НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

«На правах рукопису»
УДК 004.89: 004:336.748

До захисту допущено:
В.о. завідувачки кафедри
_____Ірина ДЖИГИРЕЙ
«___»_____2024 р.

Магістерська дисертація

на здобуття ступеня магістра

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

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Прогнозування криптовалют на основі історичних даних та даних соціальних медіа на основі глибокого навчання»

Виконав:
студент ІІ курсу, групи КІ-21мп
Олійник Богдан Олександрович _____

Науковий керівник:
проф. кафедри ШІ, д.т.н., професор,
Данилов Валерій Якович _____

Рецензент:
директор НН ФТІ, д.т.н., професор,
Новіков Олексій Миколайович _____

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

Київ – 2024

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В.о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«___»_____2023 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Олійнику Богдану Олександровичу

1. Тема дисертації «Прогнозування криптовалют на основі історичних даних та даних соціальних медіа на основі глибокого навчання», науковий керівник дисертації Данилов Валерій Якович, проф. кафедри ШІ, д.т.н., професор, затверджені наказом по університету від «08» листопада 2023 р. №5200-с

2. Термін подання студентом дисертації: 07 січня 2024 р.

3. Об'єкт дослідження: задача довгострокового прогнозування.

4. Вихідні дані: історичні дані про динаміку продажів товару в мережі роздрібної торгівлі в межах областей.

5. Перелік завдань, які потрібно розробити:

1) Здійснити огляд технічної літератури за темою роботи.
2) Дослідити актуальність обраної теми.
3) Ознайомитись з наявними методами довгострокового прогнозування.
4) Здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки.
5) Розробити та реалізувати програмний продукт, що використовує апарат ансамблювання нейронних мереж, та розв'язує задачу довгострокового прогнозування.

6) Провести експерименти, що засвідчують працездатність запропонованої моделі, виконати аналіз результатів.
7) Провести аналіз ринкових можливостей запуску стартап проєкту.
8) Розробити концептуальні висновки.
9) Підготувати ілюстративний матеріал.
10) Оформити пояснювальну записку.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: архітектура розробленого рішення, презентація.

7. Орієнтовний перелік публікацій:

Заплановано опублікувати такі результати як «Social Media Impact on the "Cosmos" Blockchain Ecosystem: State and Prospect» авторів Олійник Б.О, Петренко А. В. в журналі, який індексується у наукометричній базі Scopus.

8. Дата видачі завдання: 30 серпня 2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми МД	01.09.2023 - 07.09.2023	Виконано
2	Ознайомлення зі структурою МД згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	08.09.2023 - 14.09.2023	Виконано
3	Ознайомлення з ДСТУ 3008-2015 та стандартами ЄСПД	15.09.2023 - 21.09.2023	Виконано
4	Проведення дослідження за темою МД під керівництвом керівника	22.09.2023 - 28.09.2023	Виконано
5	Завершення роботи над першим варіантом частини МД	29.09.2023 - 05.10.2023	Виконано
6	Проведення роботи над експериментальною частиною МД	06.10.2023 - 12.10.2023	Виконано
7	Проведення роботи над програмним продуктом	13.10.2023 - 03.12.2023	Виконано
8	Оформлення МД та аналіз отриманих результатів	03.12.2023 - 07.01.2023	Виконано

Студент Богдан ОЛІЙНИК

Науковий керівник Валерій ДАНИЛОВ

РЕФЕРАТ

Магістерська дисертація: 105 с., 32 рис., 19 табл., 19 посилань.

КРИПТОВАЛЮТНА БІРЖА, СЕМАНТИЧНИЙ АНАЛІЗ, НЕЙРОННІ МЕРЕЖІ, ОБРОБКА ПРИРОДНОЇ МОВИ, ЧАСОВІ РЯДИ, ПРОГНОЗУВАННЯ, ФІНАНСОВІ ЦИКЛИ, ФРАКТАЛЬНА РОЗМІРНІСТЬ, LSTM, NLTK, NLP, BTC.

Об’єктом дослідження є криптовалютна біржа з 2014 по кінець 2023 року, саме ціна на біткоїн за цей період. Також досліджуються твіти з Твіттеру за період з 2018 по кінець 2023 року.

Предметом дослідження є прогнозування ціни на біткоїн за допомогою штучного інтелекту. При цьому цей процес буде складатися з аналізу тональності новин, історичної ціни та об’єму біткоїну.

Мета роботи – дослідити існуючі підходи для прогнозування фінансових ринків, семантичного аналізу новин. Вибрати найкращі підходи та створити на їх базі власну систему з комплексним підходом для прогнозування криптовалюти.

Виконано дослідження існуючих підходів для прогнозування часових рядів, з них найкращим стала рекурентна нейронна мережа – LSTM. Досліджені підходи в сфері NLP для аналізу тональності новин.

В результаті виконання роботи розроблена система для прогнозування ціни біткоїну. Система складається з трьох підсистем: підсистема збору даних, семантичного аналізу твітів, прогнозування ціни на біткоїн.

Результати роботи прийнято до опублікування у періодичному виданні, яке індексуєтсья у наукометричній базі Scopus.

ABSTRACT

Master's thesis: 105 pp., 32 figs., 19 tables, 19 bibl. ref.

CRYPTOCURRENCY EXCHANGE, SEMANTIC ANALYSIS, NEURAL NETWORKS, NATURAL LANGUAGE PROCESSING, TIME SERIES, FORECASTING, FINANCIAL CYCLES, FRACTAL DIMENSIONALITY, LSTM, NLTK, NLP, BTC.

The research object is a cryptocurrency exchange from 2014 to the end of 2023, specifically the price of Bitcoin during this period. Tweets from Twitter for the period from 2018 to the end of 2023 are also studied.

The subject of the research is the prediction of Bitcoin price using artificial intelligence. This process will consist of analyzing the tone of news, historical price, and volume of Bitcoin.

The aim of the work is to investigate existing approaches to forecasting financial markets, semantic analysis of news. Select the best approaches and create a system based on them with a comprehensive approach to predicting cryptocurrency.

Research on existing approaches for forecasting time series has been conducted, with the best being the recurrent neural network – LSTM. Approaches in the field of NLP for analyzing the tone of news have been studied.

As a result of the work, a system for predicting the price of Bitcoin has been developed. The system consists of three subsystems: data collection, semantic analysis of tweets, and predicting the price of Bitcoin.

The results of the work have been accepted for publication in a periodic publication indexed in the scientometric database Scopus.

ЗМІСТ

ВСТУП

Прогнозування фінансових ринків завжди було серйозним та актуальним завданням. Людське прагнення антиципувати майбутні події намагається знайти застосування у сфері економіки, зокрема на фінансовому ринку. Проте, через надзвичайну складність і непередбачуваність цих ринків, побудова стабільної та надійної математичної моделі для прогнозування їхньої динаміки залишається викликом.

Торгівля криптовалютами несе в собі ще більше ризиків. Волатильність таких активів може бути порівняна лише з американськими гірками - вона може швидко зрости на фоні позитивних новин, але так само раптово може впасти. Попри велику нестабільність, багато трейдерів вбачають у криптовалюті можливість для збільшення своїх інвестицій, використовуючи цю волатильність на свою користь.

У цій науковій роботі ми намагатимемося розробити систему, яка не тільки спрогнозує рухи біткоїна, але й включатиме аналіз настрою на ринку на основі даних з Твіттера. Для цього ми використовуватимемо різні математичні підходи, включаючи фрактальну розмірність, і звісно, застосовуватимемо передові технології глибокого навчання та нейронних мереж.

Метою розробленої системи буде надання актуального прогнозу для трейдерів, що бажають отримувати максимальний прибуток від торгівлі на криптовалютній біржі. Ця система покладе початок новому інструменту, який в майбутньому може об'єднати збір інформації, аналітику, прогнозування та навіть автоматизовану торгівлю в режимі реального часу.

РОЗДІЛ 1 КРИПТОВАЛЮТА ТА ЇЇ ВИКОРИСТАННЯ В СВІТІ

1.1 Характеристика циклічності фінансових ринків

1.1.1 Природа циклічності

Цикл описується як набір пов'язаних подій, процесів або активностей, які утворюють замкнене коло дій за визначений часовий період. Циклічність у фінансах визначається як повторюваність економічних трендів із заздалегідь визначеною регулярністю та чітко визначеними етапами ринкових змін. Ця регулярність була встановлена за допомогою емпіричних досліджень, проведених протягом багатьох десятиліть або навіть століть. Глобальна економічна спільнота активно досліджує критичні точки циклів, взаємодію між циклами різної тривалості, вплив різних умов та їх вплив на різні сфери економіки. Особливий інтерес становить дослідження факторів, які можуть згладжувати або змінювати тривалість різних фаз циклу.

