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

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

«На правах рукопису»
УДК 004.032.26

«До захисту допущено»
В.о. завідувача кафедри
_____ Едуард ЖАРІКОВ
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інженерія програмного забезпечення інформаційних систем»

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Програмна система прогнозування ринку акцій на основі даних із соціальних мереж»

Виконав (-ла):
студент (-ка) ІІ курсу, групи ІТ-04мп
Нагуляк Андрій Сергійович _____

Керівник:
професор, д.т.н.
Сидоров Микола Олександрович
_____

Рецензент:
доцент кафедри ІСТ, к.т.н
Коган Алла Вікторівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2021 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення інформаційних систем»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Едуард ЖАРІКОВ

«___»_____2021р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Нагуляку Андрію Сергійовичу

1. Тема дисертації «Програмна система прогнозування ринку акцій на основі даних із соціальних мереж», науковий керівник дисертації Сидоров Микола Олександрович, д.т.н, професор, затверджені наказом по університету від «27» жовтня 2021 р. № 3587-с

2. Термін подання студентом дисертації «6» грудня 2021 р.

3. Об’єкт дослідження – програмне забезпечення систем прогнозування ринка акцій.

4. Предмет дослідження – підходи, методи, моделі, створення і супроводження програмного забезпечення системи прогнозування ринка акцій

5. Перелік завдань, які потрібно розробити – аналіз існуючих досліджень; розробка програмного забезпечення для передбачення ціни акцій фондового ринку; дослідження ефективності розробленого програмного забезпечення.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – 4 плакати

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1. Огляд існуючих досліджень	Сидоров М.О, професор, д.т.н.		
Розділ 2. Проектування системи			
Розділ 3. Розробка системи			
Розділ 4. Оцінка результатів			
Розділ 5. Розробка стартап-проекту			
Нормоконтроль	Лісовиченко О.І., доцент		

9.Дата видачі завдання «30» вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Затвердження теми роботи	01.09.2021-06.09.2021	
2	Вивчення та аналіз завдання	06.09.2021-06.10.2021	
3	Розробка архітектури та загальної структури системи	06.10.2021-16.10.2021	
4	Розробка структур окремих підсистем	16.10.2021-20.10.2021	
5	Програмна реалізація системи	20.10.2021-28.10.2021	
6	Оцінка ефективності системи	28.10.2021-03.11.2021	
7	Оформлення пояснювальної записки	03.11.2021-22.11.2021	
8	Подання дисертації на попередній захист	22.11.2021	
9	Подання дисертації на захист	6.12.2021	

Студент Андрій НАГУЛЯК

Науковий керівник Микола СИДОРОВ

Реферат

Обсяг дисертації – 90 аркушів, містить 6 додатків та 25 посилань на використані джерела. Представлено 25 рисунків та 27 таблиць.

Актуальність теми. Акції, є найпопулярнішим фінансовим активом. На сьогоднішній день, існує багато бірж, завдяки яким, кожен може володіти акціями. Точне прогнозування прибутковості фондового ринку - дуже складне завдання. Існує безліч факторів, що ускладнюють прогнозування цін на акції, включаючи волатильний та нелінійний характер фінансових фондових ринків. З появою штучного інтелекту та збільшенням обчислювальних можливостей, методи програмного прогнозування виявилися найбільш ефективними при прогнозуванні цін на акції.

Мета дослідження. Метою дипломної роботи є створення програмного забезпечення для передбачення ринка акцій на основі соціальних мереж. Поставлена мета реалізується через **наступні завдання** :

— аналіз інструментів для зчитування даних з соціальних мереж;

— аналіз інструментів для зчитування даних про акції компаній;

— аналіз інструментів та методів обробки даних для побудови моделей прогнозування та кореляційного аналізу;

— аналіз методів для прогнозування часових рядів;

— вибір технологій та архітектури для побудови ПЗ;

— розробка програмного забезпечення.

Об’єктом дослідження є програмне забезпечення систем прогнозування ринка акцій.

Предметом дослідження є підходи, методи, моделі, створення і підтримка програмного забезпечення системи прогнозування ринка акцій.

Наукова новизна. У цьому дослідженні було проаналізовано різні моделі для прогнозування цін на акції, обрано найбільш ефективну модель, та вдосконалено його, шляхом використання даних з соціальних мереж. Фінансові дані, такі як ціни відкриття, максимуму, мінімуму та закриття акцій, а також дані з

соціальних мереж, такі як кількість згадок про компанію, впливовість людей які згадують компанію, оцінка згадки компанії, використовуються для створення нових змінних, які використовуються як вхідні дані для моделі.

Моделі оцінюються за допомогою стандартних стратегічних показників ефекту прогнозування:

— середня абсолютна помилка (MAE),

— середньоквадратична помилка (RMSE)

— R-квадрат (R^2).

Низькі значення індикаторів MAE, RMSE показують, що модель є досить точною при прогнозуванні ціни закриття акцій. А високий показник R^2 показує, що модель є досить ефективною.

Практичне значення. Спроектована та реалізована система підходить для інвесторів, які хочуть підтвердити свої думки, щодо росту акцій компанії. Архітектура розроблена таким чином, що можна додавати все більше методів передбачення та джерел даних, для подальшого дослідження.

Зв'язок з науковими програмами, планами, темами. Робота виконувалась на кафедрі інформатики та програмної інженерії Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Апробація. Наукові положення дисертації пройшли апробацію на Першій Всеукраїнській науково-практичній конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології»(SoftTech-2021) – м. Київ.

Публікації. Наукові положення дисертації опубліковані в: Нагуляк А. С. використання моделі CLSTM для прогнозування курсу акцій фондового ринку // Матеріали Першої Всеукраїнської науково-практичної конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології»(SoftTech-2021) – м. Київ. НТУУ «КПІ ім. Ігоря Сікорського», 22-26 листопада 2021 р – с. 135.

Ключові слова: ПЕРЕДБАЧЕННЯ ФОНДОВОГО РИНКУ, НЕЙРОННІ МЕРЕЖІ, ЧАСОВІ РЯДИ, ГЛИБОКЕ НАВЧАННЯ, ЗГОРТКОВА МЕРЕЖА, МЕРЕЖА ДОВГОСТРОКОВОЇ ПАМ'ЯТІ.

Abstract

The volume of the dissertation is 90 sheets, contains 6 appendices and 25 references to the sources used. There are 25 figures and 27 tables.

Topicality. Stocks are the most popular financial asset. Today, everyone can own shares through many exchanges. Accurate forecasting of stock market profitability is a very difficult task. There are many factors that make it difficult to forecast stock prices, including the volatile and non-linear nature of financial stock markets. With the advent of artificial intelligence and the increase in computing power, software-forecasting methods have proven to be the most effective in forecasting stock prices.

The aim of the study. The purpose of the thesis is to create software for predicting the stock market based on social networks. This goal is realized through the **following tasks**:

— analysis of tools for reading data from social networks;

— analysis of tools for reading data on company shares;

— analysis of tools and methods of data processing for building models of forecasting and correlation analysis;

— analysis of methods for forecasting time series;

— choice of technologies and architecture for software construction;

— software development.

The object of research is the software of stock market forecasting systems.

The subject of research is the approaches, methods, models, creation and maintenance of stock market forecasting software.

The scientific novelty. This study analyzed various models for forecasting stock prices, selected the most effective model, and improved it by using data from social networks. Financial data such as opening, maximum, minimum and closing prices, as well as social media data such as the number of mentions of the company, the influence of people who mention the company, the evaluation of the company's mention, are used to create new variables used as input for the model.

Models are evaluated using standard strategic indicators of the forecasting effect:

— mean absolute error (MAE);

— standard error (RMSE);

— R-square (R ^ 2).

Low values of MAE, RMSE indicators show that the model is quite accurate in predicting the closing price of shares. And a high value of R ^ 2 shows that the model is quite effective.

The practical value. The designed and implemented system is suitable for investors who want to confirm their views on the growth of the company's shares. The architecture is designed in such a way that more and more prediction methods and data sources can be added for further research.

Relationship with working with scientific programs, plans, topics. The work was performed at the Department of Informatics and Software Engineering of the National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorsky".

Approbation. The scientific provisions of the dissertation were tested at the First All-Ukrainian scientific-practical conference of young scientists and students "Software Engineering and Advanced Information Technologies" (SoftTech-2021) - Kyiv.

Publications. Scientific provisions of the dissertation are published in: Nagulyak AS use of CLSTM model for forecasting the stock market stock price // Proceedings of the First All-Ukrainian scientific-practical conference of young scientists and students "Software Engineering and Advanced Information Technologies" (SoftTech-2021) - m. Kiev. NTUU "KPI them. Igor Sikorsky ", November 22-26, 2021 - p. 135.

Keywords: STOCK MARKET FORECAST, NEURAL NETWORKS, TIME SERIES, DEEP STUDY, ROLLER NETWORK, LONG-TERM MEMORY NETWORK.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення;

НМ – нейронна мережа;

БД – база даних;

СМ – стохастична модель;

MAE – середня абсолютна помилка;

RMSE – середня квадратична помилка.

ВСТУП

Прогнозування майбутніх цін акцій має важливе значення, оскільки воно може зменшити ризик прийняття рішень шляхом належного визначення майбутнього руху інвестиційного активу.

Фундаментальний і технічний аналіз – це два основних підходи, які використовуються для прогнозування тенденцій акцій. Технічний аналіз перевіряє минулі дані та обсяги цін на акції, тоді як фундаментальний аналіз розглядає не лише статистику акцій, а й оцінює результати діяльності галузі, політичні події та економічні обставини.

Існує багато джерел текстових даних, таких як новини, твіти, річні звіти тощо, які можна проаналізувати, щоб отримати важливу інформацію. В порівнянні з числовими даними, текстові дані, є джерелом прихованої інформації. Вони дозволяють легко обґрунтувати прогнозування фінансових тенденції . Наприклад, повідомлення про компанію зі словами або фразами на кшталт «відставка», «ризик дефолту» допомагає інвестору передбачити зниження курсу акцій компанії. Фундаментальний аналіз більш реалістичний, оскільки він оцінює ринок у ширшому обсязі.

Ставлення людей до компанії є одним з найголовніших факторів росту акцій. Так, випадок 2021 року, пов’язаний з розкриттям особистих даних мільйонів користувачі компанією «Facebook», призвів до різкого падіння акцій компанії, або, наприклад, лише один «твіт» Ілона Маска про зміну політики компанії «Тесла» призвів до різкого росту акцій. Також якщо говорити про компанії, які не є «гігантами» ринку, то можна помітити тенденцію, що саме популярність нових компаній пропорційна кількості згадок про неї в соціальних мережах.

Зважаючи на цю інформацію можна зробити висновок, що актуальність даної роботи зумовлена тим, що на сьогоднішній день саме соціальні мережі мають великий вплив на ринок акцій, а отже це питання потребує ретельного вивчення. З цього можна зробити висновок, що фундаментальний аналіз для прогнозування ринку акцій на основі соціальних мереж є актуальною темою для дослідження.

Таким чином, метою дипломної роботи є створення програмного забезпечення для передбачення ринка акцій на основі соціальних мереж.

Об’єктом дослідження є програмне забезпечення систем прогнозування ринка акцій.

Предметом дослідження є підходи, методи, моделі, створення і підтримка програмного забезпечення системи прогнозування ринка акцій.

Індивідуальне завдання передбачає створення програмної системи , з метою реалізації можливості прогнозування росту акцій компаній, базуючись на даних соціальних мереж. При цьому важливо зробити цей процес відкритим для розширення як зі сторони збільшення методів прогнозування даних, так і зі сторони збільшення потенційних джерел інформації, щоб зробити цю систему максимально адаптивною для подальших досліджень.

Використаними у роботі методами досліджень є методи отримання, обробки даних, методи кореляційного аналізу та аналізу причинності, метод оцінки текстових даних, нейронні мережі для статистичного аналізу даних.

Для вирішення завдань даної роботи використано мову програмування, обробки та аналізу даних Python.

Практична значимість дослідження полягає в створенні програмного забезпечення, що дозволить прогнозувати зміни на ринку акцій зважаючи на дані з соціальних мереж, при використанні якого інвесторами, можливо зменшити ризики втрати грошових інвестицій.

1 ОГЛЯД ІСНУЮЧИХ ДОСЛІДЖЕНЬ

1.1 Цілі дослідження

Основна ціль цього дослідження — створення програмного забезпечення для передбачення фондового ринку акцій на основі соціальних мереж. Для проведення плану дослідження, потрібно виявити, які цілі ми ставимо у ньому. Цілі досліджень знайшли своє відображення в дослідницьких питаннях (ДП), в таблиці 1.1

Таблиця 1.1 - Дослідницькі питання

ДП1	Які програмні забезпечення існують для передбачення ринку акцій та як вони влаштовані ?
ДП2	Які технології найбільш часто використовуються для передбачення акцій ?
ДП3	Яка архітектура найбільш ефективна для передбачення курсу акцій на фондовому ринку?
ДП4	Які підходи найчастіше використовуються для отриманні даних для передбачення фондового ринку?

З ДП1 ми маємо намір вияснити, які існують програмні забезпечення для прогнозування ринку акцій.

З ДП2 ми маємо намір розглянути існуючі технології передбачення фондового ринку.

З ДП3 ми маємо намір розглянути і порівняти різні архітектури програмних забезпечень передбачення акцій.

З ДП4 ми маємо намір вияснити які підходи можуть бути використанні для передбачення фондового ринку.

1.2 Пошук релевантних досліджень

Для того щоб знайти відповіді на питання дослідження, підтвердити актуальність теми та порівняти отримані результати з даними інших дослідників, потрібно провести ретельний огляд літератури. При огляді літератури було використано методологію “system mapping study”.

Для знаходження релевантних досліджень було використано:

- критерії включення;
- критерії виключення;
- пошукові стрічки.

Пошук охоплював електронні бази даних, які вважаються найбільш релевантними науковими джерелами: ACM Digital Library1 , EngineeringVillage2 , IEEE Xplore3 та ScienceDirect5. Під час проведення пошуку було розглянуто публікації англійською мовою з обмеженням, що публікації повинні бути новіші ніж 2010 року, оскільки саме після 2010 року соціальні мережі набули великої популярності.

Для виключення досліджень, які не є релевантними були визначені критерії включення та виключення. У таблиці 1.2 наведені критерії включення (IC), у таблиці 1.3 наведені критерії виключення (EC).

Таблиця 1.2 - Критерії включення

IC1	Дослідження, що стосуються створення програмного забезпечення для передбачення часових рядів.
IC2	Магістерські дисертації, докторські дисертації, статті, наукові конференції.
IC3	Дослідження після 2010 року.
IC4	Дослідження повинно містити, методи, підходи або моделі для передбачення фондового ринку

Таблиця 1.3 - Критерії виключення

EC1	Публікації в яких не досліджується створення програмного забезпечення, системи або сервіса.
EC2	Дубльовані дослідження, що повідомляють про подібні результати. У цьому випадку було розглянуто лише найбільш повне дослідження
EC3	Публікації які мають виключно економічне дослідження і не мають технічних підкріплень
EC4	Повторні дослідження, знайдені в різних пошукових системах. У цьому випадку було розглянуто лише одне дослідження

Для ручного пошуку відокремлюємо терміни і створюємо пошукові стрічки. Приклад пошукової стрічки зображено в таблиці 1.4 Ручні результати пошуку зображені в таблиці 1.5.

Таблиця 1.4 - Нефільтровані результати пошуку

Текст пошукової стрічки
(“software for stock market prediction” or “software system for stock market prediction” or “services for stock prediction” or “system for stock market prediction”) and (“fundamental analysis”)

Таблиця 1.5 - Нефільтровані результати пошуку

Пошукова стрічка	Кількість знайдених публікацій
software for stock market prediction	26
software services for stock prediction	18
software system for stock market prediction	12
software for stock market prediction AND fundamental analysis	6
services for stock market prediction AND fundamental analysis	4

Продовження таблиці 1.5

Пошукова стрічка	Кількість знайдених публікацій
software system for stock market prediction AND fundamental analysis	5
software system for stock prediction AND fundamental analysis	6
software system for stock market prediction AND social network	5

Після ручного пошуку та за допомогою критеріїв включення та виключення було знайдено 46 робіт які задовольняють умови. Після фільтрації дублікатів було залишено 32 роботи. Далі при більш детальнішому розгляді було знайдено, що 4 статті є стислими версіями попередніх робіт, тому в результаті залишилось 28 робіт які задовольняють умовам.

1.3 Аналіз досліджень

Для того, щоб дати відповідь на дослідницькі питання, необхідно провести детальний аналіз існуючих досліджень. Після детального огляду релевантних робіт, була застосована стратегія категорій, спрямовану на розробку власної схеми класифікації для обраних первинних досліджень. Застосовуючи таку стратегію, спочатку розглядаються дослідження з метою пошуку ключових слів і понять, які відображають їхній внесок. Згодом ці ключові слова та поняття об’єднуються разом, щоб створити загальне розуміння ідеї дослідження. Зрештою, остаточний набір ключових слів використовується для визначення репрезентативних категорій.

Схема класифікації поступово розвивається до своєї остаточної версії в міру додавання, злиття або поділу нових категорій. Наприкінці зображеної стратегії було отримано 6 категорій. Категорії були створені на основі методів, підходів та структур систем, які було розглянуто.

Кожна включене первинне дослідження було віднесено до однієї або декількох з категорій :

— стохастична структура - системи основані на стохастичних моделях ARIMA та GARCH. В даних системах, прогнозування відбувалося шляхом технічного аналізу історичних та статистичних даних, для створення стохастичних моделей передбачення;

— стохастично-нейронна структура - системи основані на поєднанні стохастичних моделей та нейронних мереж. В даних системах прогнозування відбувалося, за допомогою знаходження середнього значення між даними нейронної мережі і даними стохастичної моделі;

— регресійно-опорна структура – система основана на регресійно опорному векторі SVR. Ключовим завданням алгоритмів машинного навчання з урахуванням регресії є передбачення і прогнозування з урахуванням функції зіставлення. Ця функція відображення моделюється шляхом введення набору функцій та даних прогнозування, відомих як набір навчальних даних;

— згортково-нейронна структура – система основана на моделі глибокого навчання, згортковій нейронній мережі (CNN). Мережа використовується для прогнозування ринку, шляхом агрегації характеристик даних;

— довгостроково-нейронна структура – система основана на мережі довгострокової пам’яті (LSTM). В даних системах, прогнозування відбувається шляхом передбачення часових рядів на основі історичних даних;

— фундаментальна структура – система основана на фундаментальному аналізі. В даних системах, за основне джерело інформації, бралися новини, дані з форумів чи соціальних мереж;

— технічна структура – система основана на прогнозуванні часових рядів шляхом технічного аналізу. В даних системах, прогнозування відбувалося тільки на основі історичних даних про акцій.

Розподіл публікацій по категоріям зображено в таблиці 1.7

Таблиця 1.7 - Розподіл публікацій по категоріям

Категорія	Кількість публікацій
стохастична	5
стохастично-нейронна	5
регресійно-опорна	4
згортково-нейронна	9
довгостроково-нейронна	10
фундаментальна	18
технічна	10

З огляду на таблицю 1.7 цілком очевидно, що категорії пов’язані з нейронними мережами, безумовно, категорії, які містять найбільше досліджень. За останні роки практично всі дослідження були пов'язані з використанням або згорткової нейронної мережі або мережі довгострокової пам'яті Також з останніми роками основним методом для аналізу фондового ринка став фундаментальний аналіз, який передбачає використання сторонніх даних, для аналізу акцій (новини, соціальні мережі). Тоді як технічний аналіз, який робить прогноз на основі статистики і історичних даних, залишився в минулому. Як показано на рисунку 1.1, річний розподіл категорії фундаментального аналізу показує збільшення публікацій з часом, особливо з 2019 по 2020 рік, тоді як технічний аналіз з часом стає все менш популярними. Це пов’язано з тим, що соціальні мережі стали невід’ємною частиною нашого життя, а онлайн новини є основним джерелом інформації про те, що відбувається в світі.

