. Ma G. The prediction of enterprise stock change trend by deep neural network model. *Computational Intelligence and Neuroscience*. 2022. P. 9.
52. Pradhan R. Stock trend prediction and analysis using lstm neural network and dual moving average crossover algorithm. *IOP Conf. Series: Materials Science and Engineering*. 2021. P. 11.
53. Xiao D. Research on Stock Price Time Series Prediction Based on Deep Learning and Autoregressive Integrated Moving Average. *Scientific Programming*. 2022. P. 12.
54. Shynkevich Y. Forecasting price movements using technical indicators: investigating the impact of varying input window length. 2017. P. 40.
55. Darapaneni N., Paduri A., Sharma H. Stock Price Prediction using Sentiment Analysis and Deep Learning for Indian Markets. *arXiv:2204.05783*. 2022. P. 15.
56. Singh S., Kaur A. Twitter sentiment analysis for stock prediction. *2nd International Conference on "Advancement in Electronics & Communication Engineering (AECE 2022)*. 2022. P. 4.
57. Koukaras P., Nousi C., Tjortjis C. Stock market prediction using microblogging sentiment analysis and machine learning. *Telecom 2022*. 2022. No. 3. P. 21.
58. Karlemstrand R., Ebba L. Using twitter attribute information to predict stock prices. *arXiv:2105.01402v1*. 2021. P. 53.

59. Gondaliya C. Sentiment analysis and prediction of indian stock market amid covid-19 pandemic. *IOP Conf. Series: Materials Science and Engineering*. 2021. P. 10.

60. Liu C. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *MDPI Applied Sciences* 2020. 2020. Vol. 10, no. 437. P. 20.

61. Chaweewanchon A., Chaysiri R. Markowitz mean-variance portfolio optimization with predictive stock selection using machine learning. *Int. J. Financial Stud.* 2022. 2022. Vol. 10, no. 64. P. 19.