1.1.2 Фази Економічних Циклів

В економічному контексті вирізняють чотири основні фази циклу, які чергуються:

1. Підйом, який характеризується збільшенням основних економічних індексів. На цій стадії спостерігається зростання ключових показників після періоду кризи, що веде до збільшення внутрішнього валового продукту, зниження безробіття через створення нових робочих місць, зростання споживчого попиту та зміна цінової політики. Позитивні тенденції також спостерігаються в інвестиціях.
2. Економічний бум означає досягнення максимальних показників ділової активності, після чого система перестає розвиватися.

3. Спад відбувається, коли основні індикатори починають знижуватися. Ця фаза характеризується уповільненням виробничого процесу, скороченням виробництва та робочих місць, що призводить до збитків компаній та масових банкрутств. Знижується споживчий попит і загальний рівень життя населення.

4. Криза є точкою, де всі ключові економічні індикатори досягають свого мінімуму. Економіка адаптується до нових умов, що вимагає новаторських рішень і підходів. Як і під час буму, економіка зупиняється, не відбуваючись ні позитивних, ні негативних змін.

1.1.3 Класифікація економічних циклів за тривалістю

Різні види економічних циклів розглядаються у науковій літературі, із найбільш розповсюдженим підходом є їхнє розподілення за тривалістю. На це вказує табл. 1.1.

Таблиця 1.1 – Види циклів по тривалістю

Назва	*Характерний період*
Короткострокові цикли Китчина	2-3 роки
Середньострокові цикли Жюгляра	7-11 років
Цикли Кузнеця	15-20 років
Довгі хвилі Кондратьєва	48-55 років

Короткострокові Цикли Китчина

Ці економічні цикли були названі на честь Джозефа Китчина, видатного британського статистика та бізнесмена, який виявив їх через аналіз статистичних даних ділової активності у Великобританії та США. Цикл Китчина відомий як найкоротший з усіх економічних циклів, тривалістю від 3 до 4 років. Він є частиною більш довготривалих економічних циклів, таких як цикли Жюгляра, Кузнеця та Кондратьєва.

Джозеф Китчин працював у золотодобувній промисловості, і його дослідження часто асоціювали з коливаннями світових запасів золота. Проте сьогоднішнє розуміння циклів Китчина базується на інших теоретичних підходах.

Цикли Китчина вважаються результатом затримок між подією та реакцією на неї, які можуть бути зумовлені:

1. Психологічним фактором - часом, необхідним для аналізу інформації управлінцями виробничих структур.

2. Технологічним фактором – часом, потрібним для впровадження прийнятих рішень у виробництво. Ці затримки створюють дисбаланс, коли прийняті дії починають впливати лише під час піку події та продовжують діяти, навіть коли ця подія втрачає свою актуальність, що призводить до формування циклів Китчина.

Середньострокові Цикли Жюгляра. Ці цикли були визначені Клементом Жюгляром та характеризуються періодичністю від 7 до 11 років, становлячи проміжний етап між Китчинськими та більш тривалими економічними циклами.

Цикли Китчина виникають через затримку в передачі інформації про факти господарської діяльності до управлінської структури компанії, у той час як цикли Жюгляра охоплюють більш широкий спектр затримок, зокрема у сфері інвестицій. Ці затримки важливі, оскільки інвестиції в основний капітал здійснюються не миттєво, а вимагають планування, розробки виробничих потужностей та їхнього подальшого впровадження.

Цикли Жюгляра поділяються на дві основні групи: інноваційні та цикли зростання, кожен з яких має свою структуру і стадії розвитку:

1. Пожвавлення, що характеризується поступовим зростанням виробничих потужностей, поділяється на:

- старт: ключовий поштовх, зумовлений прийняттям важливого рішення;
- прискорення: етап розширення виробничих потужностей з високою швидкістю інвестицій.

2. Процвітання, на цій стадії інвестиції дають свої плоди, виробництво досягає піку:

- зростання: виробничі обсяги задовольняють попит;
- перегрів: продовження росту виробництва та інвестицій, незважаючи на перекриття попиту.

3. Рецесія, характеризується спадом діяльності через надмірні інвестиції:

- крах: витрати не окупаються;
- спад: зменшення інвестицій та виробничих обсягів.

4. Депресія, дно економічного циклу, коли економіка шукає баланс:

- стабілізація: попит забезпечується наявними запасами та невисоким виробництвом;
- зрушення: нестача ресурсів спонукає до нових інвестицій.

Ці цикли тісно пов'язані, де короткострокові цикли є частиною триваліших, формуючи складну структуру економічних ритмів.

Цикли Кузнеця.

Ці цикли названі на честь американського економіста Саймона Кузнеця, який вивчив економічні тенденції, що сформувалися після Другої світової війни. У цей час виникла необхідність в довгостроковому плануванні та прогнозуванні економічних процесів, внаслідок чого особлива увага була звернена на циклічні коливання в економіці.

Саймон Кузнець зосередився на вивченні причин, що сприяють виникненню цих циклів, включаючи такі фактори, як демографічні зрушення, інвестиції в капітальне будівництво, перерозподіл капіталу та зміни у національному доході. Він дійшов висновку, що головними драйверами циклічності є довготермінові інвестиції в капітальні об'єкти та демографічні зміни.

Аналізуючи період в 60 років, Кузнець виявив, що ці цикли є довшими за звичайні короткострокові економічні цикли, але коротшими за довгі хвилі Кондратьєва, маючи періодичність 15-20 років.

Фази хвиль Кузнеця охоплюють:

1. Підйом: починається з оживлення після кризи, що веде до створення нових робочих місць, зростання споживчого попиту, розширення виробництва та збільшення ВВП на душу населення.

2. Пік: максимальна точка економічного підйому, де основні економічні індикатори досягають своєї кульмінації.

3. Спад: фаза, яка настає після піку, характеризується зниженням основних економічних показників.

4. Криза: нижня точка економічного циклу, що характеризується високим рівнем безробіття, низьким попитом на товари та масовими банкрутствами. Після цього етапу виникають нові ідеї та підходи, що ведуть до початку нової ітерації циклу.

Ці цикли Кузнеця можна спостерігати у світовій економіці, наприклад, у моделі коливань цін на ринку нерухомості в Японії у 1980-2000 роках. Графічне зображення цих хвиль має вигляд синусоїди, де екстремуми представляють максимальний і мінімальний розвиток економічної системи в рамках циклу.

Хвилі Кондратьєва.

Хвилі Кондратьєва, іменовані на честь їх відкривача Миколи Кондратьєва, є довготерміновими економічними циклами, ідентифікованими емпірично через аналіз макроекономічних показників світових держав протягом 100-150 років. Ці цикли, також відомі як К-цикли або К-хвилі, мають періодичність від 45 до 60 років і відображають цикли підйомів та спадів світової економіки.

Теорія Кондратьєва базується на припущенні, що різні господарські блага та матеріально-технічні ресурси мають обмежений термін існування і що для створення нових господарських благ потрібний час, умови та ресурси. Ці цикли є результатом коливань у рівновазі економічної системи, пов'язаних із періодами накопичення та розподілу капіталу, а також з впровадженням нових благ.

Фази хвиль Кондратьєва включають:

1. Фаза зростання: зазвичай починається з подій, що вимагають великих витрат, наприклад, війни. Характеризується зростанням виробництва, впровадженням нових відкриттів, інфляцією та розвитком міжнародної торгівлі.

2. Фаза вершини: початок зміни загальної тенденції у світовій економіці, супроводжується сильною інфляцією та спробами стабілізувати фінансову ситуацію.

3. Фаза зниження: відзначається зниженням рівня інфляції та процентних ставок, регулюванням фінансових ринків, спадом попиту на товари, що впливає на виробництво та ціни.

4. Фаза депресії: мінімальні темпи інфляції та процентних ставок, кредити стають доступними, але попит на них відсутній. Ця фаза часто багата на нові винаходи, що закладають основу для майбутнього розвитку.

Ці великі цикли включають менші, кожен з яких також має фази підйому та спаду. Суть циклічності полягає у заміні старих благ та цінностей на нові, особливо на критичних точках циклу, які сприяють формуванню та зростанню нових ідей і технологій. Ці цикли можна пов'язати з появою нових технологічних напрямків.