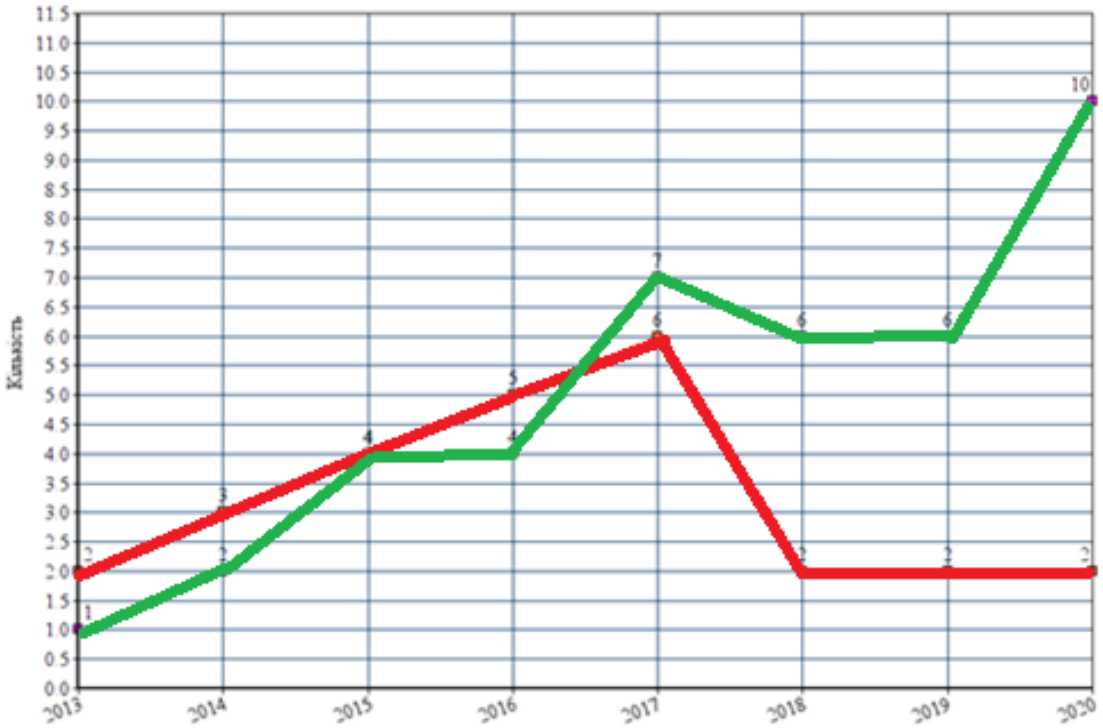


Рисунок 1.1 – Розподіл досліджень за підходами

Отже, відповідь на ДП4 полягає в тому, що основним підходом до отримання даних для прогнозування є фундаментальний аналіз, а джерелами даних для дослідження та створення часових рядів є соціальні мережі, новини і тд.

Для повідомлення про частоту та розподіл вибраних досліджень відповідно до їх категорій та дати було створено бульбашковий графік, таким чином надавши карту досліджень у прогнозуванні фондового ринку. Пухирчасті діаграми – це, по суті, дві діаграми розсіювання x-y з бульбашками на перетинах категорій. Розмір кожного бульбашки визначається кількістю первинних досліджень, які були класифіковані як такі, що належать до пари категорій, що відповідають координатам бульбашки. Отримана карта показана на рисунку 1.2.

Як показано на рисунку 1.2, аспекти, які ми використали для організації карти — це категорія, дослідження та рік публікації. З огляду на малюнок видно, що більшість відібраних досліджень використовували прогнозування фондового ринку через нейронні та стохастичні мережі як цільову платформу.

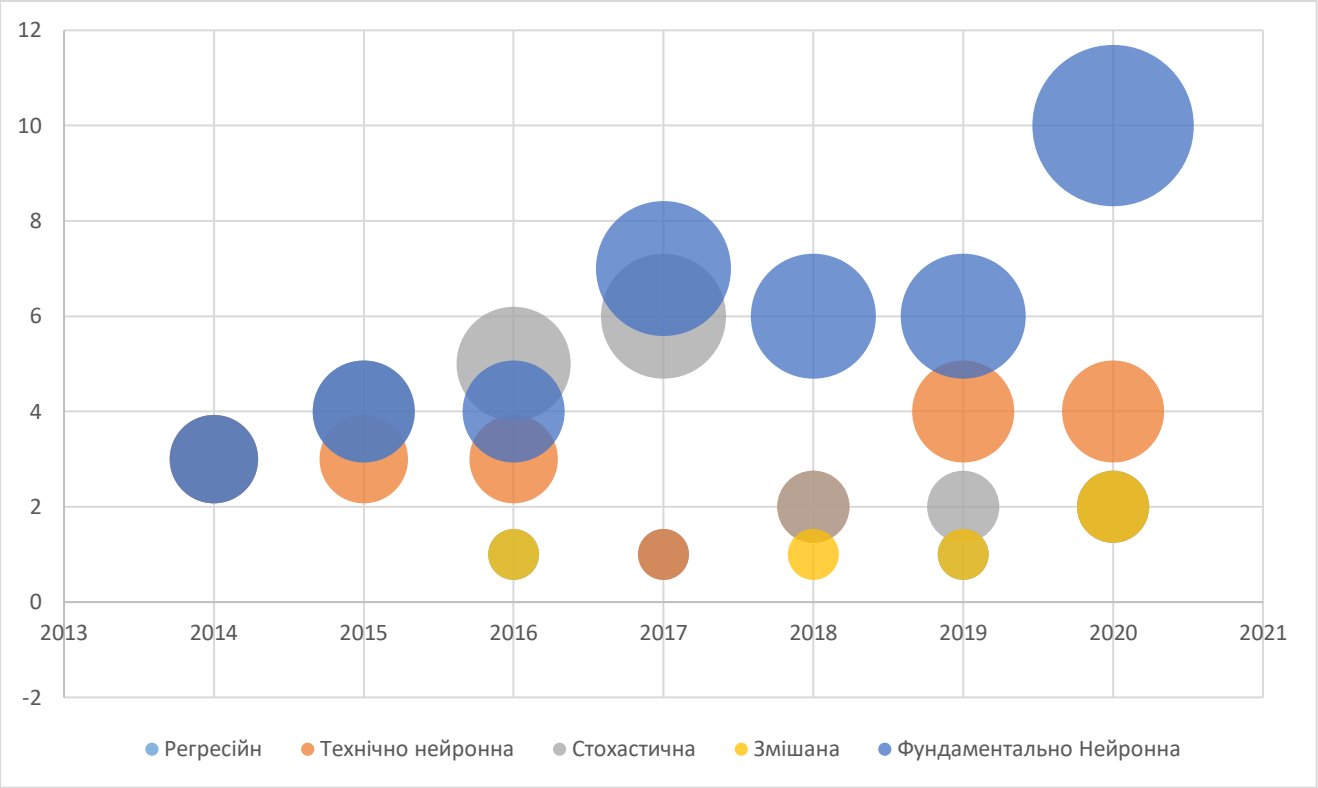


Рисунок 1.2 - Частота та розподіл вибраних досліджень відповідно до категорій

Отже, відповідь на ДП2 полягає в тому, що найбільша кількість програмних забезпечень для прогнозування фондового ринку, пов'язані з використанням нейронних мереж. Навпаки, категорії з меншою кількістю досліджень – це системи основані на регресійно опорному векторі та системи які стохастичну модель для передбачення. За останні 3 роки велика увага приділялася нейронним мережам, тоді як стохастичним методам все менше.

1.4 Обговорення

Наразі фінансовий ринок є шумною, непараметричною динамічною системою і існує два основних види методів прогнозування курсу акцій: традиційний метод аналізу та метод машинного навчання.

Традиційні економетричні методи або рівняння з параметрами не підходять для аналізу складних, масивних і шумних даних фінансових рядів.

На основі статистики та теорії ймовірності деякі вчені використовують моделі лінійного прогнозування часових рядів, такі як : векторна авторегресія (VAR), байєсовська векторна авторегресія (BVAR), авторегресійний інтегрований режим ковзного середнього (ARIMA) та узагальнена авторегресивна умовна модель гетероскедастичності (GARCH) [1]. Однак точність використання лише моделі часового ряду піддається сумніву через невизначеність та високі шумові характеристики фінансових часових рядів, а зв'язок між незалежними змінними та залежними змінними схильний до динамічних змін у часі, що обмежує подальше застосування та розширення.

В останні роки нейронна мережа стала "гарячим" напрямком досліджень у сфері прогнозування, оскільки вона може витягувати характеристики із великої кількості високочастотних необроблених даних, не покладаючись на попередні знання. У 1988 році Х. Уайт використав нейронну мережу для прогнозування акцій IBM, але експериментальні результати були не точним [3]. У 2003 році Л. Чжан використовував нейронну мережу та авторегресивну інтегровану модель ковзного середнього (ARIMA) для прогнозування акцій. Експериментальні результати показують, що нейронна мережа має очевидні переваги в нелінійному прогнозуванні даних, але точність ще потребує підвищення [4]. У 2005 році В. Сан запропонував метод прогнозування часових рядів на основі нейронної мережі. Цей метод поєднує в собі оптимальний алгоритм розподілу (OPA) і радіальну базисну функцію (RBF) нейронної мережі, хоча результат все ще не був достатньо точним [5]. У 2014 році М. Адкіхкарі. запропонував метод, що поєднує випадкове блукання (RW) і штучну нейронну мережу (ANN) для прогнозування чотирьох фінансових часових рядів, і результати показали, що точність прогнозування мала певне покращення [6]. У 2018 році Л. Чжан запропонував мережеву структуру прогнозування курсу акцій на основі нейронної мережі LM-BP, яка покращила недоліки традиційного алгоритму навчання нейронної мережі BP – повільну швидкість навчання та низьку точність [7]. У 2018 році результати експериментів М. Ху. показують, що згортка нейронна мережа може передбачати часові ряди, а глибоке навчання більше підходить для вирішення проблеми часових рядів. Однак, оскільки CNN частіше використовується для

розпізнавання зображень і виділення ознак, точність прогнозування лише CNN є відносно низькою [8] У 2020 році Хуан. створив високоточну модель короткострокового прогнозування часових рядів фінансового ринку на основі глибокої нейронної мережі LSTM та порівняв з нейронною мережею BP, традиційною RNN та вдосконаленою глибокою нейронною мережею LSTM. Результати показали, що глибока нейронна мережа LSTM має високу точність прогнозування і може ефективно передбачати часові ряди фондового ринку.

В даний час поєднання переваг різних методів та використання різних найкращих алгоритмів для створення гібридного методу є тенденцією розвитку глибокого навчання фінансових часових рядів [9]. Тому, щоб найкраще використовувати характеристики часових рядів, глибоко вивчити особливості даних і підвищити точність прогнозування ціни акцій, у цій роботі буде використано метод прогнозування ціни акцій на основі метода CNN-LSTM. Поєднання переваг згорткової нейронної мережі (CNN), яка може витягувати релевантні характеристики з даних, і мережі довгострокової пам'яті (LSTM), яка може знайти взаємозалежність даних часових рядів та автоматично визначити найкращий режим для відповідних даних, створює новий метод який може ефективно підвищити точність прогнозування ціни акцій.

Є певні обмеження для прогнозування тенденції курсу акцій використовуючи тільки одиночні моделі прогнозування лінійного часового ряду або тільки одиночні моделі нейронної мережі. З огляду на існуючі дослідження, було виявлено, що найкращим методом для передбачення часових рядів з високою точністю є LSTM, а поєднавши її зі згортковою мережею CNN для отримання взаємозв’язка між характеристиками даних соціальних мереж, та характеристиками акцій, можна отримати новий метод, який значно підвищить точність передбачення курсу акцій.

Тому відповіддю на ДП3 буде, архітектура яка базується на нейронних мережах комбінацію CNN і LSTM, а для збільшення продуктивності системи, вона буде використовувати велику кількість даних з соціальних мереж для навчання,.

Висновки до розділу

В даному розділі були поставленні дослідницькі питання, на які було дано відповідь після аналізу існуючих досліджень. Для аналізу існуючих досліджень було використано методологію “systematic mapping study”. Аналіз існуючих досліджень включав в собі:

— створення пошукових стрічок;

— створення критеріїв включення і виключення;

— категоризація;

— створення бульбашкової діаграми.

Провівши дослідження було надано відповідь на дослідницькі питання:

— відповідь на ДП1 буде, ПО на основі стохастичних моделей або нейронних мереж;

— відповідь на ДП2 буде фундаментальний аналіз;

— відповідь на ДП3 буде, архітектура яка базується на комбінації нейронних мережах CNN і LSTM;

— відповідь на ДП4 полягає в тому, що основним підходом до отримання даних для прогнозування є фундаментальний аналіз, а джерелами даних для дослідження та створення часових рядів є соціальні мережі, новини.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Вибір архітектури

Система для прогнозування курсу акцій на основі даних з соціальних мереж, являє собою набір процесів, ціль яких є встановлення майбутнього курсу акцій вказаної компанії, використовуючи історичні дані курсу акції та відгуки про компанію з твітера.

Найпопулярнішою та найефективнішою архітектурою для такої системи є мікросервісна арітектура. Мікросервісна архітектура – стиль розробки ПЗ, що полягає у розбиття моноліту системи на окремі компоненти, які є незалежними сервісами.

Кожен мікросервіс – це невелика монолітна програма, яка виконує свою функцію. У програмний продукт на основі мікросервісної архітектури можна додавати будь-яку кількість нових мікросервісів, розширюючи його функціональність. Щоб досягти такого в монолітній програмі, необхідно вносити зміни до основного продукту.

Переваги мікросервісної архітектури:

— висока стійкість до відмов: при падінні одного з сервісів, всі інші залишаються в строю. Таким чином, проблеми в окремих сервісах не завадять всьому робочому процесу;

— гнучкість: впровадження нового функціоналу не викликає труднощів, сервіси можуть писатися на різних мовах програмування;

— простота: чим менше логікі в дному модулі, тим простіше розібратися, що як працює;

— масштабованість: необхідні сервіси можна доповнити та розширити, коли з'явиться така необхідність. Вся система при цьому залишається незмінною і не потребую налаштування;

— децентралізація даних: кожному мікросервісу відповідає своя база даних. Всі дані залишаються ізольованими.

Недоліки мікросервісної архітектури:

— складність розробки. Якщо потрібне швидке рішення (прототип, невелика програма, стислі терміни), то мікросервіси не підійдуть. Швидкість розробки – висока плата за доступність та модульність;

— складність підтримки. Кожен мікросервіс потребує окремого обслуговування, тому потрібен постійний автоматичний моніторинг.

Приклад мікросервісної архітектури зображено на рисунку 2.1.

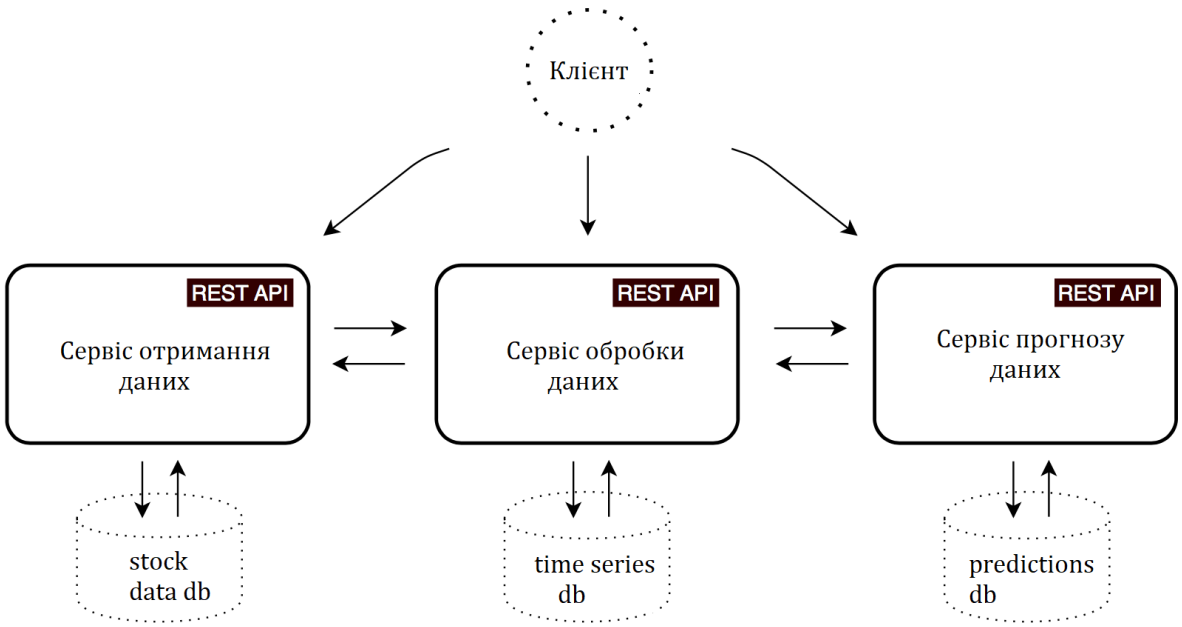


Рисунок 2.1 - Приклад мікросервісної архітектури

Зважаючи на всі переваги мікросервісної архітектури, вона є необхідним рішенням для системи прогнозування курсу акцій, оскільки однією з вимог продукту, є здатність до додавання нових джерел даних і методів їх обробки. Також завдяки такому підходу, ми зможемо гарантувати легкий поріг входу для подальших досліджень і лояльність до великого навантаження на систему.

2.2 Функціональні вимоги до системи

Система прогнозування ринку акцій на основі даних з соціальних мереж, повинна включати в собі блоки отримання, обробки та аналізу інформації, для створення прогнозу курсу акцій.

Першим і основним етапом цієї системи є отримання даних. В даному продукту, для прогнозування використовується одразу два великих джерела даних:

— дані з соціальної мережі Twitter за заданими параметрами і словами;

— дані про акції компанії за допомогою сервіса Yahoo Finance.

Також користувач повинен мати змогу задати ключові слова за якими система буде також шукати інформацію, окрім назви компанії.

Ще одним важливим етапом є задання слів виключень, які система повинна опрацювати і забезпечити видалення всіх отриманих даних , які мають слова виключення.

Щоб зробити дані ефективнішими, ми повинні надавати можливість користувачеві, вибирати період часу за який система буде опрацьовувати історичні дні, а також проміжок часу який необхідно спрогнозувати. Чим більший період отримання історичних даних, тим довше система буде їх оброблювати. А чим менший період для прогнозування, тим точніший буде результат.

Отримані дані з соціальних мереж включать в себе великі об’єкти, які містять занадто детальну інформацію про твіт. Для того, щоб пришвидшити процес прогнозування, потрібно зменшити об’єм даних, та видалити всі зайві параметри.

Процес збільшення точності інформації відбувається, через виконання наступних кроків:

— видалення шумів;

— контекстне видалення;

— аналіз настрою.

Видалення шумів передбачає видалення специфічних для твітера символів, пробілів, посилань, смайлів, тощо.

Для отримання більш якісних даних, використовується контекстне видалення, яке видаляє дані, які відповідають параметрам пошуку, але не мають значення для цін на акції, наприклад, у разі отримання твітів про фрукт «Apple», коли намір передбачити Apple ціна акцій.

Також важливим моментом є оцінка настрою твіта, потрібно зрозуміти чи є цей твіт негативним чи позитивним.

Так як ми використовуємо фундаментальний підхід, для прогнозування ринку акцій, на основі соціальних мереж, одним з найважливіших етапів системи є встановлення зв'язка між характеристиками текстових даних з соціальних мереж та характеристиками акцій. Потрібно визначити, як і які функції текстових даних взаємодіють з певними характеристиками даних про акції. Цей блок потребує використання машинного навчання

І найбільш важливий модуль системи - модуль прогнозування. Отримавши оброблені та проаналізовані дані у вигляді часових рядів, використовуючи нейронну мережу, ми можемо спрогнозувати подальший курс акцій. Діаграма процесу передбачення ринку акцій на основі даних з соціальних мереж відображена в додатку В. Ця система повинна містити лише одну роль - користувач. В додатку А наведено діаграму прецедентів для ролі користувача.

2.3 Не функціональні вимоги до системи

Проаналізувавши функціональні вимоги до системи, можна вивести наступні нефункціональні вимоги до програмного забезпечення:

— простота інтерфейсу – зробити інтерфейс настільки простим наскільки це можливо, щоб зробити сервіс доступний для будь якого користувача;

— мікросервісний підхід – процес розробки повинен бути у формі окремих сервісів, для того, щоб була можливість легко і просто додавати нові сервіси. Для того, щоб зробити процес розробки та відладки більш зручним;

— масштабованість – система повинна підтримувати можливість додавання нових джерел даних та методів передбачення;

— розширюваність – система повинна бути розширюваною для безпроблемного збільшення потоку кількості даних;

— кросплатформність – здатність системи бути розміщеною на будь якій операційній системі також буде плюсом.

2.4 Методи прогнозування

В даній системі для прогнозування курсу акцій на основі даних з соціальних мереж, було використано гібридний метод CNN-LSTM, який включає в себе поєднання двох нейронних мереж:

— CNN - для скорочення розмірності шляхом агрегування даних соціальних мереж і курсом акції в часовий ряд відповідно до їх кореляційних коефіцієнтів;

— LSTM - для прогнозування курсу акцій на основі результуючих даних мережі CNN.