Хвилі Елліота.

Хвилі Елліота – це концепція, розроблена Ральфом Нельсоном Елліотом, яка описує циклічність фінансових ринків, особливо акційних ринків, на основі психології інвесторів та колективного поведінки ринку. Ця теорія стала особливо популярною серед трейдерів і аналітиків, які використовують її для прогнозування ринкових трендів.

Елліот вважав, що рухи цін на ринках мають хвилеподібний характер і є відображенням коливань настроїв учасників ринку. Він ідентифікував зразок, згідно з яким ринкові цикли розгортаються в серії з п'яти "хвиль", які він назвав "імпульсними хвилями", за якими слідують три "корекційні хвилі".

Основні елементи теорії хвиль Елліота включають:

1. Імпульсні хвилі: ці хвилі складаються з п'яти підхвиль, які рухаються в основному напрямку тренду. Вони представляють фазу активного росту або зниження на ринку.

2. Корекційні хвилі: це хвилі, які рухаються проти основного тренду і складаються з трьох підхвиль. Вони відображають періоди виправлення після імпульсної фази.

3. Фрактальна природа: хвилі Елліота мають фрактальну природу, що означає, що їх можна спостерігати на різних часових рамках, від короткострокових до довгострокових.

4. Психологія ринку: хвилі відображають психологічний стан учасників ринку, де кожна хвиля відображає зміну настроїв – від оптимізму до песимізму та навпаки.

5. Застосування у прогнозуванні: трейдери використовують хвилі Елліота для визначення потенційних зон повороту на ринку, оцінюючи, на якій стадії розвитку знаходиться певна хвиля.

Хвилі Елліота не є точною наукою, але їх часто використовують як частину технічного аналізу для інтерпретації ринкових настроїв та ідентифікації потенційних трендів. Ця теорія заснована більше на психології та поведінці інвесторів, ніж на традиційних фундаментальних або економічних аналізах.

1.2 Криптовалюта: її природа та глобальне застосування

Криптовалюта представляє собою цифровий ресурс, який базується на криптографічних методах для забезпечення своєї безпеки та цілісності. Ці віртуальні валюти, в основному, стали засобом обміну для придбання товарів та послуг в інтернеті. Водночас, деякі криптовалюти надають своїм власникам певні права або привілеї. Важливо зазначити, що ці діджитальні валюти не мають фізичного підкладу та не контролюються жодними державними структурами, через що їх легітимність як платіжного засобу в ряді країн залишається під питанням [26].

Згідно досліджень, у світі існує більше 100 мільйонів активних користувачів криптовалют, із яких значна частина вважає цей актив дієвим інвестиційним інструментом. Як правило, люди використовують криптовалюти для електронних покупок, обираючи їх завдяки можливості здійснювати транзакції без розкриття особистої інформації. Втім, необхідно розуміти, що, хоча певний рівень анонімності зберігається, повна конфіденційність не гарантована.

Однією з ключових переваг криптовалют є їхня незалежність від традиційних фінансових інституцій. Це дозволяє користувачам зменшити

транзакційні комісії та уникнути ризиків, пов'язаних із зловживанням даними у банках. Додатково, певні криптовалюти можуть надавати користувачам додаткові права, наприклад, участь у голосуванні по питаннях розвитку певного проекту або долю власності в реальних активах, таких як об'єкти нерухомості чи арт-предмети.

1.3 Формування ринку криптовалют

Формування та коливання цін на криптовалюту в значній мірі визначаються попитом та пропозицією. З огляду на децентралізовану природу криптовалют, їхні ринки не так сильно підвержені впливу політичних або економічних подій, що характерні для традиційних валют, наприклад, переобрання глави держави.

Однак, є чітко виділені фактори, що можуть вплинути на ціну криптовалюти. Серед них: кількість монет та темп їхнього випуску або знищення; загальна ринкова капіталізація; засоби масової інформації і їхнє освітлення крипторинку; адаптація криптовалюти у сучасних платіжних системах та іншій інфраструктурі; важливі події, які включають в себе регулятивні зміни, витоки даних або економічні кризи.

Специфічно, соціальні медіа і, зокрема, Twitter, відіграють значущу роль у формуванні сприйняття та настроїв у спільноті криптовалют. Твіти від відомих особистостей чи авторитетних експертів можуть спричинити раптові зміни у вартості криптовалют. Це особливо відчутно, коли особи з великим числом підписників або значущий вплив на ринок роблять коментарі або рекомендації щодо конкретних криптовалют. Тому активні учасники ринку часто слідкують за такими повідомленнями, намагаючись вгадати настрої та відгуки громадськості.

1.4 Види криптовалют

Великий асортимент криптовалют, що наразі присутній на ринку, налічує понад тисячу найменувань. Однак, сконцентруємо увагу на найбільш популярних серед них:

1. Bitcoin (BTC): ця криптовалюта була створена в 2008 році невідомою особою або групою під псевдонімом Сатосі Накамото. У 2009 році для загального доступу було представлено її програмне забезпечення. Вона стала першою криптовалютою в історії, і сьогодні володіє найбільшою ринковою капіталізацією.

2. Ethereum (ETH): представлена як децентралізована платформа, Ethereum використовує розумні контракти та децентралізовані додатки. Цю криптовалюту анонсував у 2013 році Віталік Бутерін, і вона швидко стала популярною альтернативою Bitcoin.

3. Litecoin (LTC): Чарлі Лі, інженер з Google, представив Litecoin у 2011 році. Особливість цієї криптовалюти полягає в її здатності генерувати блоки швидше, ніж Bitcoin. Також вона використовує унікальний алгоритм майнінгу, який робить неможливим використання суперпотужних комп'ютерів для її видобутку.

Ці криптовалюти є ключовими гравцями на ринку і мають значний вплив на його динаміку.

1.5 Накладання циклів на історичні данні ціни Біткоіну

Аналізуючи графік біткоїна з точки зору теорій циклів Кондратьєва, Кузнеця, Жугляра, Китчина, а також хвиль Елліота, ми можемо спробувати зрозуміти, як ці теорії відображаються на реальних ринкових рухах криптовалюти. Важливо пам'ятати, що хоча ці теорії надають цінний інструмент

для аналізу фінансових ринків, вони не можуть забезпечити точні прогнози, особливо в контексті такого непередбачуваного та волатильного активу, як біткоїн.

1. Цикли Кондратьєва і Кузнеця: враховуючи, що ці цикли мають тривалість від декількох десятиліть до півстоліття, вони можуть бути недостатньо чіткими для аналізу поведінки біткоїна, який існує менше ніж 15 років. Однак, загальний тренд зростання вартості біткоїна може відображати елементи цих довготривалих циклів, особливо в контексті збільшення інтересу до криптовалют і розвитку цифрових технологій.

2. Цикл Жугляра: цей середньостроковий цикл, який триває 7-11 років, може бути більш релевантним для аналізу біткоїна. Проте, через відносно коротку історію існування біткоїна, важко встановити чітку кореляцію між його ціновими змінами та циклами Жугляра.

3. Цикли Китчина: ці короткострокові цикли, які тривають 3-4 роки, можуть бути більш примітними у контексті біткоїна. Певні співпадіння між циклами Китчина та короткостроковими коливаннями ціни біткоїна можуть бути ідентифіковані, але це може бути також випадковістю, а не вказівкою на суттєву кореляцію.

4. Хвилі Елліота: ці хвилі можуть надати додаткове розуміння щодо короткострокових і середньострокових цінових трендів біткоїна. Оскільки хвилі Елліота описують психологію учасників ринку, їх застосування до аналізу біткоїна може виявити певні повторювані патерни в поведінці інвесторів. Однак, важливо пам'ятати, що криптовалютні ринки часто впливаються зовнішніми факторами, які можуть порушувати будь-які звичайні моделі.

1.6 Постановка задачі

Моделювання та прогнозування курсів криптовалют вимагає високої точності, адже ціни на криптовалютному ринку можуть мінятися щохвилини.

Особливо коли мова йде про вплив соціальних медіа, зокрема Twitter, на динаміку цін.

Криптовалютний ринок відомий своєю високою волатильністю, яка може викликатися рядом чинників, зокрема новинами, що розповсюджуються у соціальних мережах. Twitter є однією з основних платформ, де обговорюються новини зі світу криптовалют, і через це може мати вплив на настрої учасників ринку.

Для аналізу та прогнозування курсів криптовалют я обрав наступний підхід:

1. Аналізувати Twitter за допомогою моделі NLP: ця модель буде оцінювати важливість новин, робити аналіз настроїв (сентімент-аналіз) та визначати їхній потенціальний вплив на ринок.

2. Використовувати глибокі нейронні мережі: на основі історичних даних та отриманих з моделі NLP результатів буде побудована модель глибокого навчання, яка буде прогнозувати можливі зміни курсу криптовалют.

Моїми цілями в цьому дослідженні є:

- дослідження та аналіз даних з Twitter та історичних даних курсів криптовалют;
- розробка моделі NLP для аналізу новин та визначення їх важливості та настроїв.
- побудова та навчання глибоких нейронних мереж для прогнозування курсу на основі отриманих даних.
- оцінювання точності та надійності обох моделей.
- розробка комплексної системи підтримки прийняття рішень для трейдерів та інвесторів.

З врахуванням високої волатильності криптовалют і великої кількості інформації у соціальних мережах, такий підхід може значно покращити точність прогнозування і допомогти трейдерам у прийнятті рішень на ринку криптовалют.

Висновки до розділу 1

Розділ 1 дипломної роботи надає всебічний аналіз циклічності фінансових ринків, їхньої природи та різних фаз. Дослідження циклів Китчина, Жугляра, Кузнеця та Кондратьєва виявляє, як ці цикли впливають на економіку в загальному та на фінансові ринки зокрема. Ключовим аспектом є розуміння того, як ці цикли взаємодіють та формують динаміку економіки. Важливість цього аналізу зростає, коли ми розглядаємо криптовалюти – новітній та швидкозмінний сегмент фінансових ринків. Криптовалюти вносять нові елементи у вже існуючу фінансову структуру світу, завдяки їх незалежності від традиційних фінансових інституцій та їх децентралізованій природі. Розуміння динаміки цін на криптовалюти, їхнього впливу на ринки та зв'язку з циклічними теоріями є ключовим для ефективного моделювання та прогнозування курсів цих валют. Це дослідження також підкреслює роль соціальних медіа, зокрема Twitter, у формуванні ринкових настроїв та цін на криптовалюти, вказуючи на необхідність інтеграції технологій NLP та глибокого навчання для розробки комплексних систем підтримки прийняття рішень на ринку криптовалют.

РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ПРОГНОЗУВАННЯ

2.1 Математичні моделі прогнозування часових рядів

Часовий ряд у контексті криптовалют - це послідовність значень динаміки ціни криптовалюти, що змінюється в певний період часу. Відмінність цього ряду від простої вибірки полягає в його хронологічній послідовності: значення ціни криптовалюти залежать одне від одного і їх порядок є суттєвим. Особливість криптовалютних ринків полягає в їх високій волатильності, що робить аналіз часових рядів особливо важливим.

Головні мети аналізу часових рядів для криптовалют включають розуміння загальних трендів ринку, виявлення можливих майбутніх змін цін та створення стратегій для інвестування чи торгівлі. Для досягнення цих цілей необхідно ідентифікувати і формально описати модель часового ряду.

В аналізі часових рядів приймається, що дані містять систематичну (регулярну) та випадкову (шумову) складові. Криптовалютні ринки часто мають яскраво виражені:

- тренди: це може бути підйом або спад цін криптовалюти протягом певного часового проміжку, наприклад, зростання популярності Bitcoin через прийняття його новими компаніями;

- сезонні коливання: наприклад, збільшення активності торгівлі під час певних глобальних подій або на свята.

Основні складові часового ряду для криптовалют можна представити як суму або добуток трендової, сезонної та випадкової компонент. Це допомагає в розробці стратегій для інвестування та прогнозування майбутньої поведінки ринку.

Знаходження трендів у криптовалютних часових рядах може бути складним завданням через їх високу волатильність. Згладжування - один із способів виявлення цих трендів. Використовуючи методи, такі як ковзне середнє або медіанне згладжування, аналітики можуть виявляти основні тренди, відокремлюючи їх від короткочасних коливань ціни.

2.2 Модель SARIMA

Модель SARIMA, звана моделлю сезонного авторегресійного інтегрованого ковзного середнього (Seasonal Autoregressive Integrated Moving Average Model), є методом прогнозування часових рядів, запропонованим Боксом і Дженкінсом на початку 1970-х років. Модель SARIMA відноситься до моделі, створеної шляхом регресії залежної змінної тільки за її значенням запізнювання і поточного значення, і значення запізнення члена випадкової помилки в процесі перетворення нестаціонарних часових рядів в стаціонарні часові ряди.

Основна ідея моделі SARIMA: обробляти послідовність даних, сформовану передбаченим об'єктом з часом, як випадкову послідовність, і використовувати певну математичну модель, щоб описати цю послідовність. Як тільки ця модель ідентифікована, майбутня вартість може бути передбачена з минулих та поточних значень часового ряду [20].

Модель SARIMA(p, d, q) для нестаціонарних часових рядів x_t має форму:

$$\Delta^d X_t = c + \sum_{i=1}^{p} a_i \Delta^d X_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} + \varepsilon_t \,,$$

де ε_t– стаціонарні часові ряди,

c, a_i, b_j – є параметрами моделі.

Δ^d – оператор різниці часових рядів порядку d (послідовно беручи d разів відмінності першого порядку – спочатку від часового ряду, потім від отриманих відмінностей першого порядку, потім від другого порядку і т.д.).

Крім того, ця модель інтерпретується як SARIMA(p + d, q) - модель з d поодинокими коренями. При $d=0$ у нас є звичайні SARIMA-моделі.

SARIMA (p, d, q) називається моделлю диференціального авторегрессивного ковзного середнього, яка включає процес ковзного середнього (MA), процес авторегрессии (AR), авторегрессивное ковзне середнє, в залежності від того, чи стабільна вихідна послідовність і які частини включені в регресію, процес (ARMA) та авторегресія змішаного ковзного середнього (SARIMA).

1.Процес ковзної середньої (MA (q)).

2.Авторегресійний процес (AR(p)).

3.Процес авторегресії ковзного середнього (ARMA (p, q)).

4.Процес авторегресії інтегрального ковзного середнього (ARIMA (p, d, q)).

Де AR - авторегрессія, p - авторегрессія, MA - ковзне середнє, q - кількість елементів ковзного середнього, а d - кількість відмінностей, зроблених, коли часовий ряд стає стаціонарним часовим рядом.

2.3 Методи рекурентних нейронних мереж

Сучасний світ технологій активно використовує штучні нейронні мережі у найрізноманітніших сферах: від розпізнавання облич до чат-ботів в соціальних мережах. Особливої уваги заслуговують рекурентні нейронні мережі (RNN), які стали основою для прогнозування в ряді важливих завдань, включаючи ринок криптовалют [18].

Рекурентні нейронні мережі (RNN) зазнали свого розвитку завдяки мережі Хопфілда. В 1993 році ця нейронна система успішно виконала завдання «дуже глибокого навчання» з використанням понад 1000 послідовних шарів. Особливість RNN полягає у використанні попередніх станів мережі для визначення поточного стану. Такий підхід стає незамінним при роботі з послідовностями даних, як-то історія зміни цін криптовалют.

RNN (Recurrent neural network), рекурентна нейронна мережа починає свій розвиток у мережі Хопфілда. У 1993 нейронна система запам'ятовування та стиснення історичних даних змогла вирішити завдання «дуже глибокого навчання», в якій у рекурентній мережі розгорталося понад 1000 послідовних шарів. Дана технологія базується на використанні минулих станів мережі, для визначення сьогодення. Такі мережі на практиці зазвичай використовують, коли вхідні дані завдання є нефіксованою послідовністю значень. На кожному кроці

«навчання» t значення $ht \in \mathbb{R}^m$ прихованого рекурентного шару нейронної мережі визначається наступним чином:

$$ht = f(Wxt + Uht{-}1 + b\ h\ )$$

де, $xt \in \mathbb{R}^m$ – вхідний вектор в момент часу t, $W \in \mathbb{R}^{m*n}$,$U \in \mathbb{R}^{m*m}$, $b\ h \in \mathbb{R}^m$ – параметри рекурентної нейронної мережі, що навчаються, f – функція нелінійного перетворення. Найчастіше в якості нелінійного перетворення використовують одну з наступних функцій: − Сигмоїда: $f(x) = 1\ 1{+}e{-}x$ − Гіперболічний тангенс: $f(x) = e\ x{-}e\ {-}x\ e\ x{+}e{-}x$. Функція-випрямляч ReLU: $f(x)$ = max (0, x).