Використання цього методу повинно підвищити ефективність та точність прогнозування ринку акцій, відносно інших методів.

2.4.1 CNN

CNN – згорткова нейронна мережа, потужний інструмент машинного навчання, націлений на ефективне розпізнавання та класифікацію, в першу чергу, зображень [10]. Однак ефективність скорочення розмірності дозволяє використовувати її також для покращення часових рядів [11]. Обчислення згорткових нейронних мереж засноване на роботі з тензорами [12]. Під тензором розуміється багатовимірний масив чисел. Оперування тензорами, а не матрицями дозволяє не втрачати інформацію при переході до меншої розмірності. В даній системі, мережа використовується для обробки вхідних даних для покращення остаточного результату, шляхом агрегування кількох змінних(історичних даних, провідних показників), щоб потім передати їх до мережі LSTM

Розглянемо на прикладі, принцип роботи цієї мережі. Отже, ми маємо завдання виділити на картинці якийсь об'єкт, наприклад, автомобіль. Людина легко розуміє,

що перед ним автомобіль і розпізнає його по тисячі дрібних ознак. Для мережі це безліч пікселів. Чим більша якість зображення, тим важче мережі обробити картинку. Саме тому, мережа використовуючи алгоритм згортки, зменшує кількість символів, при цьому зберігаючи характеристики об’єкта.

Згорткова нейронна мережа за рахунок застосування спеціальної операції - власне згортки - дозволяє одночасно зменшити кількість інформації, що зберігається в пам'яті, за рахунок чого краще справляється з картинками вищого розширення, і виділити опорні ознаки зображення, такі як ребра, контури або грані[13]. На наступному рівні обробки з цих ребер і граней можна розпізнати фрагменти текстур, що повторюються, які далі можуть скластися в фрагменти зображення. Приклад роботи мережі зображено на рисунку 2.2.

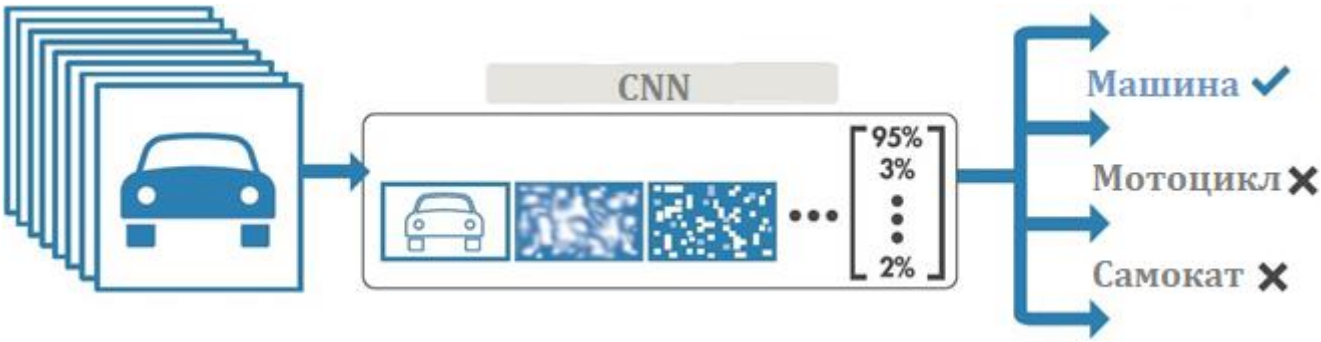


Рисунок 2.2 - Робота згорткової нейронної мережі

Згортка - універсальна операція. Її можна застосувати для будь-якого сигналу, будь то дані з датчиків, аудіосигнал або прогнозування ринку [14]. Головною задачею буде агрегування характеристик даних, до звичайного часового ряду, враховуючи кореляційні коефіцієнти.

В нашому випадку, ми має величезні масиви характеристик даних з соціальних мереж та акцій компанії:

— кількість підписників власника твіту;

— кількість оцінок твіту;

— контекстна оцінка;

— кількість коментарів;

— кількість переглядів ;

— кількість ретвітів;

— ціна відкриття;

— найвища ціна;

— найнижча ціна;

— обсяг;

— ціна закриття;

— підйом;

— падіння;

— зміна за день.

Всі ці показники являть собою масиви даних. Для того, щоб алгоритми прогнозування працювали ефективніше нам потрібно звести характеристики цих об’єктів до найменш можливого виду, при цьому зберегти кореляцію між даними та їх характеристики. Зважаючи на розглянуту інформацію, можна зробити висновок, що використання CNN – не надає достатньої можливості для прогнозування ринку акцій, проте є досить потужим інструментом, для підготовки даних до наступної нейронної мережі. За допомогою операції згортки, ми може отримати досить інформативний часовий ряд для подальшого передбачення, наприклад за допомогою LSTM.

2.4.2 LSTM

LSTM - тип рекурентної нейронної мережі, здатний вчитися довгостроковим залежностям [15]. Цей тип, є найкращим варіантом рекурентних мереж для прогнозування часових рядів.

Рекурентні нейронні мережі додають пам'ять до штучних нейронних мереж, але пам'ять, що реалізується, виходить короткою — на кожному кроці навчання інформація в пам'яті поєднується з новою і через кілька ітерацій повністю перезаписується [16].

Мережа широко використовується в розпізнаванні мовлення, емоційному аналізі та аналізі тексту, оскільки має власну пам’ять і може робити відносно точне

прогнозування [17,18]. Останніми роками метод також був прийнятий у сфері прогнозування фондового ринку. Комірка пам’яті LSTM складається з трьох частин: гейта забуття, вхідного гейта та вихідного гейта, як показано на рисунку 2.3.

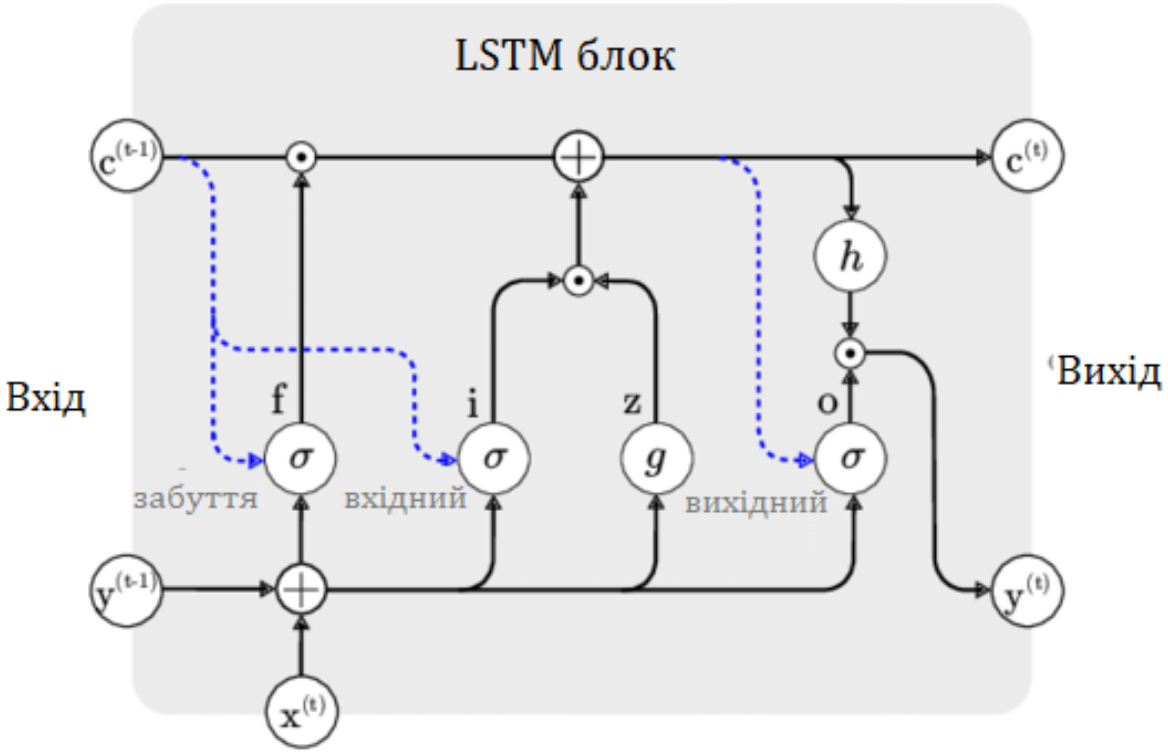


Рисунок 2.3 – Блок LSTM

Процес обчислення LSTM виглядає наступним чином:

На першому етапі LSTM потрібно вирішити, яку інформацію ми збираємось викинути зі стану комірки. Це рішення приймається сигмовидним шаром, званим шаром гейту виходу. Він отримує на вхід h і x і видає число від 0 до 1 для кожного номера в стані комірки C. 1 означає "повністю зберегти", а 0 - "повністю видалити" . Сигмовидний шар обчислюється за формулою 2.1.

$$f_t = \partial(W_t|h_{t-1}, x_t| + b_f) \quad (2.1)$$

де діапазон значень f дорівнює (0,1), W - вага гейта забуття, а bf – зміщення гейта забуття, x – це вхідне значення поточного часу, а h – це вихідне значення останнього моменту

На наступному етапі необхідно вирішити, яку нову інформацію зберегти в стані комірки. Розіб'ємо процес на дві частини. Спочатку сигмовидний шар, званий шаром гейту входу, вирішує, які значення потрібно оновити за формулою 2.2. Потім шар tanh створює вектор нових значень-кандидатів S, які додаються до поточного стану за формулою 2.3.

$$i_t = \sigma(W_i|h_{t-1}, x_t| + b_i), \quad (2.2)$$

$$S_t = \tanh(W_c|h_{t-1}, x_t| + b_c) \quad (2.3)$$

де діапазон значень σ ((0,1), Wi — вага вхідного кандидата, bi - зміщення вхідного гейта, Wc є вага кандидата вхідного гейта, а bc є зміщення даних кандидата вхідного гейта.

Оновлення поточного стану клітинки відбувається за формулою 2.4.

$$C_t = f_t * C_{t-1} + i_t * S_t \quad (2.4)$$

де діапазон значень Ct = (0,1).

Зрештою, треба вирішити, що хочемо отримати на виході. Результат буде відфільтрованим станом комірки. Спочатку запускаємо сигмовидний шар, який вирішує, які частини стану осередку виводити, вираховується за формулою 2.5.

$$\sigma_1 = \sigma W_0[h_{t-1}, x] + b_0 \quad (2.5)$$

де діапазон значень σ (0,1), W є вагою вихідного гейта, а bo — зміщення виходу

Потім пропускаємо стан комірки через tanh (щоб розмістити всі значення в інтервалі [-1, 1]) та множимо його на вихідний сигнал сигмоподібного гейту.

$$h_t = o_t * \tanh(C_t) \qquad (2.6)$$

Отриманий результат представляє собою часовий ряд, створений на основі історичних даних

2.4.3 CNN-LSTM

Поєднавши мережу CNN та LSTM ми отримаєм новий метод CNN-LSTM, який повинен покращити прогнозування часових рядів. Процес навчання та прогнозування CNN-LSTM показаний на рисунку 2.4.

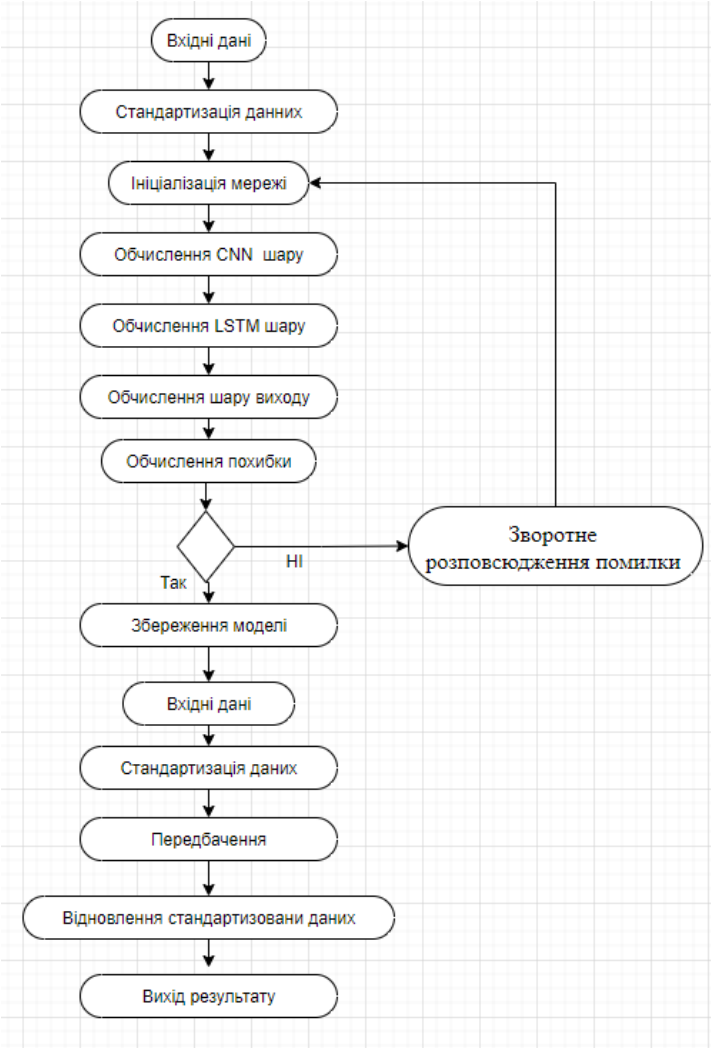


Рисунок 2.4 – Процес навчання та прогнозування

Основні кроки такі:

— вхідні дані: дані, необхідні для навчання CLSTM;

— стандартизація даних: оскільки існує великий пробіл у вхідних даних. Щоб краще тренувати модель, для стандартизації вхідних даних використовується метод стандартизації z-score, як показано в наступній формулі 2.7;

$$y_i = \frac{x_i - z}{s} \qquad (2.7)$$

— ініціалізація мережі: ініціалізація ваги та зміщення кожного рівня CLSTM;

— обчислення CNN шару: вхідні дані послідовно пропускаються через шар згортки та шар об’єднання в шарі CNN, виконується вилучення ознак вхідних даних;

— обчислення LSTM шару: вихідні дані шару CNN обчислюються через шар LSTM;

— обчислення вихідного шару: вихідні дані шару LSTM виводиться в загальний рівень для отримання фінального значення даних;

— обчислення похибки: вихідне значення, отримане з вихідного шару, порівнюється з реальним значенням;

— перевірка на виконання умов: виконано попередньо визначену кількість циклів, рівень помилок прогнозування нижчий за певний поріг. Якщо одна з умов завершення буде не виконана, навчання буде не завершено, оновлюється вся мережа CLSTM і переходимо до кроку 9, інакше переходимо до кроку 10;

— зворотне розповсюдження помилки: оновлення ваги та зміщення кожного шару та перехід до кроку 4, щоб продовжити навчання мережі;

— збереження моделі: зберегти навчену модель для прогнозування.

— вхідні дані: вхідні дані, необхідні для прогнозування;

— стандартизація даних: вхідні дані стандартизуються за формулою 2.8;

— Процес прогнозування: вхід стандартизованих даних в навчену модель CLSTM, та отримання вихідного значення;

— стандартизоване відновлення даних: вихідне значення, отримане за допомогою моделі CLSTM, є стандартизованим значенням, а стандартизоване значення відновлюється до вихідного значення. Як показано в наступній формулі 2.8;

$$x_i = y_i * \mathrm{s} + \mathrm{z} \qquad (2.8)$$

де – y стандартизоване значення, – x вхідні дані, – z середнє значення вхідних даних, s – стандартне відхилення вхідних даних.

— вихід результату: вивести результати для завершення процесу прогнозування.

2.5 Структура системи

На основі функціональних та не функціональних вимог, було розроблено структуру проекту. Система прогнозування курсу акцій фондового ринку з основою на соціальних мережах, складається з таких компонентів:

— сервіс отримання даних про курс акцій, повинен повертати історичну інформацію про курс акцій;

— сервіс отримання даних з соціальних мереж, повинен збирати інформацію про компанію за певний період;

— сервіс обробки даних, повинен відсіювати слова добавлені твітером, посилання, тощо;

— сервіса контекстного видалення даних, повинен видаляти дані які включають слова виключення;

— сервіс емоційної оцінки твіта, повинен за допомогою бібліотеки Vader, визначати коефіцієнт враження;

— сервіс кореляції даних, повинен знаходити взаємозв’язок між даними курсу акцій, та даними з соціальних мереж;

— сервіс прогнозування, повинен прогнозувати курс акцій, на основі даних отриманих з кореляційного сервісу;

— клієнтська частина, яка надає необхідну інформацію сервісами через зручний інтерфейс;

— база даних, для збереження історичних даних курсу акцій, дані соціальних мереж та результату дослідження.

Структурна схема системи прогнозування ринку акцій відображена на рисунку 2.5. Структурна схема сервісів відображена в додатку Г.

Висновки до розділу

В цьому розділі були створені функціональні та не функціональні вимоги для створення системи прогнозу ринку акцій, на основі даних з соціальних мереж. На основі цих вимог було створено структурну схему системи, створено діаграму процесів прогнозування курсу акції за допомогою соціальних мереж, та відображено об’єкти системи в діаграмі класів.

Аналізуючи фінансові часові ряди цін акцій фондового ринку, буде використано гібридний метод глибокого навчання (CNN-LSTM) для прогнозування ціни активу. У цьому методі CNN використовується для вилучення характеристик даних, а LSTM використовується для прогнозування даних. Метод може повністю використовувати часову послідовність даних ціни акції для отримання більш надійного прогнозування.

3 РОЗРОБКА СИСТЕМИ

3.1 Вибір технологій розробки

Ця робота складається з клієнтської частини та групи сервісів, які виконують математичні та статистичні операції. Інструменти, які використовуються для цієї системи:

— Python - мова програмування, яка відома багатьма науковими бібліотеками про аналіз даних та математичних операцій [19]. Найкращий варіант для мікросервісів. Його зручно читати та легко застосовувати. Крім того, Python володіє широким функціоналом для веб-розробки та динамічною екосистемою сторонніх модулів для будь-яких потреб. До цих потреб відноситься підключення до інших систем, наприклад, до баз даних, зовнішніх API тощо;

— Flask - фреймворк на Python, який має багато вбудованих можливостей та легкий у використанні разом з мікросервісним підходом;

— Numpy - пакет Python, який обробляє математичні операції та операції з масивами в Python. Основним типом numpy є «ndarray», який є N-вимірним масивом;

— SciKit-Learn: найвідоміший пакет у світі для аналізу даних. Він підтримує та реалізує більшість моделей машинного навчання, включаючи лінійні моделі, нейронні мережі, KNN, операції кластеризації [20].;

— Pandas: широко використовуваний пакет Python для обробки та зберігання даних. Він має багато операцій, пов’язаних з даними, як-от додавання стовпців, об’єднання наборів даних або операції з діапазоном дат для часових рядів;

— Tensorflow: відкрита програмна бібліотека для машинного навчання, розроблена Google для вирішення завдань побудови та тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття;

— Kerras - відкрита нейромережна бібліотека, написана мовою Python, яка здатна працювати поверх TensorFlow, спроектована для уможливлення швидких експериментів з мережами глибинного навчання [21];

— Angular - це фреймворк для розробки веб сайтів, створений на JavaScript, який надає розробникам можливість створення клієнтської сторони системи [22]. Основною причиною вибору Angular є те, що цей фреймворк оснащений компонентами, які допомагають розробникам писати ефективний, повторно-використовуваний та надійний для користування код;

— Docker часто рекламується як один із найважливіших інструментів для мікросервісів. Він дозволяє інкапсулювати та копіювати додаток у зручних стандартизованих контейнерах [23]. Це зменшує невизначеність та складність середовища. Також це значно полегшує перехід від розробки до виробництва додатків. Ви можете розмістити кілька контейнерів у різних середовищах (навіть у різних операційних системах) в одній фізичній коробці або віртуальній машині;

— Kubernetes дозволяє розгортати кілька контейнерів Docker, які працюють скоординовано [24]. Це змушує розробників мислити кластеризовано, пам'ятаючи про виробниче середовище. Також це дозволяє визначити кластер за допомогою коду, щоб нові розгортання або зміни конфігурації визначалися у файлах. Все це робить можливими методи на кшталт GitOps, зберігаючи при цьому повну конфігурацію в системі управління версіями. Кожна зміна вноситься способом легким для відновлення, оскільки вона являє собою регулярне git-злиття. Завдяки цьому можна дуже легко відновлювати чи дублювати інфраструктуру;

— Git - система управління версіями з розподіленою архітектурою. На відміну від колись популярних систем на кшталт CVS і Subversion (SVN), де повна історія версій проекту доступна лише в одному місці, у Git кожна робоча копія коду сама по собі є репозиторієм. Це дозволяє всім розробникам зберігати історію змін у повному обсязі [25].

Дана система являє собою набір мікросервісів з якими користувач може взаємодіяти через клієнтську частину. Найкращим технологічним стеком для цієї архітектури є Docker, Kubernetes і Python. Діаграма класів відображена в додатку Б.

3.2 Сервіс отримання даних

Основним елементом для прогнозування ринку акцій є нейронні мережі. Проте для того, щоб вони працювали в необхідному руслі, вони повинні отримати велику кількість даних для навчання. Враховуючи особливості цієї роботи, для ефективного передбачення ринку акцій нам потрібні одразу два джерела даних : дані з соціальних мереж, та історичні дані про курс акцій.

Найкращим інструментом для отримання необхідних даних з соціальних мереж, виявився Twitter, який пропонує два основних типи доступу до їхніх даних: Search API і Stream API.

Search API використовує групу ключів та ідентифікаторів, щоб знайти групу твітів, що зберігаються як історичні дані.

Stream API повертає безперервний потік даних у реальному часі, який фільтрує вхідні твіти за допомогою групи ключових слів або термінів. Результатом є об’єкт, який містить дуже детальну інформацію про твіт : текст, час, інформацію про користувача, географічне розташування тощо. Об’єкт твіту зображений в додатку Ґ. Щоб отримати доступ до цих даних, потрібно зареєструвати обліковий запис розробника в Twitter і отримати ключ, для доступу до API. Максимальна кількість твітів, які можна отримати за допомогою стандартного API-інтерфейсу Twitter Stream, становить 5 мільйонів твітів в день, що є досить значним показником для отримання статистики.

Іншим важливим джерелом даних є інформація про акції. Найкращим інструментом для отримання даних про акції є Yahoo Finance. Цей сервіс не є ідеальним джерелом, оскільки багато компаній не розкривають свої фінансові дані і публікують їх на своїх сайтах, проте це найкращий сервіс, який отримує дані про ціни на акції в режимі реального часу. Також сервіс дає можливість отримувати історичну інформацію про різноманітні компанії

В клієнтській частині це відображено, як строка для пошуку компаній за назвою, та кнопкою запуску, яка запустить процес збору інформації в режимі реального часу.

Інтерфейс для взаємодії з цим модулем зображений на рисунку 3.1.

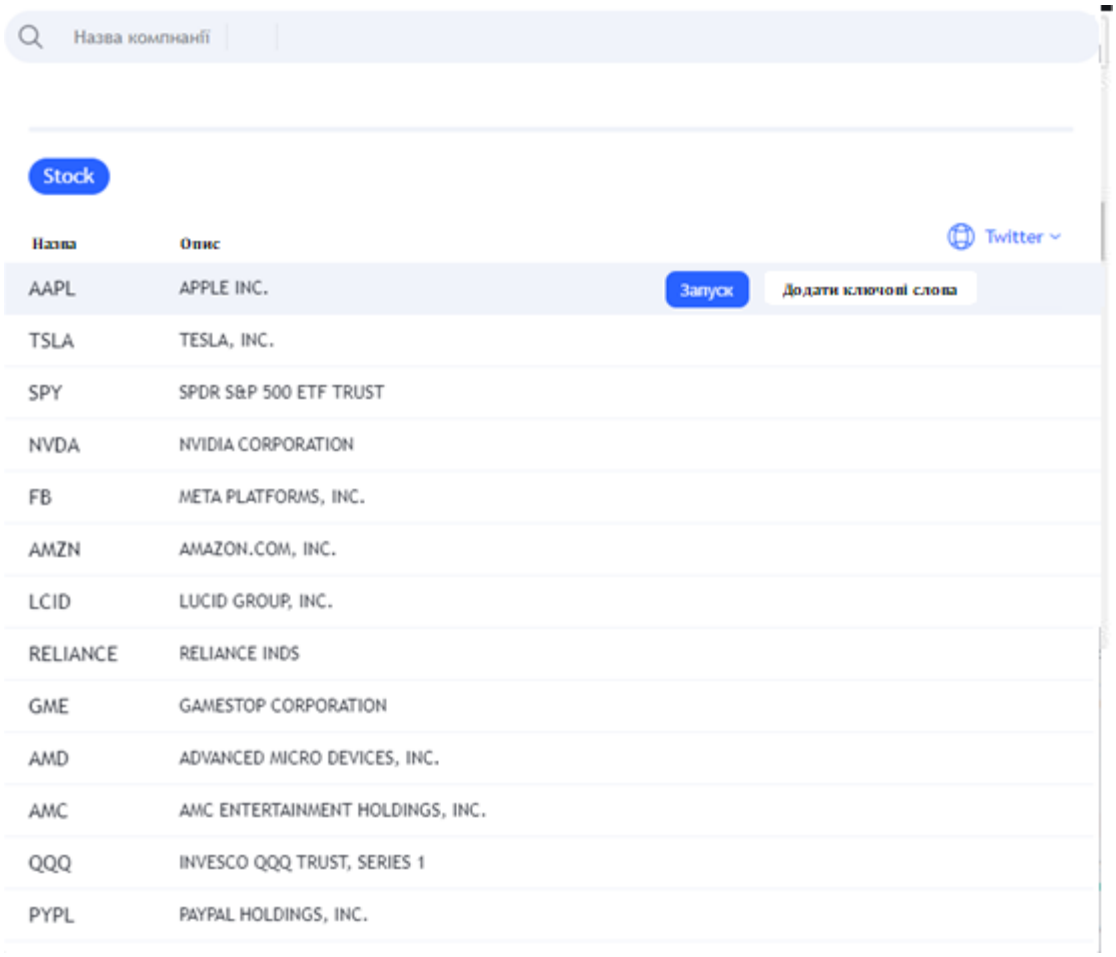
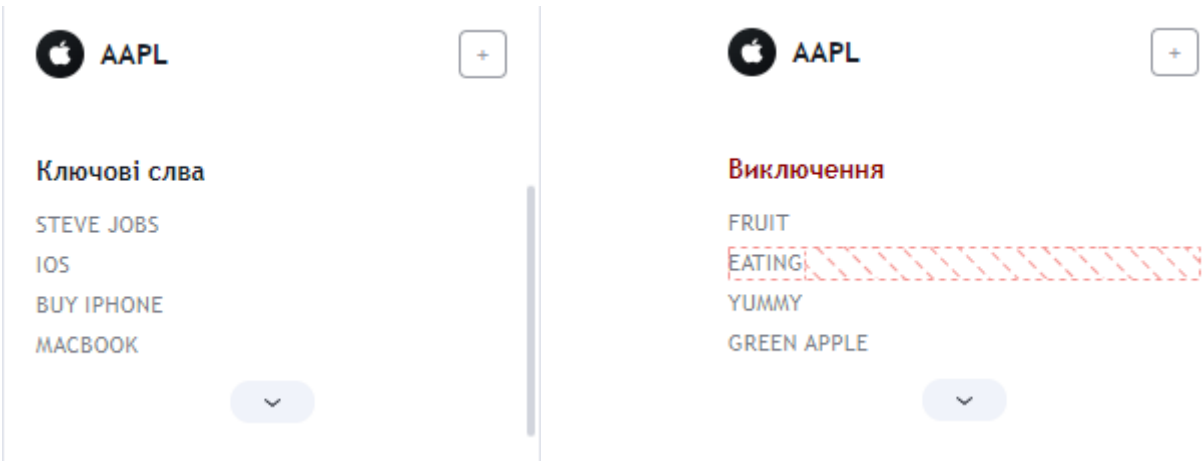


Рисунок 3.1 - Інтерфейс запуску програми

Також для кожної компанії, можна додати список ключових слів, для того, щоб збільшити точність пошуку даних в твітері. А також додати список виключень, для того, щоб виключити ці слова з пошуку. Інтерфейс додавання слів відображений на рисунках 3.2 – 3.3.



Рисунки 3.2 – 3.3 - Відображення ключових слів та слів виключення

3.3 Сервіс обробки даних

Дані, які ми отримуємо на попередньому етапі, необхідно обробити, перш ніж їх можна буде використовувати для прогнозування. Існує три категорії операцій для обробки даних.

Очищення даних - дані містять багато деталей, які не потрібні, певний шум. Наприклад, нам потрібно видалити ключові слова, пов’язані з Twitter (RT означає ретвіт), або URL -адреси чи згадки користувачів, смайли, розділові знаки, цифри та пробіли. Нам також потрібно видалити слова, які не додають жодного значення. Приклад очищення даних відображено на рисунку 3.4.

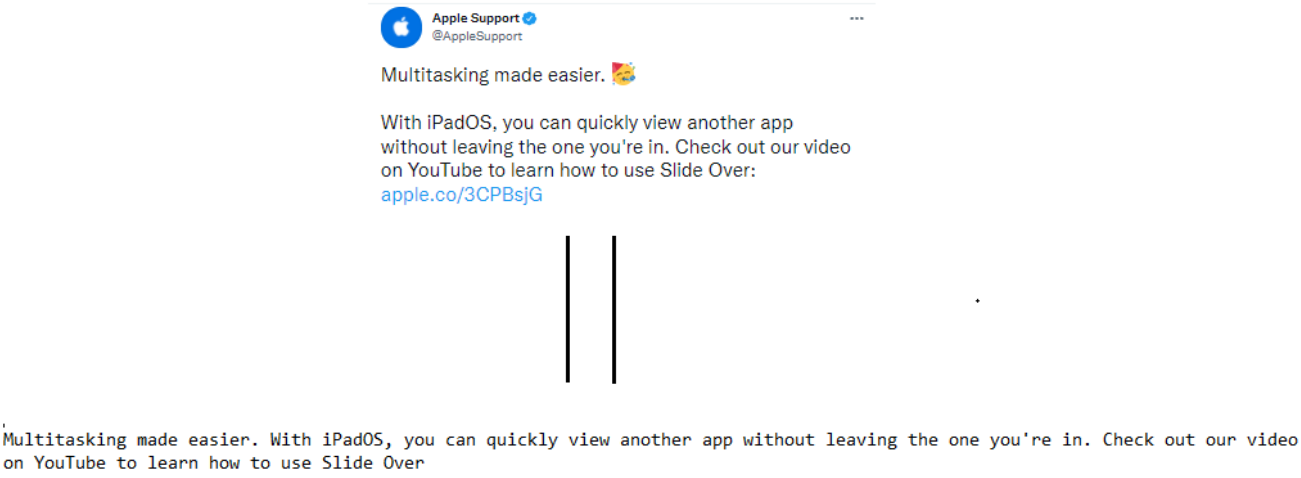


Рисунок 3.4 – Приклад очищення даних

Контекстне очищення - для отримання більш якісних даних, використовується контекстне видалення, яке видаляє дані, які відповідають параметрам пошуку, але не мають значення для цін на акції, наприклад, у разі отримання твітів про фрукт «Apple», коли намір передбачити Apple ціна акцій.

Оцінка даних - для було використано інструмент з відкритим кодом під назвою «VADER». Це інструмент, який спеціалізується на аналізі настроїв вмісту соціальних мереж. Результатом цього інструменту є оцінки тексту: позитивна оцінка, негативна оцінка та нейтральна оцінка. Алгоритм визначає складний бал, який коливається від -1 (найбільш негативний) до +1 (найбільш позитивний). Приклад роботи Vader зображено на рисунку 3.5.

```
1st statement :
Overall sentiment dictionary is :  {'neg': 0.165, 'neu': 0.588, 'pos': 0.247, 'compound': 0.5267}
sentence was rated as  16.5 % Negative
sentence was rated as  58.8 % Neutral
sentence was rated as  24.7 % Positive
Sentence Overall Rated As Positive

2nd Statement :
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
sentence was rated as  0.0 % Negative
sentence was rated as  100.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Neutral

3rd Statement :
Overall sentiment dictionary is :  {'neg': 0.508, 'neu': 0.492, 'pos': 0.0, 'compound': -0.4767}
sentence was rated as  50.8 % Negative
sentence was rated as  49.2 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Negative
```

Рисунок 3.5 – Оцінка текстових даних

3.4 Сервіс кореляції та аналізу причинності

Для того, щоб перевірити, як пов’язані дані з соціальних мереж на курс акцій, потрібен математичний інструмент, який може допомогти з вибором відповідних характеристик або частин даних, які можна використовувати на наступних етапах. Наприклад розглянемо структуру об’єкта, що повертається з API Twitter. Він має ідентифікатори користувача та твіту. Ці ідентифікатори є унікальними для кожного з твітів або користувачів, і, отже, не можуть дати нам ніякої цінності при прогнозуванні поведінки фондового ринку, оскільки ми шукаємо закономірність, і вона не може існувати з унікальними значеннями рядка, які не мають іншого значення. Саме тому ці значення вилучаються з об’єкту.

Для того, щоб визначити інші більш неоднозначні фактори, наприклад, кількість твітів у користувача, використовується кореляційний аналіз. Ми зацікавлені в пошуку зв’язку між змінною, яку ми хочемо передбачити (ціною акцій), і будь-якою іншою частиною зібраних нами даних. Зазвичай це лінійна залежність. Кореляційний аналіз, приймає як вхідні параметри - дві змінні і вивчає, як одна з них впливає на іншу. Результатом є коефіцієнт кореляції, який приймає значення від -1 до +1. Найбільш позитивне значення (+1) означає, що дві змінні позитивно корелюють, тобто збільшення одиниці в одній зі змінних призведе до збільшення на одну

одиницю в другій змінній, і те саме стосується зменшення. Якщо коефіцієнт кореляції дорівнює (-1), це означає, що змінні рухаються або поводяться в протилежних напрямках. Це означає, що збільшення одиниці в першій змінній призведе до зменшення одиниці у другій змінній, і навпаки. Значення від ±1 до 0 змінюють лише розмір зміни, а не напрямок. Коефіцієнт кореляції 0 означає, що дві змінні взагалі не пов’язані, і зміни однієї з них взагалі не впливають на іншу.

І хоча цей зв’язок можна розглядати як зв’язок між двома змінними, він іноді може помилятися, оскільки виходить за рамки контексту і може використовуватися нераціонально. Наприклад, ми точно знаємо, що витрати на розвиток науки і технологій не пов’язані з кількістю бездомних тварин, але ми можемо виявити, що існує дуже висока кореляція між витратами Україну на науку і технології та кількістю бездомних тварин. Співвідношення між витратами на науку та бездомними тваринами, зображено на рисунку 3.6.

Рисунок 3.6 – Співвідношення між витратами на науку та бездомними тваринами

Це випливає з того факту, що кореляція не означає причинно-наслідковий зв’язок, тобто ми не можемо вважати, що дві події викликані одна одною, лише тому, що вони відбуваються разом. Щоб встановити причинно-наслідковий зв’язок, нам потрібно використовувати інші методи, деякі з яких залежать від кореляції, наприклад, тест причинно-наслідкового зв’язку Грейнджера або конвергентне перехресне відображення. Тест Грейнджера на причинність — процедура перевірки причинно-наслідкового зв'язку («причинність по Грейнджеру») між тимчасовими

рядами. Ідея тесту полягає в тому, що значення (зміни) тимчасового ряду xt, що є причиною змін тимчасового ряду yt, повинні передувати змінам цього тимчасового ряду, і крім того, повинні робити значний внесок у прогноз його значень. Якщо кожна зі змінних робить значний внесок у прогноз іншої, то, можливо, існує деяка інша змінна, яка впливає на обидві. У тесті Грейнджера послідовно перевіряються дві нульові гіпотези: "x не є причиною у за Грейнджером" і "y не є причиною x за Грейнджером". В даній роботі, тест причинності Грейнджера реалізований за допомогою бібліотеки Pandas.

Попередньою умовою для тесту причинності по Грейнджеру є те, що тимчасовий ряд повинен бути стабільним, інакше можуть виникнути проблеми помилкової регресії. Отже, тест на одиничний корінь має виконуватися на стаціонарності часового ряду кожного індикатора перед тестом причинності Грейнджера.

3.5 Сервіс передбачення

Це ключовий сервіс серед усіх інших служб, оскільки він включає етапи створення моделі та прогнозування моделей. У цій службі користувач може створити передбачення, визначивши моделі, з якими він хоче експериментувати, проміжок в днях, які він хоче використати, і діапазон дат, у якому дані будуть розглядатися (скільки днів у майбутньому він хоче передбачити). Інтерфейс вибору параметрів для прогнозування зображений на рисунка 3.7 – 3.8.

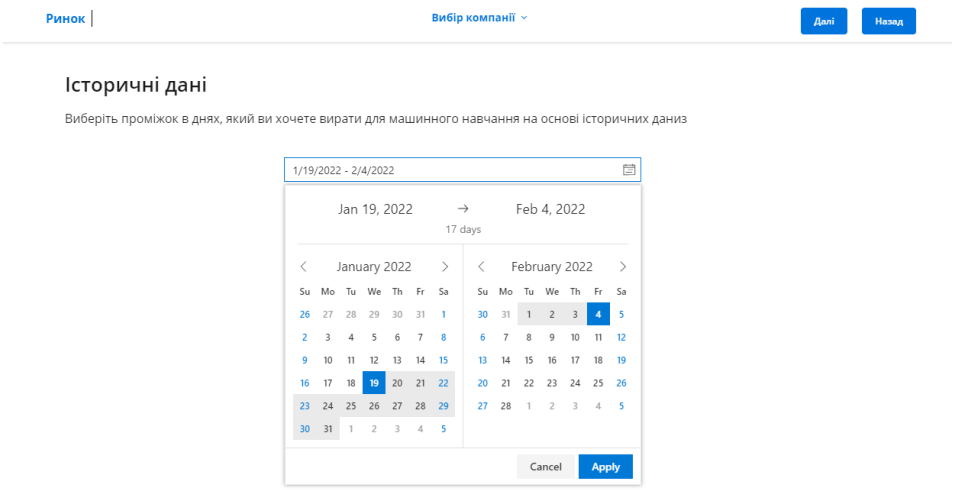


Рисунок 3.7 – Вибір проміжку дат для історичних даних

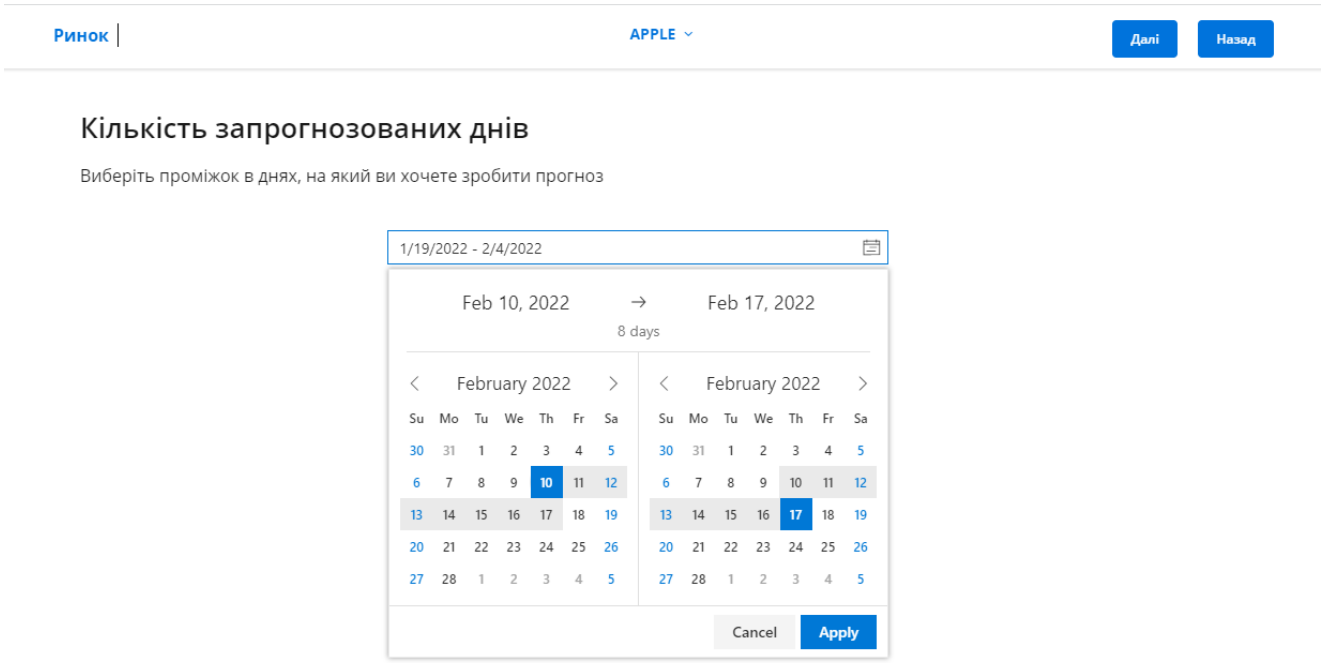


Рисунок 3.8 – Вибір проміжку дат, для якого система буде створювати прогнозування

Все це перенесе користувача на наступний етап, де почнеться виконання машинного навчання. Час затрачений на прогнозування залежить від кількості даних які потрібно проаналізувати. Прогнозування відбувається на основі гібридного методу CNN-LSTM. Спочатку CNN – отримує характеристики зв'язків між даними з соціальних мереж та курсом акцій, які ми отримали на етапі кореляційного аналізу, коефіцієнти емоційної оцінки повідомлень і агрегує їх до специфічного числового ряду. Після отримання певного числового ряду, LSTM – починає процес прогнозування, використовуючи як історичні дані про курс акцій, так і дані які ми отримали в даний момент.