В простій рекурентній нейронній мережі, що показана на рис. 2.1, вихідне значення y_t на кроці t рахується за формулою: $y_t = Wh_t + b$, де W, b – навчальні параметри.

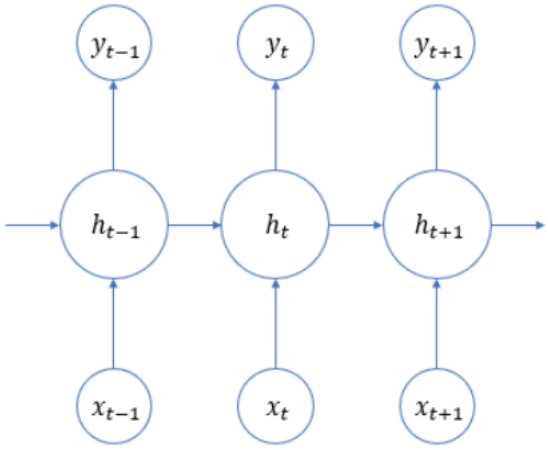

Рисунок 2.1 – Проста рекурентна нейронна мережа [18]

Проста RNN дуже добре працює для задач підбору контексту на базі попередніх кроків збереження інформації [21]. Наприклад підбір фрагментів для відеоролика, або передбачення наступного слова у тексті. Але існують випадки коли потрібно звернутися до інформації, згаданої кілька кроків назад. І на жаль, на практиці, при збільшенні розриву RNN втрачають зв'язок між інформацією. Тому, щоб усунути цю проблему, було розроблено метод LSTM нейронної мережі.

2.4 LSTM модель

LSTM (long short-term memory, дослівно (довга короткострокова пам'ять) — тип рекурентної нейронної мережі, здатний навчатися довгостроковим залежностям. LSTM були представлені в роботі Hochreiter & Schmidhuber (1997) [22].

LSTM спеціально розроблені для вирішення проблеми довгострокової залежності. Їхня спеціалізація — запам'ятовування інформації протягом тривалих періодів часу, тому їх навчання займає значно менший час.

Всі рекурентні нейронні мережі мають форму ланцюжка модулів нейронної мережі, що повторюються. У стандартних RNN цей повторюваний модуль має просту структуру, наприклад, один шар tanh [23].

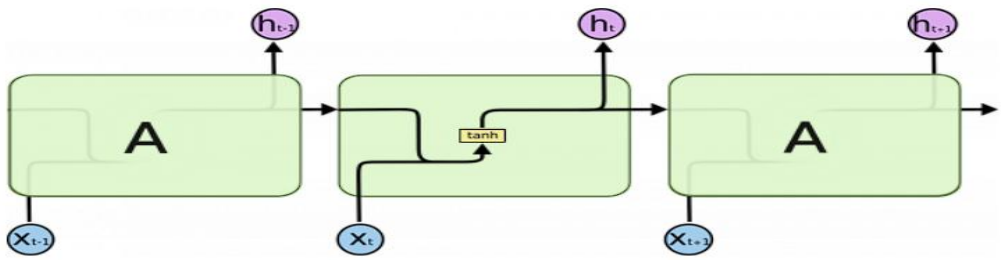

Рисунок 2.2 — Модуль, що складається з одного шару [18]

На представленій діаграмі (рис 2.2) кожна лінія є вектором. Рожеве коло означає крапкові операції, наприклад, підсумовування векторів. Під жовтими осередками розуміються шари нейронної мережі. Поєднання ліній є об'єднання векторів, а знак розгалуження - копіювання вектора з подальшим зберіганням у різних місцях (рис. 2.3).

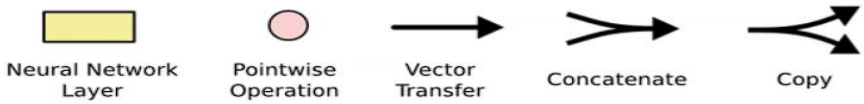

Рисунок 2.3 — Позначення [18]

Ключовим поняттям LSTM є стан осередку: горизонтальна лінія, що проходить через верхню частину діаграми (рис 2.4).

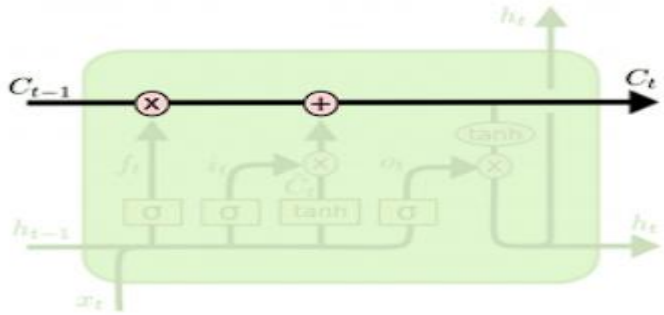

Рисунок 2.4— Стан осередку [18]

Стан осередку нагадує конвеєрну стрічку. Він проходить через весь ланцюжок, зазнаючи незначних лінійних перетворень.

LSTM зменшує або збільшує кількість інформації в стані осередку, залежно від потреб. Для цього використовуються структури, що ретельно настроюються, так звані гейти.

Гейт - це "ворота", що пропускають або не пропускають інформацію. Гейти складаються із сигмовидного шару нейронної мережі та операції поточкового множення.

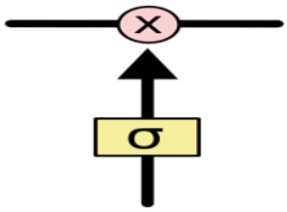

Рисунок 2.5 — Гейт [18]

На виході сигмовидного шару видаються числа від нуля до одиниці, визначаючи скільки відсотків кожної одиниці інформації пропустити далі. Значення "0" означає "не пропустити нічого", значення "1" - "пропустити все". LSTM має три такі гейти для контролю стану осередку.

На першому етапі LSTM потрібно вирішити, яку інформацію ми збираємось викинути зі стану комірки. Це рішення приймається сигмовидним шаром, званим шаром гейту втрати. 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 - "цілком видалити"(рис. 2.6).

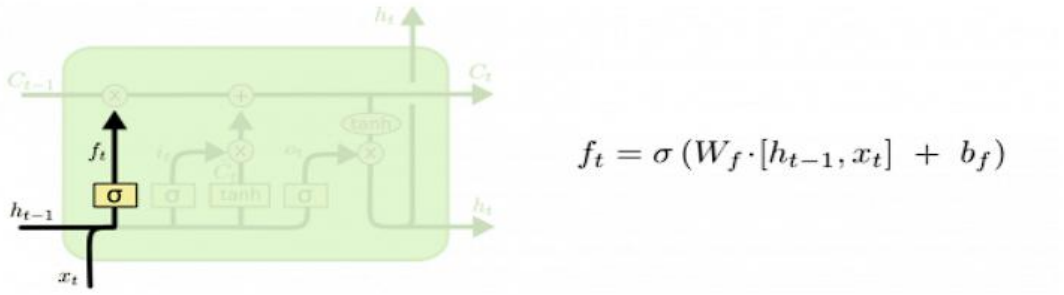

Рисунок 2.6 — Шар втрати [18]

На наступному етапі необхідно вирішити, яку нову інформацію зберегти в стані осередку. Розіб'ємо процес на дві частини. Спочатку сигмоїдний шар, званий шаром гейта входу, вирішує, які значення потрібно оновити. Потім шар tanh створює вектор нових значень-кандидатів C, які додаються до стану. Наступний крок об'єднає ці два значення для оновлення стану (рис 2.7).

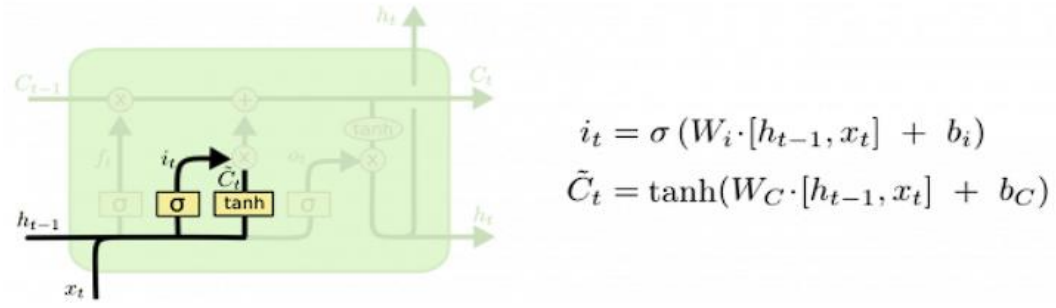

Рисунок 2.7 — Шар збереження [18]

Тепер оновимо попередній стан осередку для отримання нового стану C. Спосіб оновлення вибраний, тепер реалізуємо саме оновлення.