Після того, як мережа CNN-LSTM закінчила процес аналізу, користувач зможе переглянути результати, за допомогою plotly створюється та відображається у вигляді графічного графіку та експортуються результати у формат JSON. Приклад прогнозованого графіку відображено на рисунку 3.9

Рисунок 3.9 – Спрогнозований курс акції

Висновки до розділу

В даному розділі було описано етап розробки програмної системи для прогнозування курсу акцій на основі соціальних мереж. Було описано створення таких сервісів як :

— сервіс отримання даних;

— сервіс обробки даних;

— сервіс кореляції та причинності;

— сервіс передбачення.

Також було зроблено оцінку метода для прогнозування. Порівнявши критерії оцінки CNN-LSTM на основі соціальних мереж, з багатошаровим персептроном (MLP), CNN, RNN, LSTM і CNN-LSTM, доведено, що CNN-LSTM на основі даних з соціальних мереж, має високу точність прогнозування і використовуючи дані з соціальних мереж, підвищує ефективність системи

4 ОЦІНКА РЕЗУЛЬТАТІВ

4.1 Оцінка ефективності системи

Щоб довести ефективність використання соціальних мереж для CNN-LSTM, було проведено порівняння з MLP, CNN, RNN, LSTM та CNN-LSTM (лише на історичних даних), використовуючи той самий навчальний набір і дані тестового набору в одному операційному середовищі.

Усі експерименти проводяться в робочому середовищі Intel i7-5500H, 18 GB RAM, 1 TB жорсткого диска та Windows 10. Було включено такі фактори впливу, як ціна відкриття, найвища ціна, найнижча ціна, ціна закриття, обсяг, оборот, підйоми та падіння, зміна за день.

У цьому експерименті в якості експериментальних даних для систем технічного аналізу, було вибрано Yahoo Finance, а для систем фундаментального аналізу, дані з Твітеру. Були використанні щоденні торгові дані з 1 вересня 2019 року по 30 грудня 2020 року.

Кожна частина даних містить вісім пунктів, а саме: ціна відкриття, найвища ціна, найнижча ціна, ціна закриття, обсяг, оборот, підйоми та падіння та зміна за день. Приклад даних наведено в таблиці 4.1.

Таблиця 4.1 - Вибіркові дані

Дата	Ціна відкриття	Найвища ціна	Ціна закриття	Оборот	Підйоми падіння	Зміна
20.12.2020	123.4	124.6	123.7	275982	-0.4	-0.24%
21.12.2020	124.2	125.1	124.1	252458	-0.6	+0.15%
22.12.2020	123.1	124.1	123.5	295922	-0.4	-0.10%
23.12.2020	122.1	123.1	122.5	315922	+0.4	-0.10%
24.12.2020	122.5	124.1	123.1	255922	+0.6	+0.15%
25.12.2020	123.1	124.1	124.6	325922	+1.5	+0.40%

Для оцінки ефекту прогнозування CNN-LSTM на основі соціальних мереж були використані критерії оцінки, такі як: середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE) і R-квадрат (R^2).

MAE розшифровується як Mean Absolute Error – середня абсолютна помилка. Розрахунок MAE відображено в формулі 4.1.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|g_i - ty_i| \quad (4.1)$$

де – g прогнозоване значення, y – справжнє значення. Чим менше значення MAE, тим краще прогнозування.

RMSE розшифровується як Root Mean Squared Error і перекладається як Середня квадратична помилка. Суть методу полягає в тому, щоб мінімізувати суму квадратів відхилень фактичних значень від розрахункових (SSE - "Sum of Squared Errors").

Розрахунок RMSE відображено за формулою 4.2.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(g_i - y_i)^2} \quad (4.2)$$

де – g прогнозоване значення, a – справжнє значення. Чим менше значення RMSE, тим краще прогнозування.

R2 - коефіцієнт детермінації. Характеризує ступінь подібності вихідних даних та передбачених.

Розрахунок R2 відображений в формулі 4.3.

$$R^2 = \frac{(\sum_{i=1}^{n}(g_i - y_i)^2)/n}{(\sum_{i=1}^{n}(z_i - tg_i)^2)/n} \quad (4.3)$$

де – g прогнозоване значення, – y справжнє значення, а z – середнє значення. Діапазон значень R2 становить (0,1).

Чим ближче значення MAE і RMSE до 0, тим менша похибка між прогнозованим і реальним значенням, тим вище точність прогнозування. Чим ближче R^2 до 1, тим краще ступінь створення моделі.

Для оцінки результату роботи різних систем, було проведено ряд експериментів з мережами MLP, CNN, RNN, LSTM CNN-LSTM(технічний аналіз) і CNN-LSTM(на основі соціальних мереж), на рисунках 4.1 – 4.6 порівняні реальні значення акції Apple з прогнозованим. Отриманні результати роботи програми збережені у форматі JSON, та за допомогою plotly створено та відображення залежності між прогнозованими і реальними цінами на акції.

а) Багатошарові персептрони, або скорочено MLP, є класичним типом нейронної мережі. Вони складаються з одного або кількох шарів нейронів. MLP підходять для завдань прогнозування класифікації, коли входам присвоюється клас чи мітка. Вони також підходять до завдань прогнозування регресії, коли реальне значення прогнозується з урахуванням набору вхідних даних.

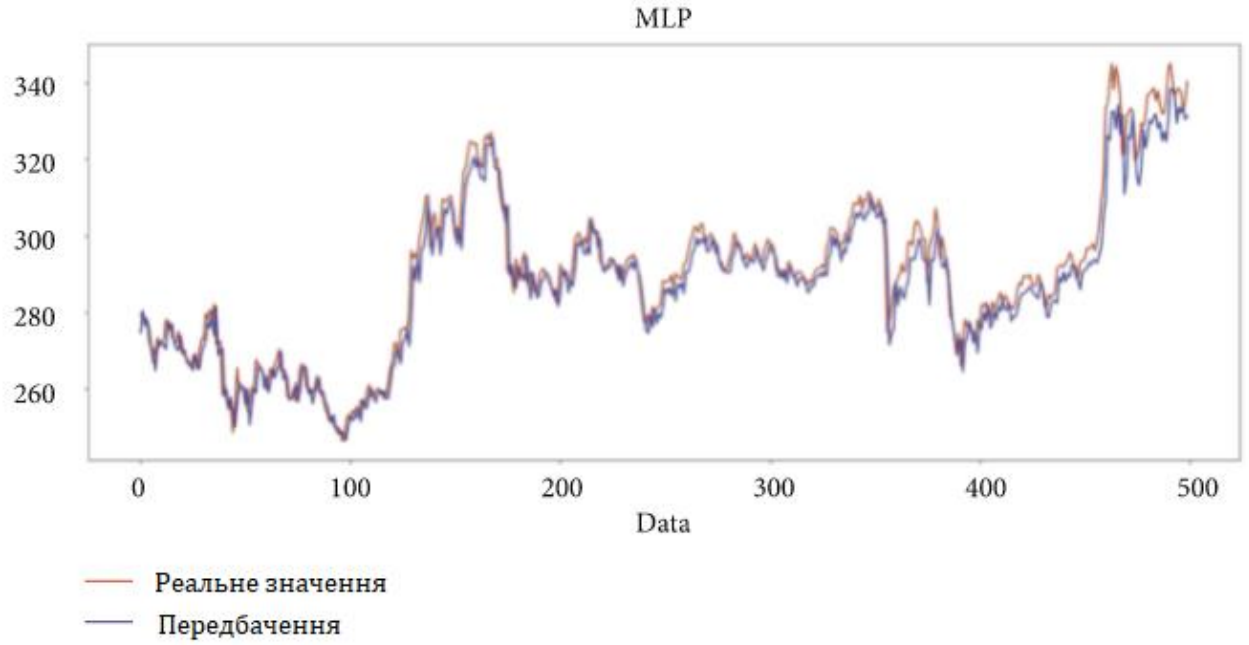


Рисунок 4.1 – Графік порівняння прогнозованого значення та реального значення для MLP

б) Згорткові нейронні мережі, або CNN, мережа, яка ефективна при розпізнавання та класифікації зображень. Вхід CNN традиційно є двовимірним, полем або матрицею, але також може бути змінений на одновимірний. Хоча CNN спеціально не розроблена для даних, що не належать до зображень, мережа досягає хороших результатів з таких проблем, як класифікація характеристик даних, що використовується при аналізі настроїв або прогнозуванні часових рядів. На рисунку 4.2 відображено графік прогнозування акцій, за допомогою мережі CNN.

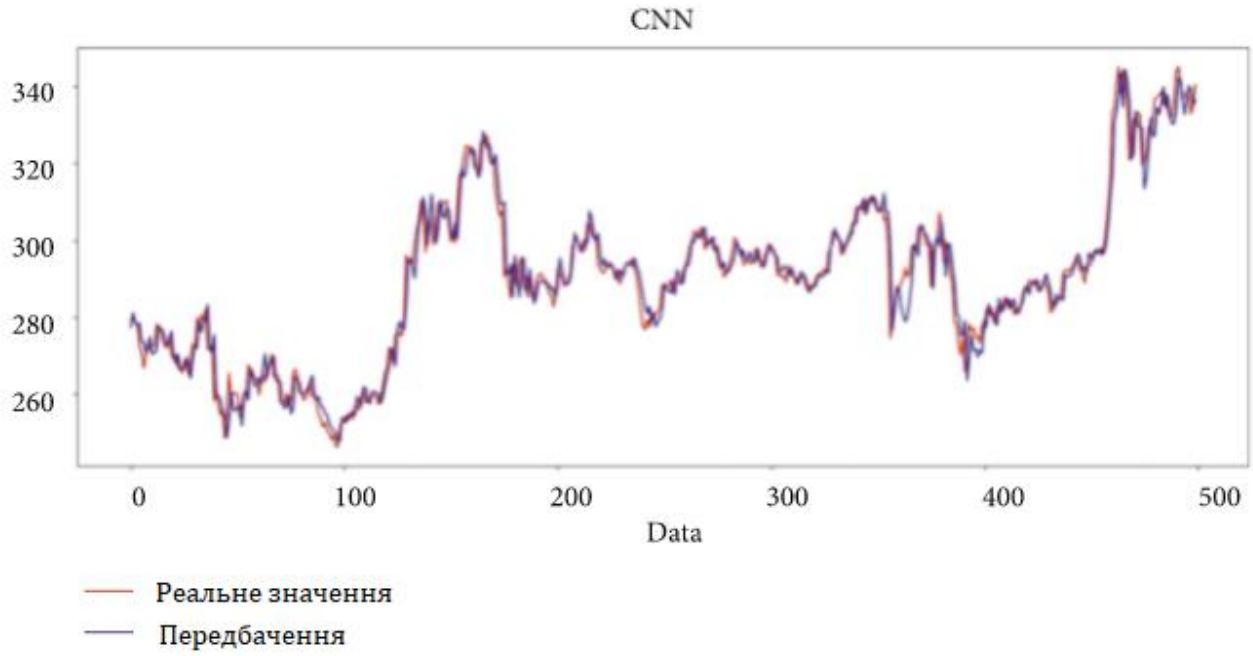


Рисунок 4.2 – Графік порівняння прогнозованого значення та реального значення для CNN

в) RNN - вид нейронних мереж, де зв'язки між елементами утворюють спрямовану послідовність. Рекурентні нейронні мережі або RNN були розроблені для роботи з проблемами прогнозування послідовності. На рисунку 4.3 відображено графік прогнозування акцій, за допомогою мережі RNN

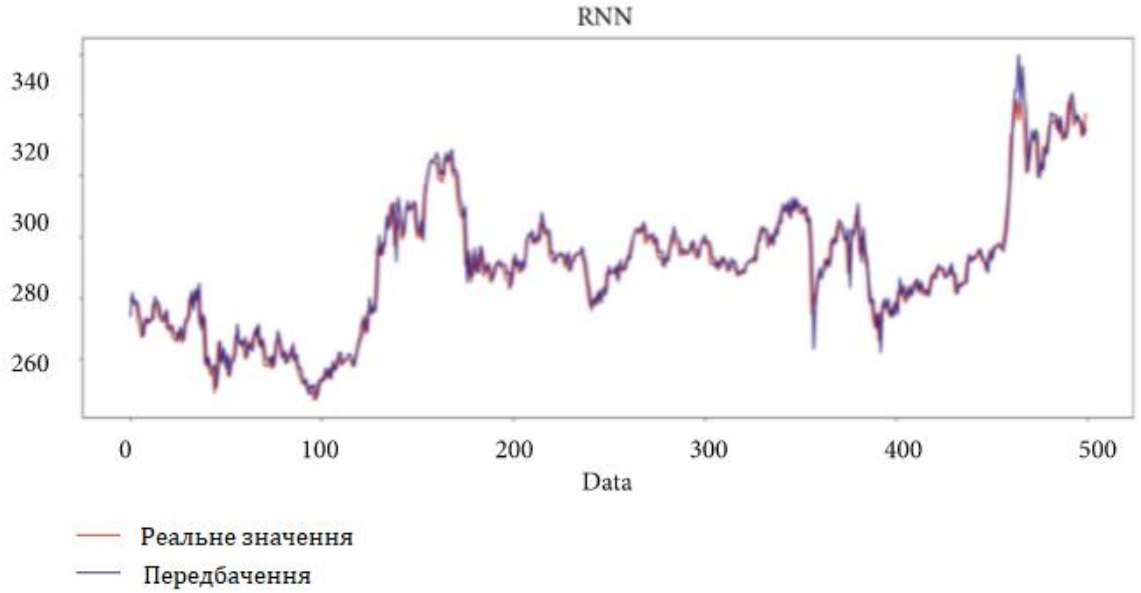


Рисунок 4.3 - Графік порівняння прогнозованого значення та реального значення для RNN

г) LSTM - тип рекурентної нейронної мережі, здатний вчитися довгостроковим залежностям. Вважається кращою версією RNN. На рисунку 4.4 відображено графік прогнозування акцій, за допомогою мережі RNN.

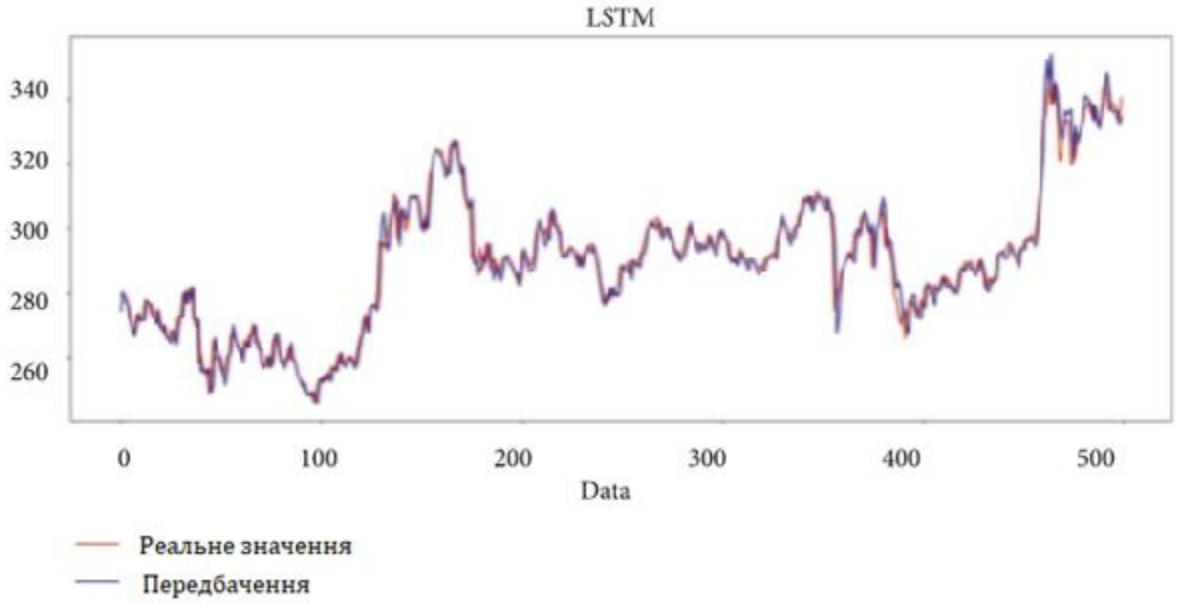


Рисунок 4.4 - Графік порівняння прогнозованого значення та реального значення для LSTM

г) CNN-LSTM – гібридний метод, який поєднує переваги CNN, в витягуванні характеристик з історичних даних для подальшого прогнозування на основі LSTM. Базується тільки на технічному аналізі, використовуючи лише історичні дані про акцію як джерело інформації. На рисунку 4.5 зображено графік порівняння прогнозованого значення та реального значення для CNN-LSTM.

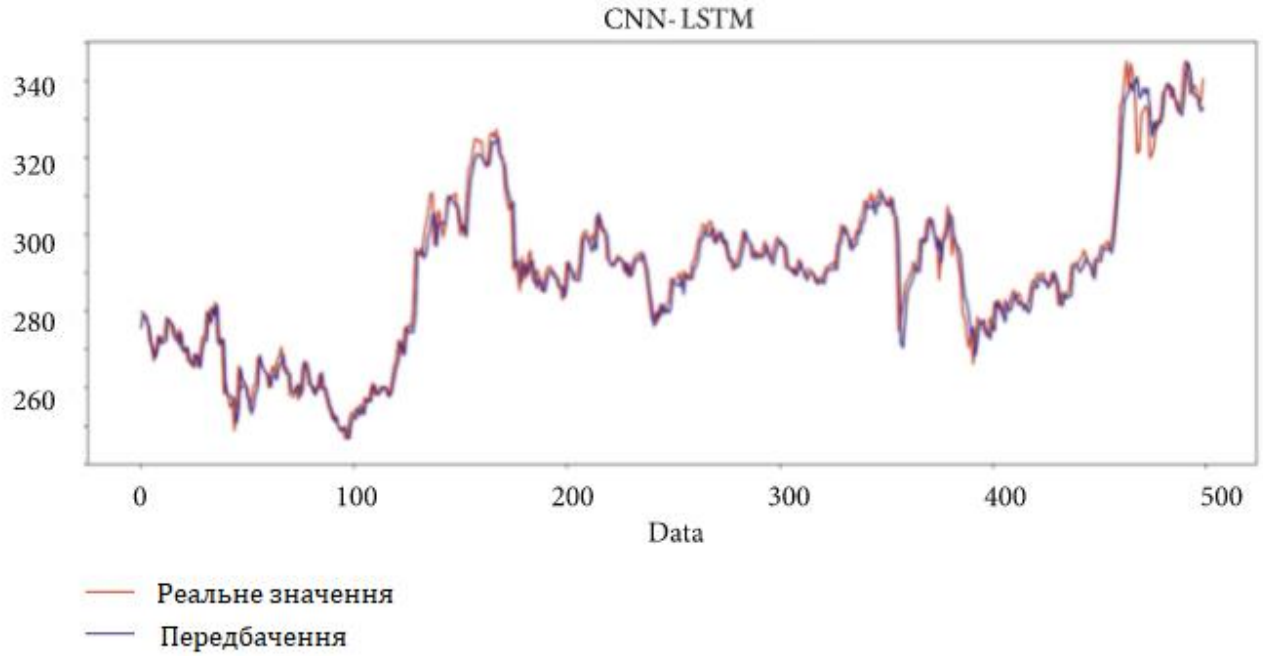


Рисунок 4.5 - Графік порівняння прогнозованого значення та реального значення для CNN-LSTM (тех аналіз)

д) CNN-LSTM (на основі соціальних мереж) – гібридний метод, який поєднує переваги CNN, в витягуванні характеристик з історичних даних, а також характеристик даних з соціальних мереж. Використовуючи отриманні дані агрегує їх в часовий ряд зі збереженням кореляційних індексів, для подальшого прогнозування на основі LSTM. LSTM отримує на вхід готовий ефективний часовий ряд, та створює прогноз на його онові. Даний метод базується на фундаментальному аналізі, використовуючи за основу дані з соціальних мереж. На рисунку 4.6 зображено графік порівняння прогнозованого значення та реального значення для CNN-LSTM .

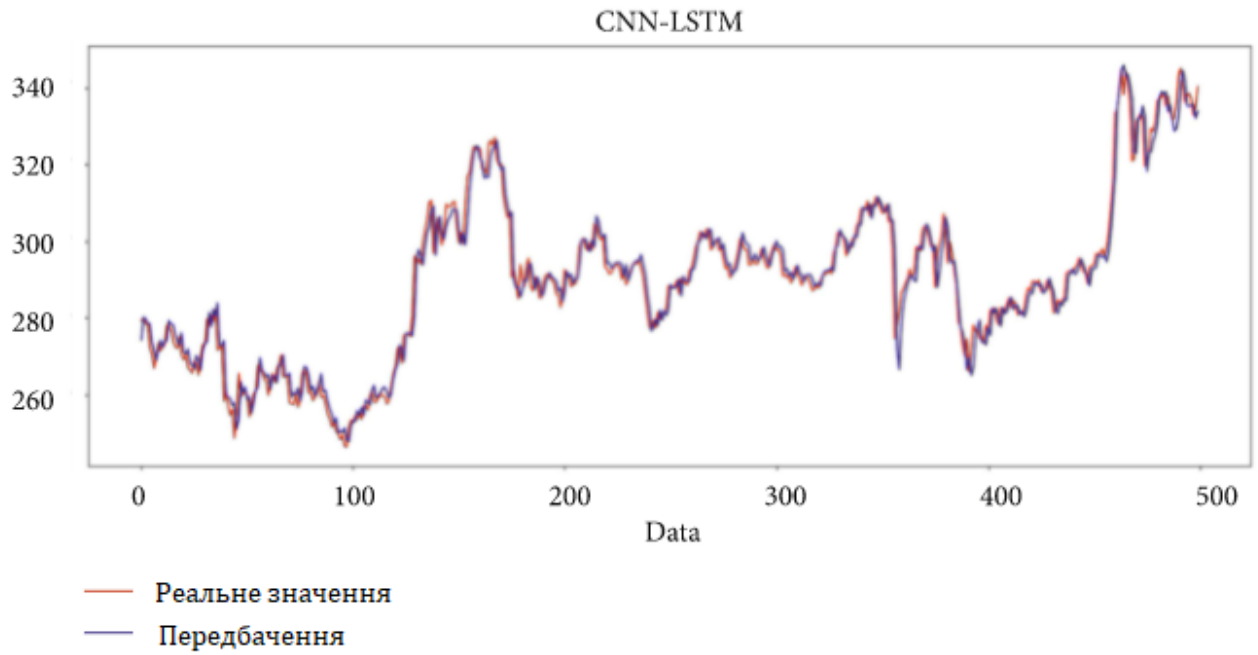


Рисунок 4.6 - Графік порівняння прогнозованого значення та реального значення для CNN-LSTM (на основі соціальних мереж)

Після того, як було створено графік порівняння для кожного з методів, потрібно зробити оцінку результатів.

Методи оцінюються, критеріями ефективності, такими як: середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE) і R-квадрат (R^2). Результати порівняння шести методів наведено на малюнках 4.7– 4.9.

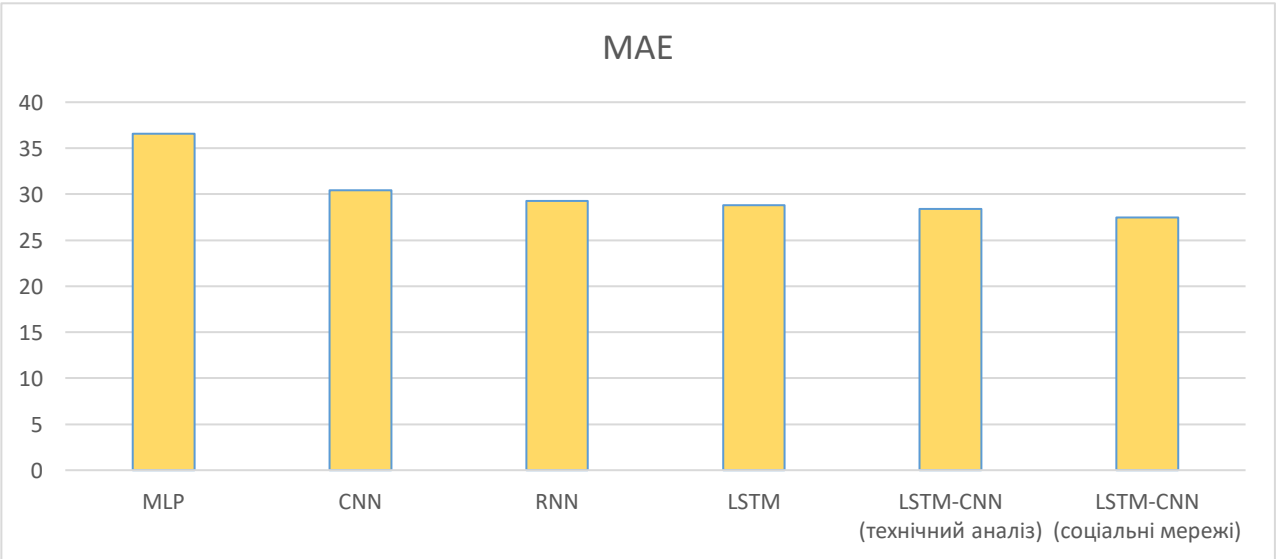


Рисунок 4.7 – Діаграма порівняння методів прогнозування за критерієм MAE

З рисунка 4.8, MAE для MLP є найбільшими, а для LSTM-CNN(на основі соціальних мереж) є найменшим. Це означає, що LSTM-CNN(на основі соціальних мереж) – є найбільш точною системою ля прогнозування. В порівнянні LSTM-CNN з LSTM-CNN(на основі соціальних мереж), можемо зробити висновок, що використання соціальних мереж, може покращити точніть прогнозування акцій.

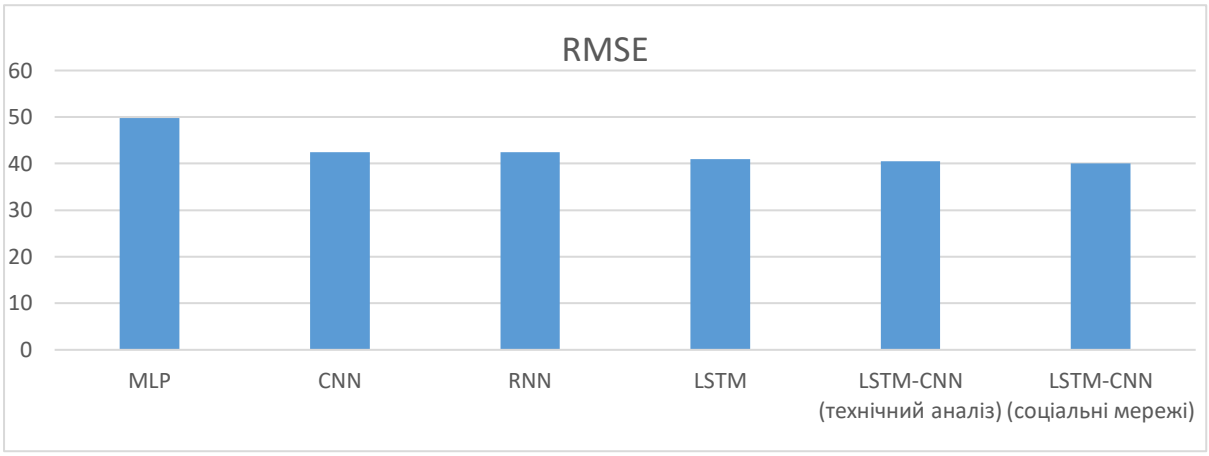


Рисунок 4.8 – Діаграма порівняння методів прогнозування за критерієм RMSE

З рисунка 4.8, RMSE для MLP є найбільшими, а для LSTM-CNN(на основі соціальних мереж) є найменшим. Метод середньоквадратичної помилки, також доводить, що LSTM-CNN в порівнянні з LSTM-CNN(на основі соціальних мереж), є менш точною , а отже, ми підтверджуємо, що використання соціальних мереж, може покращити точність прогнозування акцій.

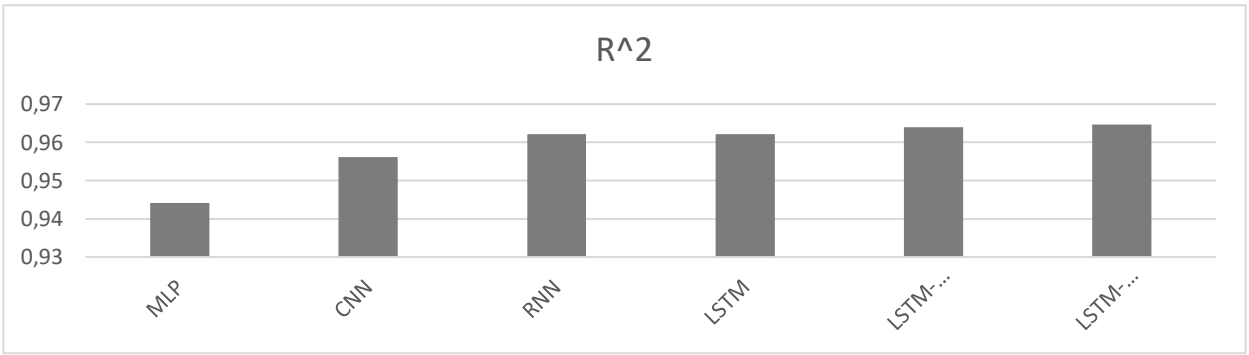


Рисунок 4.9 – Діаграма порівняння методів прогнозування за критерієм R-квадрат

З рисунка 4.9, R-квадрат для MLP є найменшим, а для LSTM-CNN(на основі соціальних мереж) є найбільшим. Метод середньоквадратичної помилки, також доводить, що LSTM-CNN в порівнянні з LSTM-CNN(на основі соціальних мереж), є менш ефективною, а отже, ми підтверджуємо, що використання соціальних мереж, як джерела даних, може покращити ефективність прогнозування акцій.

Результати оцінки ефективності CNN-LSTM(соціальні мережі) порівняно з MLP, CNN, RNN, LSTM і CNN-LSTM(без мереж), наведено в таблиці 4.2.

Таблиця 4.2 – Результати експериментів різних методів

	MLP	CNN	RNN	LSTM	CNN-LSTM	CNN-LSTM соц. мережі
MAE	36.583	30.423	29.281	28.781	28.385	27.473
RMSE	49.729	42.429	42.419	41.003	40.529	40.102
R^2	0.9442	0.9562	0.9622	0.9621	0.9640	0.9646

Відповідно до прогнозованих значень та реальних значень, кожного методу, можна розрахувати індекс оцінки. З отриманих результатів ми можемо побачити, що з точки зору точності прогнозування, критерій MAE для CNN-LSTM на основі соціальних мереж становить 27,564, а критерій RMSE - 39,688, що є найбільш ефективним показником серед шести моделей прогнозування, а з точки зору ефективності прогнозування, критерій R2 для CNN-LSTM на основі соціальних мереж становить 0,9646, що також є найбільш ефективним показником серед всіх досліджуваних моделей. Це показує, що ефективність прогнозування LSTM можна ефективно покращити, витягуючи характеристики даних акції та поєднуючи їх з даними з соціальних мереж через CNN.

Модель може добре передбачити ціну акцій та надати орієнтир для інвестицій інвесторів

Висновки до розділу

Було зроблено оцінку методів прогнозування ціни акції на фондовому ринку. Методи були оцінені критеріями ефективності, такими як: середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE) і R-квадрат (R^2). Порівнявши критерії оцінки CNN-LSTM на основі соціальних мереж, з багатошаровим персептроном (MLP), CNN, RNN, LSTM і CNN-LSTM, доведено, що CNN-LSTM на основі даних з соціальних мереж, має високу точність прогнозування і використовуючи дані з соціальних мереж, підвищує ефективність системи

5 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП ПРОЄКТУ

5.1 Опис ідеї проекту

Аналіз динаміки часових рядів та прогнозування їх еволюції мають велике значення для управління різними процесами у соціальних (наприклад, виборчі кампанії), економічних (фондові, ф'ючерсні та сировинні ринки) та фундаментальних системах (наприклад, соціальні мережі).

Слід зазначити, що складність процесів, що спостерігаються, істотно зростає при переході від технічних до фундаментальних і далі – до соціальних систем. Багато технічних систем за непередбачуваністю спостережуваних процесів поступаються системам, у яких однією з елементів є людина. Якщо для прогнозування динаміки процесів у технічних системах основною проблемою є наявність суттєвої стохастичності, то у соціальних, економічних та фундаментальних системах з'являється людський фактор.

Одним із найважливіших прикладів та об'єктів фундаментальних систем є соціальні мережі. Важливість вивчення нестаціонарних часових рядів, що описують процеси в фундаментальних системах, полягає в тому, що останні кілька років спостерігається значне зростання впливу подій, що відбуваються в інтернеті, як на політичні, так і на соціальні процеси у світі.

Сайти мікроблогів, такі як Twitter (https://twitter.com/) і Sina Weibo (http://weibo.com/), дозволяють підписникам створювати, ділитися та коментувати повідомлення визначеної максимальної довжини (у як Twitter, так і Sina Weibo, максимум 140 символів). Формат дозволяє користувачам спонтанно та швидко відповідати на теми, часто формуючи думки. Методи відповіді включають пересилання, згадування та функції відповіді, які подібні до Twitter. Сайти також надають функції коментарів і пошти для спілкування користувачів.

Twitter, найбільший у світі мікроблог, що був широко вивчений, надає цінну інформацію про настрої користувачів. У Китаї найбільш репрезентативними веб-сайтами мікроблогів є Sina Weibo і Tencent Weibo, кожен з яких містить понад 0,4 мільярда зареєстрованих користувачів. Термін Weibo означає мікроблог. Подібно

до того, як повідомлення в Twitter називаються твітами, повідомлення в Weibo називаються weibos.

Нещодавно головний економіст Банку Англії Енді Холдейн сказав, що джерела даних, які дають уявлення про настрій і емоції, можуть допомогти зрозуміти поведінку прийняття рішень, яка рідко відповідає раціональним процесам, які передбачають у багатьох економічних моделях. Розвиток аналізу настроїв дозволив деяким дослідникам досліджувати способи використання вмісту Twitter для прогнозування тенденцій на фондових ринках. Дослідження показали, що його можна використовувати для покращення прогнозування індексу Доу-Джонса.

Що ще важливіше, розвиток інформатики дозволив більш детально досліджувати економічні механізми мікрорівня, дозволяючи аналізувати високочастотні часові ряди (з роздільною здатністю секунд або навіть мікросекунд) Такі розробки можуть допомогти розширити розуміння фондових ринків і підвищити ефективність розподілу капіталу за допомогою комп'ютерних торгових систем. За оцінками, близько 10–40% торгівлі припадає на високочастотну торгівлю, створену комп’ютером (Aldridge & Krawciw, 2017). Програми часто купують або продають на основі висновків про настрої щодо ринку, тому краще розуміння короткострокових настроїв інвесторів щодо фондових ринків є важливим напрямком дослідження, який може підвищити точність та ефективність торгівлі та зменшити ризики, пов’язані з неконтрольованою високошвидкісною торгівлею.

Людство завжди було зачаровано майбутнім і тим, що воно може принести нам. Це поняття походить від глибоко закладеної цікавості та уяви всередині кожного. На жаль, це зробило нас вразливими для багатьох шахраїв протягом всієї історії, які хотіли розповісти нам про наше майбутнє або спробувати вгадати, використовуючи всілякі теорії. З розвитком наукового мислення і методологій та з освітою, що робить раціональне мислення доступним кожному, передбачати майбутнє стало важче, тому що кожному, кому потрібно це зробити, знадобиться емпіричний доказ своїх очікувань щодо будь-яких явищ, які нас оточують.

Це завдання буде вирішено шляхом впровадження набору інструментів, якими може скористатися будь-яка сторона, зацікавлена в тому, щоб передбачити індекси фондового ринку за допомогою історичних даних, зібраних із соціальних мереж на цю тему, за допомогою набору ключових слів. Після цього ці дані будуть оброблені, а шумні частини та відсутні значення будуть оброблені. Це призведе до створення набору даних, який може бути випущений і використаний як продукт сам по собі, але робота на цьому не закінчується, оскільки ці дані повинні давати деякі попередні ознаки впливу на поведінку ринку, і тут буде проведений аналіз кореляції та причинно-наслідкового зв’язку. Кореляційний аналіз вивчає вплив однієї змінної на іншу або зв'язок між змінами однієї змінної та іншої, і це може бути повністю позитивною кореляцією, що означає, що зміна одиниці в першій змінній призведе до зміни одиниці в іншій. змінні в одному напрямку (збільшення або зменшення), і навпаки. Аналіз причинності вивчає, як змінна може змінюватися залежно від змін іншої змінної, але з часовим лагом.

Якщо деякі змінні показали, що вони можуть впливати, це означає, що ми можемо перейти до наступного етапу застосування алгоритмів машинного навчання до цих змінних, щоб створити модель з цього набору даних. Це включає кілька моделей, які можна перевірити, як-от регресію, нейронні мережі та ARIMA, який є алгоритмом прогнозування часових рядів на основі регресії. Останнім і одним із найважливіших кроків у цьому процесі є чітке відображення результатів із гарною чіткою візуалізацією. Це може допомогти зрозуміти проблеми в моделях, створених з цими даними, і як їх виправити або продемонструвати успіх, якщо це сталося.

Таким чином, враховуючи фактори наведені вище, в рамках магістерської дисертації пропонується розглянути систему прогнозування ринку акцій на основі соціальних мереж в якості стартап-проекту, метою якої є зробити прогнозування вартості акцій фондових ринків за допомогою соціальних мереж легшим та більш передбачуваним.

Зміст ідеї, напрямки її застосування та основні вигоди наведено в табл. 5.1.

Таблиця 5.1 - Опис ідеї

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Систему прогнозування ринку акцій на основі соціальних мереж	Ознайомлення з методами передбачення руху цін	Можливість набути базових навичок передбачення руху цін та побудови простих торгових стратегій.
	Отримання прогнозованого значення акції на вказаний проміжок часу базуючись на вказаному періоді навчання	Дає змогу дізнатися майбутню ціну акції, для вказаного проміжку часу
	Отримання даних для прогнозування майбутньої ціни акції за вказаними ключовими словами та критеріями	Задавати свої унікальні пошукові запити
	Фільтрування даних для прогнозування майбутньої ціни акції за вказаними словами виключення	Задавати фільтр для навчання моделі прогнозування

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) наведено у табл. 5.2.

Аналогами є:

— Investing.com;

— Autochartist.com;

— Tradingview.com.

Таблиця 5.2 - Визначення сильних, слабких характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Мій проект	Конкуренти			W	N	S
			1	2	3			
1.	Прогноз напряму руху цін	+	+	+	-		+	
2.	Простота	+	+	-	-	+		
3.	Вибір діапазону дат для прогнозування	+	-	-	-	+		
4.	Зміна параметрів індикаторів	+	-	+	+		+	
5.	Пропозиції відкриття позицій	+	-	+	+		+	

Перелік сильних, слабких та нейтральних характеристик дозволяє виявити фактори, які впливають на конкурентоспроможність продукту.

5.2 Технологічний аудит ідеї проекту

Таблиця 5.3 - Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Система прогнозування майбутньї ціни акції	Мова програмування Python Бібліотека tensorflow Реляційна база даних PostgresSQL Нейронні мережі CNN-LSTM Індикатори фундаментального аналізу	Наявні	Доступні

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Для ефективного планування напрямів розвитку стартапу необхідно визначити ринкові можливості проекту та загрози, які можуть зробити проект неконкурентноздатним. Також необхідно врахувати стан ринку, попит потенційних клієнтів та пропозиції конкурентів.