Помножимо старий стан на f, втрачаючи інформацію, яку вирішили забути. Потім додаємо i*C. Це нові значення кандидатів, які масштабуються залежно від того, як ми вирішили оновити кожне значення стану (рис. 2.8).

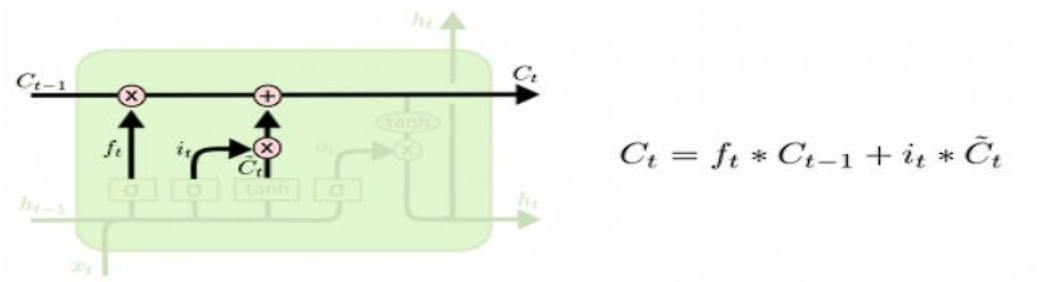

Рисунок 2.8 — Новий стан [18]

Зрештою, треба вирішити, що хочемо отримати на виході. Результат буде відфільтрованим станом осередку. Спочатку запускаємо сигмоїдний шар, який вирішує, які частини стану осередку виводити(рис 2.9). Потім пропускаємо стан осередку через tanh (щоб розмістити всі значення в інтервалі [-1, 1]) і множимо його на вихідний сигнал гейта сигмовидного.

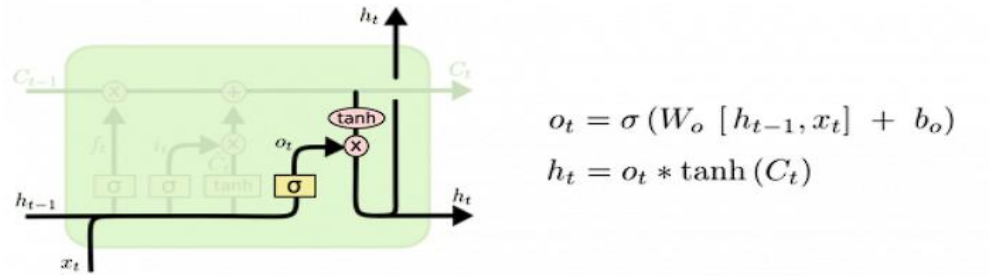

Рисунок 2.9 — Результат на виході [18]

2.5 Технологія обробки природної мови (NLP)

У сучасному цифровому світі соціальні мережі відіграють величезну роль в формуванні громадської думки. Інформація, що ширириться через платформи як Twitter, може впливати на ринкові настрої, особливо у сфері криптовалют. Технологія обробки природної мови (NLP) дозволяє автоматизовано аналізувати текстові дані і виявляти корисну інформацію з величезних обсягів даних.

Основні компоненти NLP для аналізу твітів:

Попередня обробка тексту: цей етап включає в себе видалення зайвих символів, URL-адрес, згадок користувачів (наприклад, @username) та інших непотрібних елементів. Також можуть застосовуватися стемінг (приведення слова до його основи) та лематизація (приведення слова до його словникової форми).

Токенізація: розбиття тексту на окремі слова або фрази. Це полегшує аналіз окремих одиниць тексту.

Сентимент-аналіз: дозволяє визначити емоційний заряд твіту – позитивний, негативний або нейтральний. Це допомагає зрозуміти загальний настрій громадськості щодо певної криптовалюти.

Тематичне моделювання: визначення ключових тем у тексті. Наприклад, якщо більшість твітів говорять про партнерство між криптовалютою і великою корпорацією, це може вказувати на позитивний стимул для зростання ціни.

Однією з ключових особливостей ринку криптовалют є його висока чутливість до новин та публічних висловлювань видатних осіб. Зокрема, твіти Ілона Маска, генерального директора Tesla і SpaceX, виявилися вельми впливовими на рух цін криптовалют у 2021 році.

Приклади твітів Ілона Маска та їх вплив:

1. Твіт про Bitcoin: у лютому 2021 року, коли Tesla вклала 1,5 мільярда доларів у Bitcoin і заявила про прийом цієї криптовалюти як оплати за автомобілі, Ілон Маск твітнув: "Тільки в тупому додатку можна купити тупий автомобіль". Після цього ціна Bitcoin різко піднялася.
2. Твіт про Dogecoin: у кілька місяців Маск написав: "Dogecoin – народна

криптовалюта". Цей твіт призвів до миттєвого зростання ціни Dogecoin більш ніж на 40%.

3. Твіт про екологічність Bitcoin: у травні 2021 року Маск твітнув про те, що Tesla призупиняє прийом Bitcoin як оплати через питання екологічності виробництва криптовалюти. Це призвело до різкого зниження ціни Bitcoin.

Ці приклади показують, як великий вплив може мати один твіт від відомої особистості на ринкову вартість криптовалюти. В результаті багато трейдерів і аналітиків почали використовувати NLP для моніторингу і аналізу твітів від ключових осіб для прогнозування руху цін.
Твіттер, як і більшість соціальних мереж, використовує спеціалізований алгоритм для визначення контенту, який відображається у стрічці користувача. Цей алгоритм базується на ряді факторів, включаючи активність (лайки, ретвіти, відповіді), імпрешони (кількість показів твіту користувачам) та інші метрики.

Твіти, які отримують велику кількість імпрешонів, лайків та ретвітів, часто вважаються більш важливими або цікавими для користувачів, і тому їх може бути показано пріоритетно у стрічці.

Необхідність збору метрик:

1. Аналіз активності: збір метрик допомагає зрозуміти, які твіти найбільше зацікавлюють аудиторію. Лайки, ретвіти та відповіді відображають реакцію аудиторії на конкретний твіт.
2. Оцінка впливу: твіти з великою кількістю імпрешонів можуть мати великий вплив, навіть якщо їх активно не ретвітять або не лайкають. Вони досягли великої аудиторії.
3. Оптимізація стратегії: аналізуючи метрики, компанії можуть коригувати свою маркетингову стратегію на Твіттері, зосереджуючись на тому, що найбільше резонує з аудиторією.
4. Визначення трендів: метрики можуть допомогти виявляти нові тренди або теми, які стають популярними серед аудиторії.
5. Співставлення з конкурентами: збір метрик також може допомогти компаніям порівняти свою ефективність на Твіттері з конкурентами.

2.6 Аналіз сентіменту тексту

Аналіз сентіменту тексту, тобто його тональності, що розпочався ще у 1990-х роках, значно розвинувся за останні декілька десятиліть, ставши ключовим інструментом у багатьох областях науки. Цей напрям досліджень спрямований на вивчення людських думок, емоцій, цінностей та вражень відносно різних об'єктів, таких як продукти, послуги, організації, проблемні питання, події тощо.

Основною метою аналізу тональності є ідентифікація та класифікація емоційних відгуків у текстах, що виражають громадську думку. Тональність може бути класифікована як позитивна або негативна (бінарна класифікація), а також може включати більш деталізовані категорії, що відображають спектр емоцій від сильно негативних до сильно позитивних.

Аналіз тональності має широке застосування, від допомоги споживачам у виборі товарів і послуг до допомоги компаніям у зборі та аналізі відкритих даних замість проведення опитувань. Підходи до аналізу тональності включають:

- рівень аспектів (рівень ознак): дозволяє отримувати думки щодо конкретних характеристик об'єктів оцінки;
- рівень речення: тут оцінюється, чи є речення суб'єктивним або об'єктивним;
- рівень документа: визначає загальну тональність усього тексту або документа.

Інструменти для аналізу тональності можуть бути засновані на лексичному підході або використовувати методи машинного навчання. Лексичний підхід базується на використанні списків слів із визначеними емоційними значеннями, тоді як підходи на основі машинного навчання включають навчання моделей на прикладах текстів із відомою тональністю для прогнозування емоційного забарвлення нових текстів.