Таблиця 5.4 - Характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Більше 10
2	Динаміка ринку (якісна оцінка)	Спадає
3	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
4	Специфічні вимоги до стандартизації та сертифікації	Можливі в окремих країнах світу
5	Середня норма рентабельності в галузі (або по ринку), %	Невідома

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Оцінка ринку без знань про нього	Інвестори без знань	Через доступність продукту та простоту у використанні	Продукт повинен бути простим у використанні

Таблиця 5.6 - Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Складність у користуванні системою; незручний інтерфейс	Незадоволення процесом роботи з системою	Підтримка користу-вачів, розробка нового інтерфейсу з врахуванням по-бажань користувачів
2	Недостатня кількість індикаторів фундаментального аналізу	Відмова від користування системою через незадоволення потреб	Впровадження в систему більше індикаторів фундаментального аналізу
3	Відсутність фінансування	Неможливість підтримки застосунку	Введення певного способу оплати

Таблиця 5.7 - Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Можливість зацікави-ти нових користувачів біржовою торгівлею	Розширення аудиторії	Збільшення прибут-ку та довіри до компанії
2	Підвищення попиту систем прогнозу фондового ринку	Наявність широкої цільової аудиторії	Залучення інвестицій через збільшення привабливості галузі
3	Співпраця з біржами	Збільшення притоку ресурсів та довіри	Співпраця з провід-ними компаніями

Таблиця 5.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Конкуренція - чиста	Кількість конкурентів	Підвищення якості
2. Рівень конкурентної боротьби - світовий	Робота в Інтернеті	Техідтримка користувачів різними мовами
3. За галузевою ознакою - галузева	Розрахована на трейдерів	Впровадження лише необхідного функціоналу
4. Товарно-родова конкуренція	Схожий функціонал у конкурентів	Реалізація унікального функціоналу
5. Нецінова	Простий користувацький інтерфейс	Створення простого та зручного інтерфейсу
6. За інтенсивністю - немарочна	Акцент на простоті та доступності	Не передбачено

Таблиця 5.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти в галузі	Постачальники	Клієнти	Товари-замінники
	Можливі	Можливі	Відсутні	Диктують	Наявні
Висновки:	Інтенсивна конкурентна боротьба	Є можливості для входу на ринок	Відсутні	Прислухатися до побажань клієнтів	Обмежень для виходу нема

Таблиця 5.10 - Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Прогноз напряму руху цін	Отримання конкретно прогнозу напряму руху цін
2.	Використання соціальних мереж як джерела інформації	Можливість швидше реагувати на зміни прогнозів
3.	Зміна параметрів індикаторів	Можливість підлаштувати прогнозування під свої потреби (часові проміжки, тощо)

Продовження таблиці 5.10

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
4.	Пропозиції відкриття позицій	Отримання конкретних рекомендацій щодо відкриття позицій
5.	Сповіщення про варіант відкриття позиції	Отримання сповіщення в будь-який час на бажаній платформі
6.	Простота у використанні	Швидке освоєння системи та комфортна робота з нею.
7.	Здатність до розширюваня	Можливість використання нових джерел даних та методів прогнозування

Таблиця 5.11 - Порівняльний аналіз сильних та слабких сторін

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з						
			-3	-2	-1	0	1	2	3
1.	Прогноз напряму руху цін	15				+			
2.	Використання соціальних мереж як джерела інформації	20	+						

Продовження таблиці 5.11

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з						
			-3	-2	-1	0	1	2	3
4.	Пропозиції відкриття позицій	14				+			
5.	Сповіщення про варіант відкриття позиції	15				+			
6.	Простота у використанні	20	+						
7.	Здатність до розширюваня	16			+				

Таблиця 5.12 - SWOT- аналіз стартап-проекту

Сильні сторони: 1) Соціальні мережі як джерело даних 2) Простота використання 3) Розширюваність	Слабкі сторони: 1) Мала кількість індикаторів
Можливості: 1) Зайняття ніші «для початківців» 2) Зацікавлення багатьох людей 3) Співпраця з великими компаніями	Загрози: 1) Перехід до конкурентів клієнтів після набуття досвіду 2) Недостатня зручність у використанні продукту

Таблиця 5.13 - Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Безкоштовний, дохід з реклами	30	2019р
2.	Щомісячна оплата	100	2019р

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 - Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Досвідчені трейдери	Низька	Низький	Низька	Складний вхід
2.	Трейдери-початківці	Достатня	Достатній	Середня	Легко увійти і конкурентам
Які цільові групи обрано: трейдери-початківці.					

Таблиця 5.15 - Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентос-проможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Безкоштовний	Концентрований маркетинг	Простота інтерфейсу	Диференціація
2	Введення певного способу оплати	Концентрований маркетинг	Відсутність реклами	Диференціація

Таблиця 5.16 - Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Шукати нових і забирати існуючих	Частково	Заняття конкурентної ніші

Таблиця 5.17 - Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сфо-рмувати комплексну позицію власного проекту (три ключових)
1	Простота у використанні	Стратегія диференціації	Простота у використанні, API в режимі реального часу	Простота Доступність Швидкодія

5.5 Розроблення маркетингової програми стартап-проекту

Таблиця 5.18 - Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Простий у використан-ні продукт для допомоги трейдерам-початківцям	Простота, API в режимі ре-ального часу	Простота, API в режимі реального часу, концентрація на конкретній групі користувачів

Таблиця 5.19 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Цільові клієнти – освоюють нову сферу	Інтернет	Простота для новачків	Залучити більше клієнтів	Простота Доступність Швидкодія

Висновок до розділу

Проект націлений на доволі складний та консервативний ринок з довгою історією та великою кількістю потенційних конкурентів. Однак, проект використовує у прогнозуванні достатньо новий для ринка підхід, використовуючи дані із соціальних мереж. Тому такий продукт підійде інвесторам, що намагаються поглянути на ринок без досвіду, а також на прихильників інновацій.

Для задоволення потреб своєї цільової аудиторії проект пропонує простий функціонал, прогнозування акції, враховуючи настрої та емоційне забарвлення користувачів соціальних мереж відносно тих чи інших компаній.

Даний функціонал забезпечить підвищення інтересу до ринку торгівлі акціями та систем прогнозування руху цін в цілому.

Крім цього, у перспективі проект може почати співробітництво з великими компаніями-лідерами фондового ринку в цілому, або принаймні приватними біржами.

ВИСНОВКИ

Дана магістерська дисертація спрямована на вирішення задачі прогнозування фондового ринку, шляхом створення системи для прогнозу ринку акцій на основі даних з соціальних мереж.

Під час виконання роботи було розглянуто актуальність та новизну даної системи. Було визначено дослідницькі питання, які необідні для подальшої розробки. Був проведений аналіз існуючих досліджень, використовуючи методологію systematic mapping study, завдяки чому було надано відповідь на дослідницькі питання. Проаналізовано головні ідеї існуючих досліджень і на їх основі був вибраний метод прогнозування.

Базуючись на існуючих дослідженнях, було встановлено, що найбільш ефективним методом прогнозування ринку акцій є гібридний метод CNN-LSTM. CNN-LSTM – складається зі згорткової мережі, яка агрегує характеристики даних соціальних мереж з характеристиками акцій зі збереженням кореляційних властивостей та мережі LSTM, яка є найкращою у прогнозуванні часових рядів.

Було виявлено, що основний підхід для прогнозування фондового ринку є фундаментальний аналіз, а тому було виявлено, що метод можна вдосконалити, використовуючи дані з соціальних мереж .

Проведено дослідження предметної області, визначено вимоги і завдання, які повинна вирішувати система, наведено перелік модулів з яких повинна складатися система. Виходячи із заданих функціональних вимог, проведено аналіз архітектури та технологій розробки необхідних для даної системи. Систему реалізовано у вигляді веб додатка реалізованого на основі мікросервісної архітектури.

На основі функціональних вимог, було виявлено, що процес прогнозування буде включати в себе 4 модуля :

- отримання даних;
- обробки даних;
- кореляції та аналізу причинності№

— прогнозування.

Після створення системи для прогнозування ринку акцій, було проведено ряд експериментів, для підтвердження ефективності та точності метода CNN-LSTM на основі соціальних мереж. Для перевірки було використано ідентичний набір даних, для шістьох різних методів прогнозування, використовуючи три критерії оцінки:

— середня абсолютна помилка (MAE);

— середньоквадратична помилка (RMSE);

— R-квадрат (R^2).

Виявлено, що метод CNN-LSTM на основі соціальних мереж показав найкращі результати, та виявився найбільш ефективним та найбільш точним з усіх запропонованих методів. Було доведено, що використання соціальних мереж збільшує ефективність та точність прогнозу курсу акції фондового ринку

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Krantz M. Fundamental analysis for dummies [Текст] / M.Krantz –Hoboken: Wiley Publishing Inc., 2009. – 387 p.

2) Armstrong J.S. Illusions in Regression Analysis [Текст] / J.S. Armstrong – Pennsylvania: Penn Press, 2011. – 147 p.

3) Haykin S.S. Neural networks [Текст] / S.S. Hayking – Hamilton: Pearson Education, 2009. – 938 p.

4) Deng L. Neural learning: Methods for prediction / Deng L. and Yu D. //

5) Foundations and Trends in Trading Processing, 7(4–5) – 2015. – pp. 217–382.

6) Adhikari R. An Introductory Study on Time Series Modeling and Forecasting [Текст] / Adhikari R. – Riga: LAP Lambert Academic Publishing, 2013. – 76 p.

7) Бідюк П. І. Аналіз часових рядів (навчальний посібник) / Бідюк П. І., Романенко В. Д., Тимощук О. Л. – К.: Політехніка, 2010. – 317 с.

8) RMSProp. [Електронний ресурс] / Tieleman T. and Hinton G // COURSERA: Neural Networks for Machine Learning. – 2012. – Режим доступу: https://www.coursera.org/learn/deep-neural-network/lecture/BhJlm/rmsprop.

9) Deng L. Deep learning: Methods and applications / Deng L. and Yu D. //

10) Foundations and Trends in Signal Processing, 7(3–4) – 2014. – pp. 197–387.

11) Zeiler M. D. Visualizing and understanding convolutional networks / Zeiler M.D. and Fergus R. // Proceedings of the European Conference on Computer Vision. – 2014. – pp. 818–833.

12) Szegedy C. Going deeper with convolutions / Szegedy C. and Liu W. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – pp. 1–9.

13) Casey M.J. The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order [Текст] / M.J. Casey – London: St. Martin's Press, 2015. – 368 p.

14) Antonopoulos A.M. Mastering Bitcoin: Unlocking Digital Crypto-Currencies [Текст] / A.M. Antonopoulos – London: O'Reilly Media, 2017. – 416 p.

15) Ecash [Електронний ресурс]. – Режим доступу https://en.bitcoinwiki.org/wiki/Ecash

16) Pre-Bitcoin Virtual Currencies That Bit the Dust [Електронний ресурс]. – Режим доступу https://www.coindesk.com/3-pre-bitcoin-virtual-currencies-bit-dust/.

17) Ian Goodfellow. Deep Learning / Ian Goodfellow, Yoshua Bengio and Aaron Courville. – Boston: The MIT Press, 2016. – 800 p.

18) Zhu Xiaojin. Supervised learning literature survey / Zhu Xiaojin. – Department of Computer Science and Engg, University of Wiskonsin-Maddison, 2005. – 1530 p.

19) NumPy [Електронний ресурс]. – Режим доступу: http://www.numpy.org/

20) Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования / В.В. Вьюгин. – М.: МЦНМО, 2013. – 387 с.

21) Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org

22) SciKit-Learn [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/stable/

23) Kerras [Електронний ресурс] – Режим доступу до ресурсу: https://keras.io/

24) Angular [Електронний ресурс] – Режим доступу до ресурсу: https://angular.io

25) Docker [Електронний ресурс] – Режим доступу до ресурсу: https://www.docker.com

26) Kubernetes [Електронний ресурс] – Режим доступу до ресурсу: https://kubernetes.com

27) Git [Електронний ресурс] – Режим доступу до ресурсу: https://www.atlassian.com/ru/git

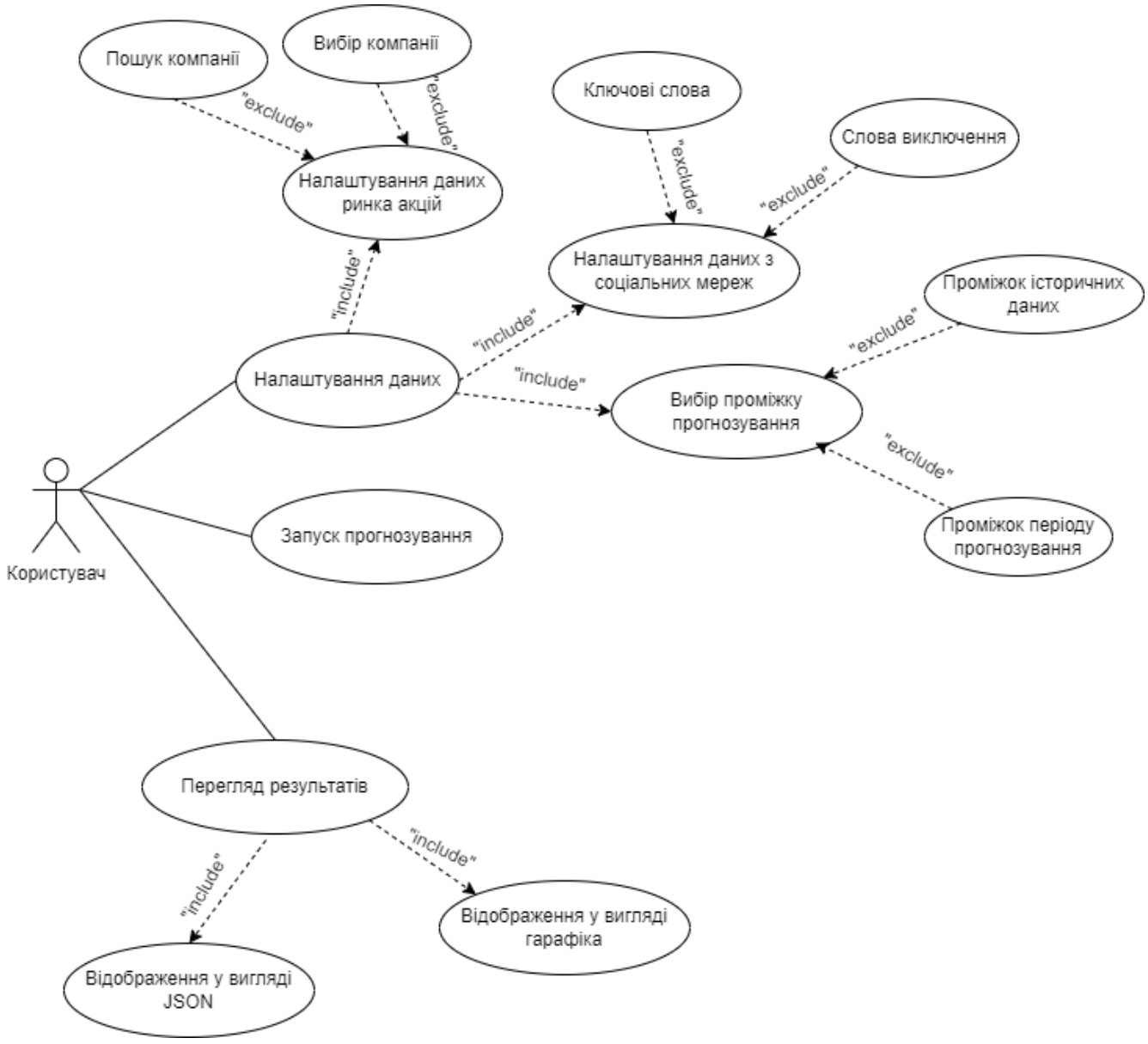
Додатки
на магістерську дисертацію

на тему: «Програмна система прогнозування ринку акцій на основі даних з соціальних мереж»