2.6.1 Огляд підходів

Метод аналізу тональності, що базується на лексиконі, є одним із найпростіших та найбільш інтуїтивно зрозумілих підходів у аналізі настроїв. Цей метод використовує лексикон суб'єктивних вражень, зосереджуючись на впливі певних слів або фраз. Емоційний відтінок визначається через присутність термінів у спеціальному словнику, який включає слова з позитивним, негативним або нейтральним емоційним забарвленням.

2.6.2 Словниковий метод

Словниковий метод розширює підхід, що базується на лексиконі, використовуючи більш складні бази даних із суб'єктивними судженнями. Він передбачає створення та постійне оновлення масиву емоційно забарвлених слів, використовуючи синоніми та антоніми для розширення бази. Такі бази даних, як WordNet, HowNet, SentiWordNet, SenticNet, та MPQA, є прикладами використання цього методу.

Ху та Лю використовували словниковий метод для аналізу клієнтських рецензій, ідентифікуючи ключові емоційні теги, що впливають на загальне сприйняття товару. Їхній підхід досягнув вражаючої точності у 85%.

Хові та Кім досліджували загальну полярність документів, аналізуючи емоційне забарвлення слів у контексті цілих речень. Вони створили системи, які враховували взаємозв'язки між синонімами та антонімами, а також відносну силу позитивного та негативного забарвлення слів.

Новітній підхід, описаний у праці [11], включає використання трьох окремих словників синонімів, що дозволяє більш глибоко аналізувати пости соціальних мереж. Однак, цей метод має певні складнощі з використанням часових ресурсів та адаптацією до специфічної тематики.

Кожен з цих підходів має свої переваги та недоліки, але вони всі спрямовані на розуміння та аналіз емоційного контенту тексту, що є важливим для різних застосувань, від маркетингового аналізу до дослідження соціальних мереж.

2.6.3 Корпусний метод

Корпусний метод у тональному аналізі заснований на використанні специфічних даних або текстів, які представляють певний домен чи контекст. Як пояснюється Бінгом Лю у своїй роботі [9], цей метод використовується у двох основних сценаріях: для створення нового лексикону, специфічного для конкретного домену, та для визначення емоційної полярності слів в залежності від контексту.

Hazivassiloglou та McKeown [8] розробили метод, що дозволяє визначити семантичну напрямленість сполучених прикметників у корпусі. Цей підхід включає аналіз контексту та використання логістичної регресії для визначення полярності. Наприклад, якщо прикметники сполучені словом "і", то вони зазвичай мають однакову полярність.

Дослідження Ding [12] сконцентроване на важливості контексту для визначення полярності відгуків та критики. Виявлено, що деякі прикметники можуть мати різну полярність залежно від контексту. В дослідженні використовувалися спеціально анотовані ідіоми для визначення настроїв, а також лінгвістичні правила для аналізу слів та словосполучень, поєднаних логічним зв'язком "але".

Особливість корпусного методу полягає в здатності адаптуватися до специфіки конкретного домену чи контексту, що робить його особливо корисним для аналізу даних із високою ступенем специфічності. Однак, це також означає, що для кожного нового домену чи контексту потрібно розробляти власний корпус і лексикон, що може бути ресурсозатратним.

2.6.4 Машинне навчання в тональному аналізі

Методи машинного навчання відкривають значні можливості для тонального аналізу, дозволяючи автоматизувати процес виявлення емоційної полярності текстів. Вони засновані на використанні алгоритмів, які можуть навчатися розпізнавати патерни і взаємозв'язки в даних, а потім застосовувати це розуміння для визначення емоційного забарвлення нових текстів.

Популярними методами машинного навчання для тонального аналізу є:

1. Навчання з підкріпленням: алгоритми, що використовують підхід до навчання з підкріпленням, де модель "нагороджується" за правильні прогнози та "штрафується" за помилки, що дозволяє їй самостійно удосконалювати свої прогнози.
2. Нейронні мережі: глибокі нейронні мережі, особливо з використанням механізмів уваги та рекурентних нейронних мереж (RNN), є ефективними в аналізі секвенційних даних, таких як текст.
3. SVM (Support Vector Machines): алгоритми, що здатні класифікувати текст на основі об'єктів, представлених у високовимірному просторі.
4. Випадковий ліс (Random Forest): енсемблеві методи, що використовують багато дерев рішень для підвищення точності та уникнення перенавчання.

2.6.5 Моделі GPT у тональному аналізі

Моделі GPT (Generative Pre-trained Transformer) відкрили нові горизонти у тональному аналізі, завдяки своїй здатності генерувати, розуміти та витягувати значення з тексту. Ці моделі використовують архітектуру трансформера, яка дозволяє ефективно обробляти великі обсяги текстових даних, враховуючи контекстуальні взаємозв'язки між словами.

Застосування GPT в тональному аналізі охоплює такі складники.

1. **Генерацію тексту**: створення текстів з певною емоційною направленістю.
2. **Контекстуальне розуміння**: визначення емоційного тону тексту, враховуючи не лише окремі слова, а й загальний контекст фрази або абзацу.
3. **Класифікація тексту**: віднесення тексту до категорій залежно від його емоційного забарвлення.

2.6.6 Інші методи

Крім традиційних методів машинного навчання та передових технік, які використовуються в моделях GPT, існують інші підходи, такі як:

Аналіз частоти слів: використання статистичних методів для визначення найбільш часто вживаних слов, які мають емоційне забарвлення.

Аналіз мережевих структур: дослідження зв'язків між словами у тексті для виявлення загального емоційного забарвлення.

Усі ці підходи можуть бути застосовані для розробки систем автоматичного тонального аналізу, кожен з яких має свої переваги та обмеження в залежності від конкретних завдань і доменів застосування.

Висновки до розділу 2

У цьому розділі були розглянуті кілька методів прогнозування, які можуть бути застосовані для аналізу часових рядів, включаючи фінансові дані, які характеризуються високою волатильністю, на прикладі біткоїну. Статистичні методи, основані на авторегресії та інтегрованості, хоча й ефективні у деяких сценаріях, можуть не відображати всіх нюансів волатильності біткоїну.

Значною увагою у цьому розділі заслужив метод прогнозування за допомогою рекурентних нейронних мереж із довгою короткостроковою пам'яттю

(LSTM). Цей підхід ефективно враховує мінливість часових рядів, як у випадку з біткоїном. Проте, необхідно зазначити, що LSTM вимагають значних обчислювальних ресурсів та часу на навчання і схильні до перенавчання. Архітектура та властивості LSTM роблять їх придатними для прогнозування часових рядів, і саме цей метод було обрано як основний для реалізації у дослідженні.

Також було оцінено застосування різних методів семантичного аналізу, включаючи корпусний, словниковий, лексиконний та машинне навчання. В контексті тонального аналізу твітів, буде використовуватися підхід машинного навчання з LSTM та бібліотекою NLTK, що дозволяє здійснювати ефективний семантичний аналіз. LSTM підходить для вивчення тональності текстів, зокрема твітів, завдяки своїй здатності обробляти секвенційні дані та вловлювати зв'язки у великих масивах тексту.

Враховуючи, розвиток технологій у галузі обробки природної мови, в цій роботі особлива увага приділяється розвитку та впровадженню LSTM моделей для аналізу тональності текстових даних. Цей метод обрано через його здатність точно визначати емоційні нюанси та забезпечувати глибокий аналіз контекстуальної інформації

РОЗДІЛ 3 АНАЛІЗ СЕЗОННОСТІ ТА КОРРЕЛЯЦІЙ ДАНИХ З ТВІТЕРА ТА БІТКОІНУ

Прогнозування криптовалюти є складним завданням, оскільки її часові ряди є високо нелінійними і часто не піддаються традиційним статистичним методам прогнозування. В цьому контексті, методи машинного і глибокого навчання демонструють значні переваги, зокрема завдяки їх здатності враховувати складність і фрактальну природу даних.

Особливої уваги заслуговує вплив настрою криптотрейдерів та вплив твітів на ринкові настрої. Тональний аналіз цих твітів та визначення емоційного стану ринку в даний момент можуть стати важливими показниками, які покращують точність прогнозування.

У цьому розділі буде розглянуто використання методу глибокого навчання з використанням LSTM (Long Short-Term Memory) мереж та бібліотеки обробки природної мови NLTK для аналізу тональності новин в Твіттері. LSTM є оптимальним вибором для таких задач, оскільки вони здатні ефективно обробляти та аналізувати секвенційні дані, враховуючи довгострокові залежності, що є критично важливим для розуміння контексту та настроїв у тексті.