Київ – 2021

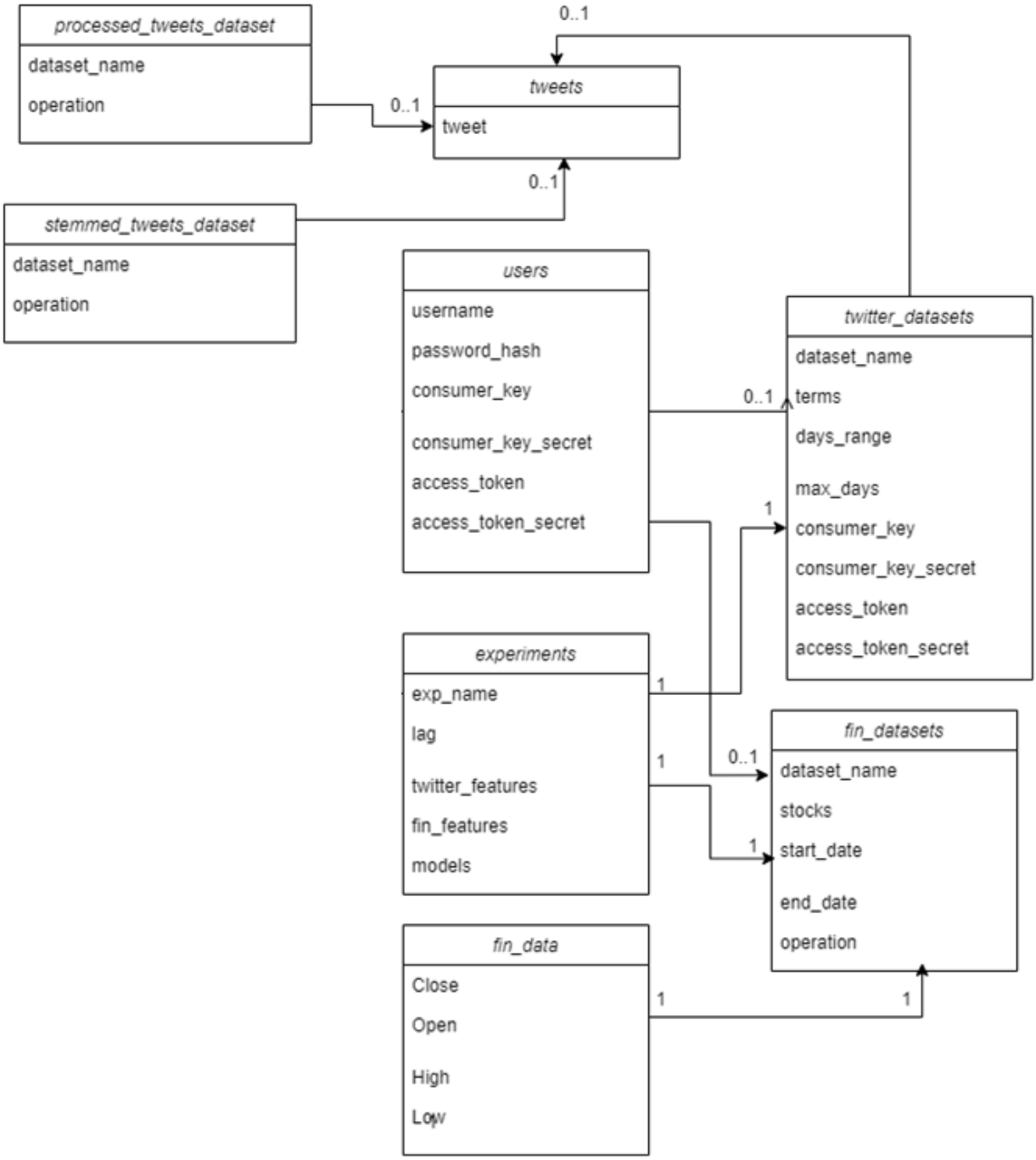
ДОДАТОК А

Діаграма прецедентів системи прогнозування курсу акцій



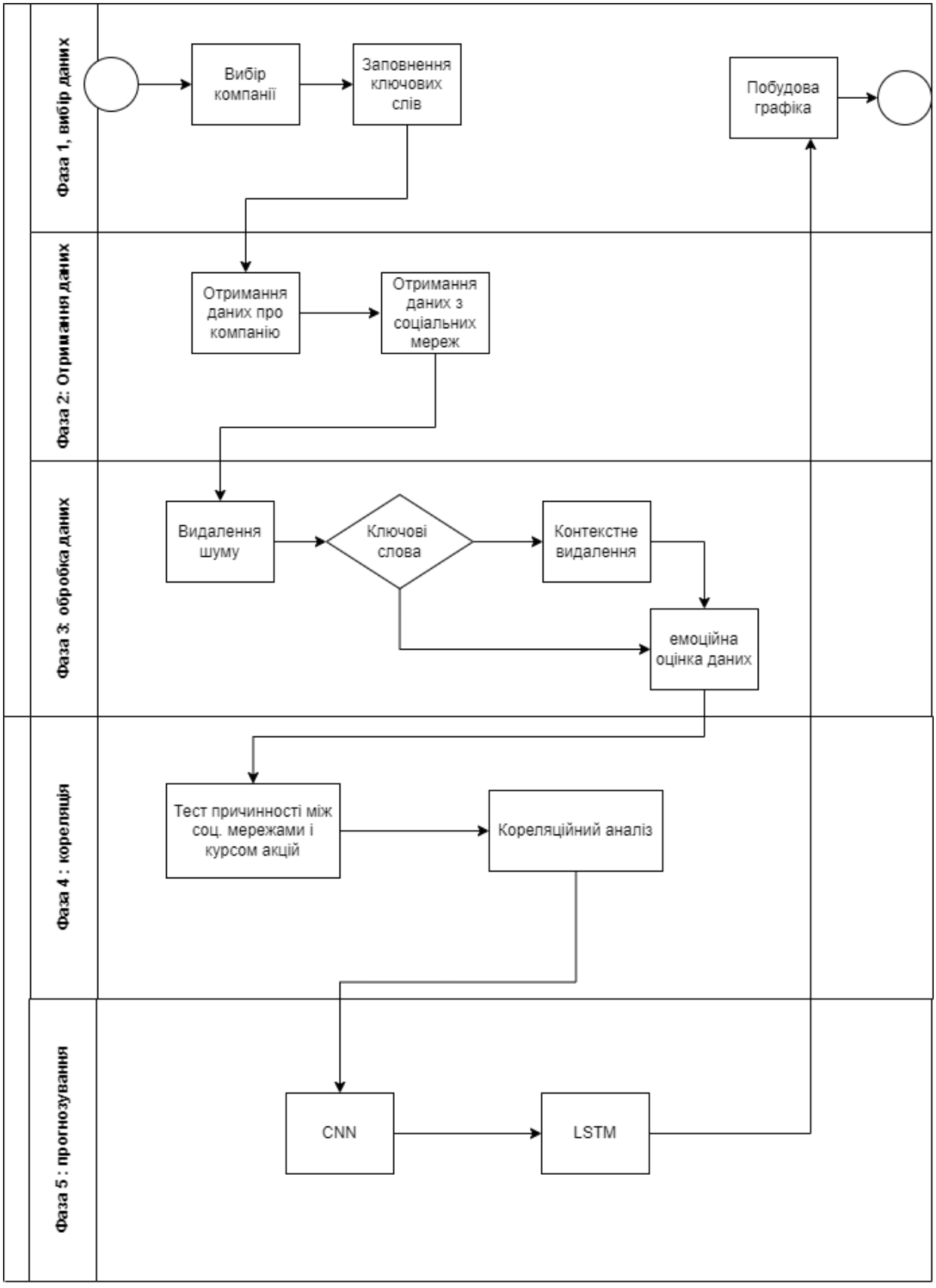
ДОДАТОК Б

Діаграма класів системи прогнозування курсу акцій



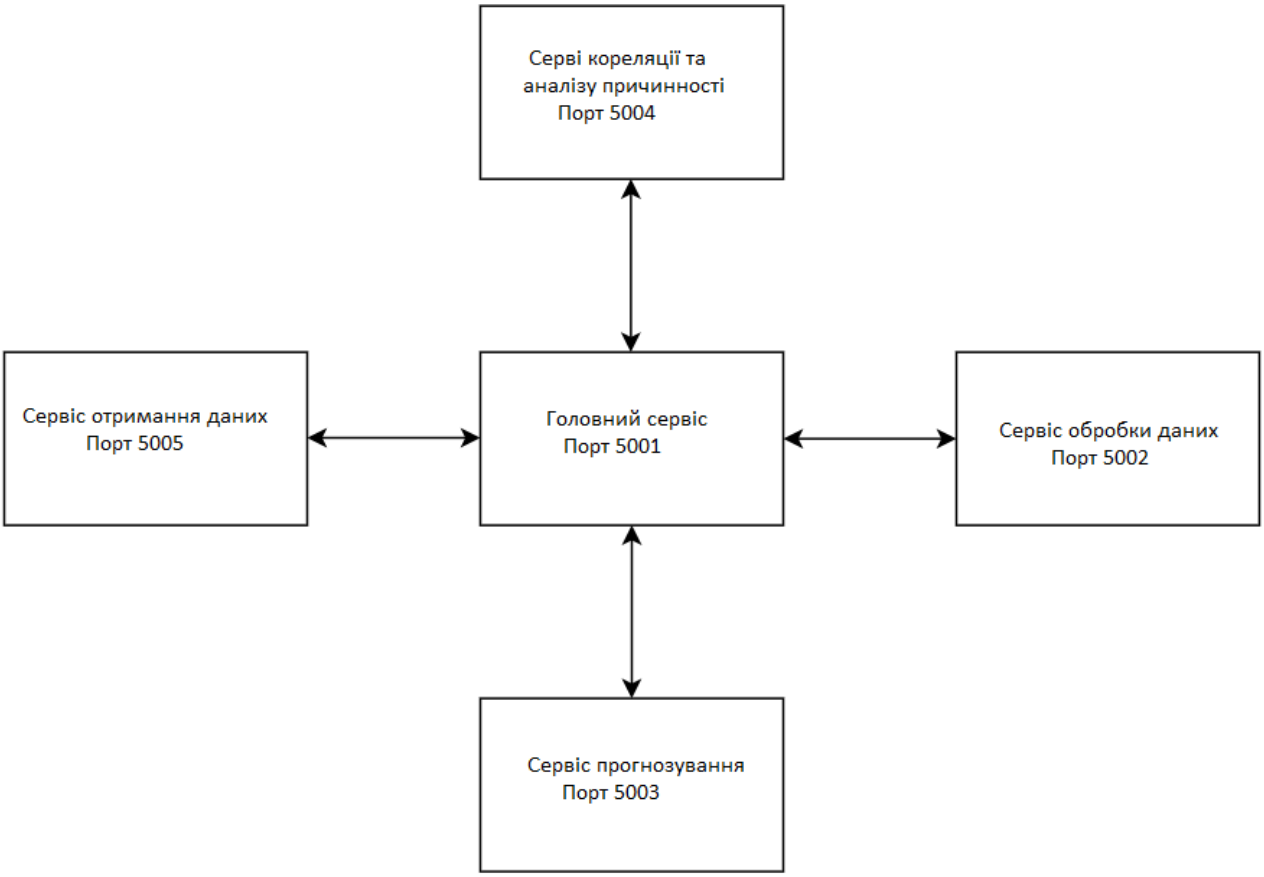
ДОДАТОК В

Діаграма процесу системи прогнозування курсу акцій



ДОДАТОК Г

Схема зв’язку між сервісами



ДОДАТОК Ґ

Об’єкт твітеру

Attribute	Type	Description
created_at	String	UTC time when this Tweet was created. Example: `"created_at": "Wed Oct 10 20:19:24 +0000 2018"`
id	Int64	The integer representation of the unique identifier for this Tweet. This number is greater than 53 bits and some programming languages may have difficulty/silent defects in interpreting it. Using a signed 64 bit integer for storing this identifier is safe. Use `id_str` to fetch the identifier to be safe. See Twitter IDs for more information. Example: `"id":1050118621198921728`
id_str	String	The string representation of the unique identifier for this Tweet. Implementations should use this rather than the large integer in `id`. Example: `"id_str":"1050118621198921728"`
text	String	The actual UTF-8 text of the status update. See twitter-text for details on what characters are currently considered valid. Example: `"text":"To make room for more expression, we will now count all emojis as equal—including those with gender and skin t… https://t.co/MkGjXf9aXm"`
source	String	Utility used to post the Tweet, as an HTML-formatted string. Tweets from the Twitter website have a source value of `web`. Example: `"source":"Twitter Web Client"`
truncated	Boolean	Indicates whether the value of the `text` parameter was truncated, for example, as a result of a retweet exceeding the original Tweet text length limit of 140 characters. Truncated text will end in ellipsis, like

		this … Since Twitter now rejects long Tweets vs truncating them, the large majority of Tweets will have this set to false . Note that while native retweets may have their toplevel text property shortened, the original text will be available under the retweeted_status object and the truncated parameter will be set to the value of the original status (in most cases, false). Example: "truncated":true
in_reply_to_status_id	Int64	*Nullable*. If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's ID. Example: "in_reply_to_status_id":1051222721923756032
in_reply_to_status_id_str	String	*Nullable*. If the represented Tweet is a reply, this field will contain the string representation of the original Tweet's ID. Example: "in_reply_to_status_id_str":"1051222721923756032"
in_reply_to_user_id	Int64	*Nullable*. If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's author ID. This will not necessarily always be the user directly mentioned in the Tweet. Example: "in_reply_to_user_id":6253282
in_reply_to_user_id_str	String	*Nullable*. If the represented Tweet is a reply, this field will contain the string representation of the original Tweet's author ID. This will not necessarily always be the user directly mentioned in the Tweet. Example: "in_reply_to_user_id_str":"6253282"
in_reply_to_screen_name	String	*Nullable*. If the represented Tweet is a reply, this field will contain the screen name of the original Tweet's author. Example: "in_reply_to_screen_name":"twitterapi"
user	User object	The user who posted this Tweet. See User data dictionary for complete list of attributes. Example highlighting select attributes: { "user": { "id": 6253282,

		"id_str": "6253282", "name": "Twitter API", "screen_name": "TwitterAPI", "location": "San Francisco, CA", "url": "https://developer.twitter.com", "description": "The Real Twitter API. Tweets about API changes, service issues and our Developer Platform. Don't get an answer? It's on my website.", "verified": true, "followers_count": 6129794, "friends_count": 12, "listed_count": 12899, "favourites_count": 31, "statuses_count": 3658, "created_at": "Wed May 23 06:01:13 +0000 2007", "utc_offset": null, "time_zone": null, "geo_enabled": false, "lang": "en", "contributors_enabled": false, "is_translator": false, "profile_background_color": "null", "profile_background_image_url": "null", "profile_background_image_url_https": "null", "profile_background_tile": null, "profile_link_color": "null", "profile_sidebar_border_color": "null", "profile_sidebar_fill_color": "null", "profile_text_color": "null", "profile_use_background_image": null, "profile_image_url": "null", "profile_image_url_https": "https://pbs.twimg.com/profile_images/942858479592554497/BbazLO9L_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/6253282/1497491515", "default_profile": false, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null } }
coordinates	Coordinates	*Nullable*. Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as geoJSON (longitude first, then latitude). Example: "coordinates": {

		"coordinates": [-75.14310264, 40.05701649], "type":"Point" }

place	Places	*Nullable* When present, indicates that the tweet is associated (but not necessarily originating from) a Place . Example:
		"place":
		{
		"attributes":{},
		"bounding_box":
		{
		"coordinates":
		[[
		[-77.119759,38.791645],
		[-76.909393,38.791645],
		[-76.909393,38.995548],
		[-77.119759,38.995548]
		]],
		"type":"Polygon"
		},
		"country":"United States",
		"country_code":"US",
		"full_name":"Washington, DC",
		"id":"01fbe706f872cb32",
		"name":"Washington",
		"place_type":"city",
		"url":"http://api.twitter.com/1/geo/id/0172cb32.json"
		}

quoted_status_id	Int64	This field only surfaces when the Tweet is a quote Tweet. This field contains the integer value Tweet ID of the quoted Tweet. Example: "quoted_status_id":1050119905717055488
quoted_status_id_str	String	This field only surfaces when the Tweet is a quote Tweet. This is the string representation Tweet ID of the quoted Tweet. Example: "quoted_status_id_str":"1050119905717055488"
is_quote_status	Boolean	Indicates whether this is a Quoted Tweet. Example: "is_quote_status":false

quoted_status	Tweet	This field only surfaces when the Tweet is a quote Tweet. This attribute contains the Tweet object of the original Tweet that was quoted.
retweeted_status	Tweet	Users can amplify the broadcast of Tweets authored by other users by retweeting . Retweets can be distinguished from typical Tweets by the existence of a retweeted_status attribute. This attribute contains a representation of the *original* Tweet that was retweeted. Note that retweets of retweets do not show representations of the intermediary retweet, but only the original Tweet. (Users can also unretweet a retweet they created by deleting their retweet.)
quote_count	Integer	*Nullable*. Indicates approximately how many times this Tweet has been quoted by Twitter users. Example: "quote_count":33 Note: This object is only available with the Premium and Enterprise tier products.
reply_count	Int	Number of times this Tweet has been replied to. Example: "reply_count":30 Note: This object is only available with the Premium and Enterprise tier products.
retweet_count	Int	Number of times this Tweet has been retweeted. Example: "retweet_count":160
favorite_count	Integer	*Nullable*. Indicates approximately how many times this Tweet has been liked by Twitter users. Example: "favorite_count":295
entities	Entities	Entities which have been parsed out of the text of the Tweet. Additionally see Entities in Twitter Objects . Example: "entities": { "hashtags":[], "urls":[], "user_mentions":[],

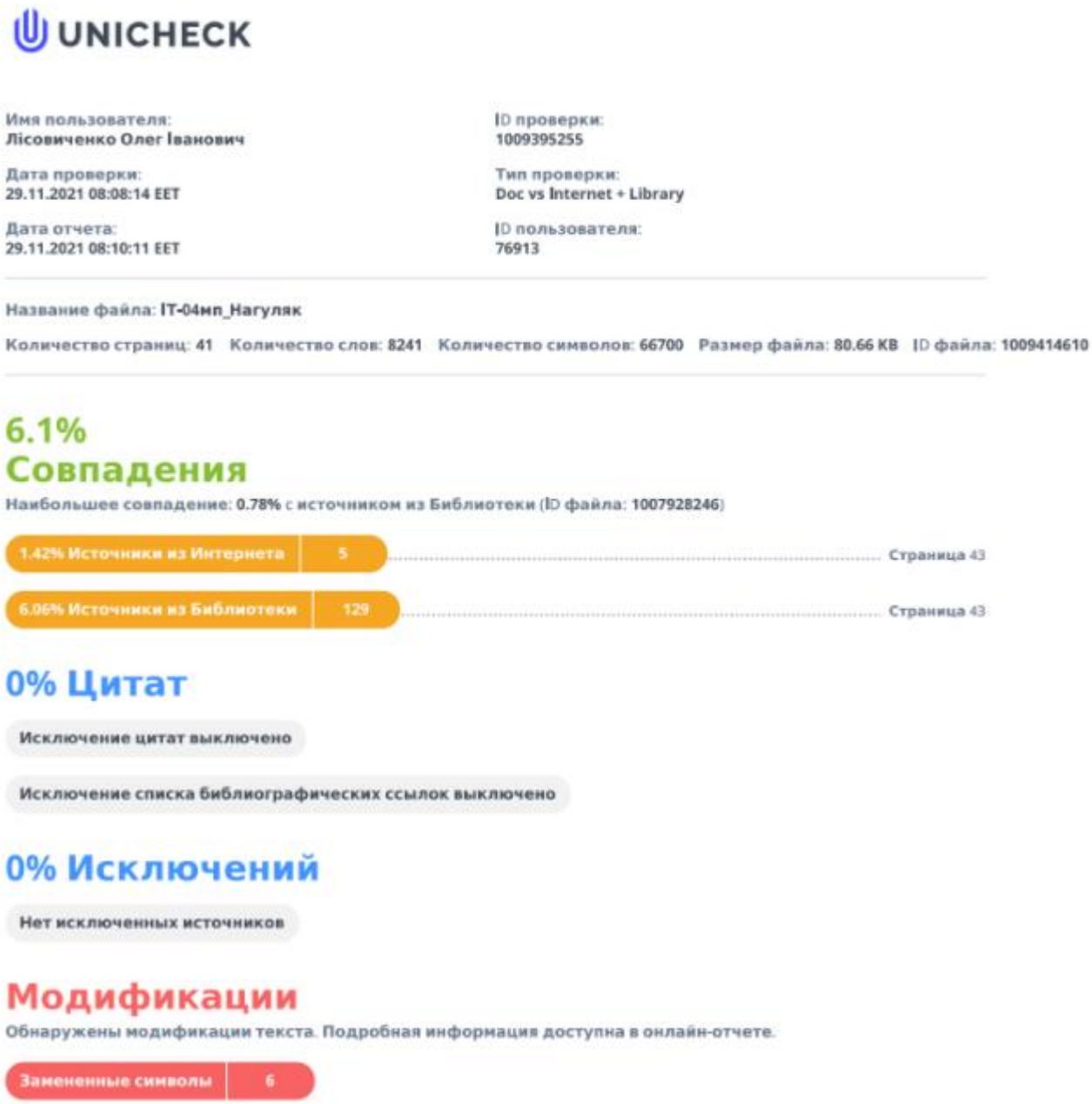
		"media":[], "symbols":[] "polls":[] }

extended_entities	Extended Entities	When between one and four native photos or one video or one animated GIF are in Tweet, contains an array 'media' metadata. This is also available in Quote Tweets. Additionally see Entities in Twitter Objects . Example:
		"entities":
		{
		"media":[]
		}

favorited	Boolean	*Nullable*. Indicates whether this Tweet has been liked by the authenticating user. Example: "favorited":true
retweeted	Boolean	Indicates whether this Tweet has been Retweeted by the authenticating user. Example: "retweeted":false
possibly_sensitive	Boolean	*Nullable*. This field only surfaces when a Tweet contains a link. The meaning of the field doesn't pertain to the Tweet content itself, but instead it is an indicator that the URL contained in the Tweet may contain content or media identified as sensitive content. Example: "possibly_sensitive":false
filter_level	String	Indicates the maximum value of the filter_level parameter which may be used and still stream this Tweet. So a value of medium will be streamed on none, low, and medium streams. Example: "filter_level": "low"
lang	String	*Nullable*. When present, indicates a BCP 47 language identifier corresponding to the machine-detected language of the Tweet text, or und if no language could be detected. See more documentation HERE. Example: "lang": "en"

ДОДАТОК Д

Результат перевірки на співпадіння



ДОДАТОК Е

Лістинг програмного коду

```
import numpy as np
import pandas as pd
import hvplot.pandas
# Use 70% of the data for training and the remainder for testing
split = int(0.70*len(X))
X_train = X[:split-1]
X_test = X[split:]
y_train = y[:split-1]
y_test = y[split:]
```

In [11]:

```
# Use MinMaxScaler to scale the data between 0 and 1.
# Importing MixMaxScaler from sklearn
from sklearn.preprocessing import MinMaxScaler

# Scale the features training and testing sets
# NOTE: It is good practice to fit only the training data to avoid look-ahead bias
scaler = MinMaxScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
scaler = MinMaxScaler().fit(y_train)
y_train = scaler.transform(y_train)
y_test = scaler.transform(y_test)
```

In [12]:

```
print("*** Feature Data***")
print (f" Scaled X sample values:\n{X_train[:3]} \n")
print("***Target Data***")
print (f"Scaled y sample values:\n{y_train[:3]}")
*** Feature Data***
 Scaled X sample values:
[[0.33333333]
 [0.10606061]
 [0.48484848]]

***Target Data***
Scaled y sample values:
[[0.68162134]
 [0.72761425]
 [0.60270722]]
```

In [13]:

```
print(f"Original X_train dimensions: {X_train.shape}")
Original X_train dimensions: (377, 1)
```

In [14]:

```
# Reshape the features for the model
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1],1))
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1],1))

print(f"Reshaped X_train dimensions: {X_train.shape}")
Reshaped X_train dimensions: (377, 1, 1)

# This function accepts the column number for the features (X) and the target (y)
# It chunks the data up with a rolling window of Xt-n to predict Xt
# It returns a numpy array of X any y
def window_data(df, window, feature_col_number, target_col_number):
    X = []
```

```
    y = []
    for i in range(len(df) - window - 1):
        features = df.iloc[i:(i + window), feature_col_number]
        target = df.iloc[(i + window), target_col_number]
        X.append(features)
        y.append(target)
    return np.array(X), np.array(y).reshape(-1, 1)
```

In [8]:

```
# Predict Closing Prices using a 10 day window of previous closing prices
# Try a window size anywhere from 1 to 10 and see how the model performance
changes
window_size = 1

# Column index 1 is the `Close` column
feature_column = 1
target_column = 1
X, y = window_data(df, window_size, feature_column, target_column)
```

In [9]:

```
# Use 70% of the data for training and the remaineder for testing
split = int(0.70*len(X))
X_train = X[:split-1]
X_test = X[split:]
y_train = y[:split-1]
y_test = y[split:]
```

In [10]:

```
# Use MinMaxScaler to scale the data between 0 and 1.
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

scaler = MinMaxScaler().fit(y_train)
y_train = scaler.transform(y_train)
y_test = scaler.transform(y_test)
```

In [11]:

```
print("*** Feature Data***")
print (f" Scaled X sample values:\n{X_train[:3]} \n")
print("***Target Data***")
print (f"Scaled y sample values:\n{y_train[:3]}")
*** Feature Data***
 Scaled X sample values:
[[0.7111066 ]
 [0.68162134]
 [0.72761425]]

***Target Data***
Scaled y sample values:
[[0.68162134]
 [0.72761425]
 [0.60270722]]
```

In [12]:

```
print(f"Original X_train dimensions: {X_train.shape}")
Original X_train dimensions: (377, 1)
```

In [13]:

```
# Reshape the features for the model
X_train = X_train.reshape((X_train.shape[0],X_train.shape[1],1))
```