Застосування LSTM у поєднанні з NLTK дозволить ефективно аналізувати емоційне забарвлення твітів, а також ідентифікувати ключові тенденції та настрої, які можуть вплинути на рухи криптовалютного ринку. Це поєднання технологій дозволить створити більш точну та ефективну систему прогнозування для криптовалютних бірж.

3.1 Вхідні данні

В цьому розділі усі дослідження були проведенні за допомогою мови програмування Python. Дані та графіки були візуалізовані в середовищі розробки Jupyter Notebook.

Jupyter Notebook дозволяє дуже зручно візуалізувати отримані результати у процесі розробки програмного продукту. Виконувати можна окремі ділянки коду у будь-якому порядку, тому для задачі прогнозування криптовалюти це дуже зручна середа розробки.

Дані про курс біткоіна були взяти з відкритого ресурсу finance.yahoo.com.

Розглядаються данні за кожний день у проміжку від 2014.09.17-2023.10.23. Ми маємо 6 стовпчиків, що відповідають параметрам: Date, Open, High, Low, Close, Volume; та 30146 строчок, в яких передається інформація за кожний день у вказаному періоді, про ці параметри (рис 3.1).

Робити прогноз будемо за параметром Close. Він означає ціну біткоіна на момент закриття торгів.

	Date	Time	Open	High	Low	Close
0	2022.01.03	00:02	1.13674	1.13674	1.13674	1.13674
1	2022.01.03	00:03	1.13663	1.13674	1.13663	1.13674
2	2022.01.03	00:04	1.13663	1.13687	1.13663	1.13687
3	2022.01.03	00:05	1.13674	1.13687	1.13674	1.13674
4	2022.01.03	00:06	1.13684	1.13684	1.13644	1.13644

Рисунок 3.1 – Вхідні дані

Трансформуємо колонку Date до зручного виду та побудуємо графік ціни біткоіну за параметром Close у вказаний період (рис 3.2).

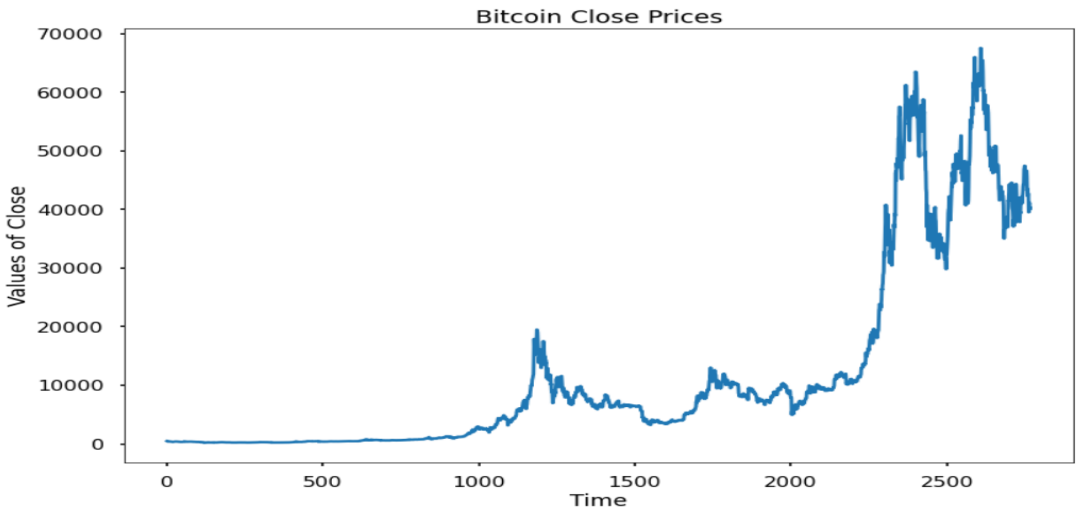

Рисунок 3.2 – Графік ціни біткоіну за параметром Close у проміжку від 2014.09.17-2023.10.23

Крім цього, додатково досліджуються дані з Twitter, що містять інформацію про твіти, пов'язані з біткоїном, починаючи з 2018 року.

	id	account	date	retweet_count	reply_count	quote_count	like_count
0	947631673021104129	austen	2018-01-01	4985	0	0	0
1	947634214421143552	austen	2018-01-01	110	0	0	0
2	947635762849128449	austen	2018-01-01	56	13	5	345
3	947673105287151617	austen	2018-01-01	18	2	10	175
4	947694155593068545	martypartymusic	2018-01-01	1	0	0	1

Рисунок 3.3 – Дані з твітеру

Дані по твітах містять такі колонки:

id – унікальний ідентифікатор твіта,

account – назва облікового запису, який зробив твіт,

date – дата публікації твіта,

retweet_count – кількість ретвітів,

reply_count – кількість відповідей на твіт,

quote_count – кількість цитувань твіта,

like_count – кількість лайків.

Зауваження: хоча API Twitter надає дані про кількість переглядів (impressions) для твітів, ця інформація не була включена в наш набір даних, оскільки такі дані з'явилися тільки в листопаді 2022 року. Отже, в якості основного показника використовується кількість ретвітів.

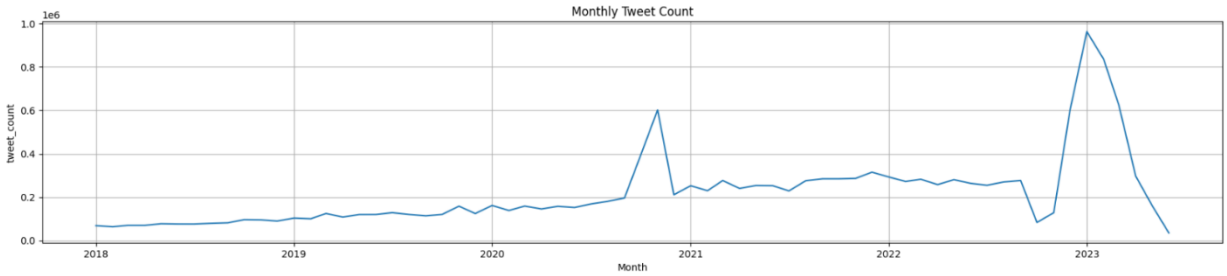

Рисунок 3.4 – Графік зміни кількості твітів

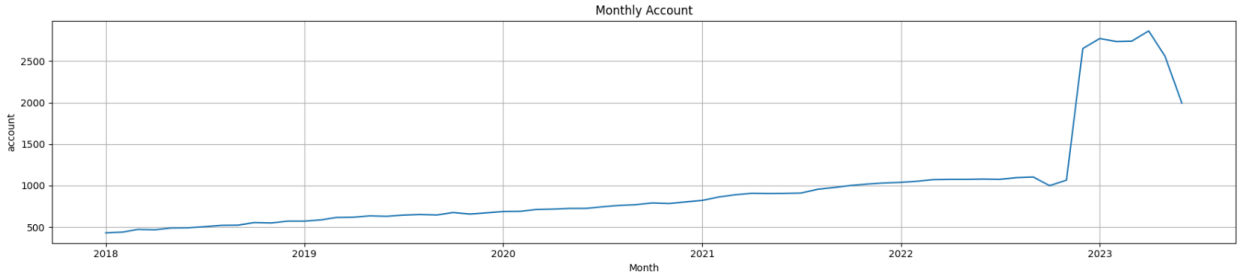

Рисунок 3.5 – Графік зміни кількості акантів

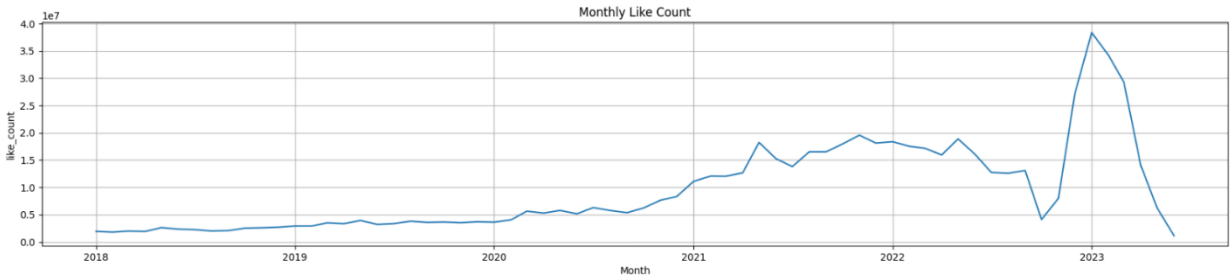

Рисунок 3.6 – Графік зміни кількості лайків у твітах

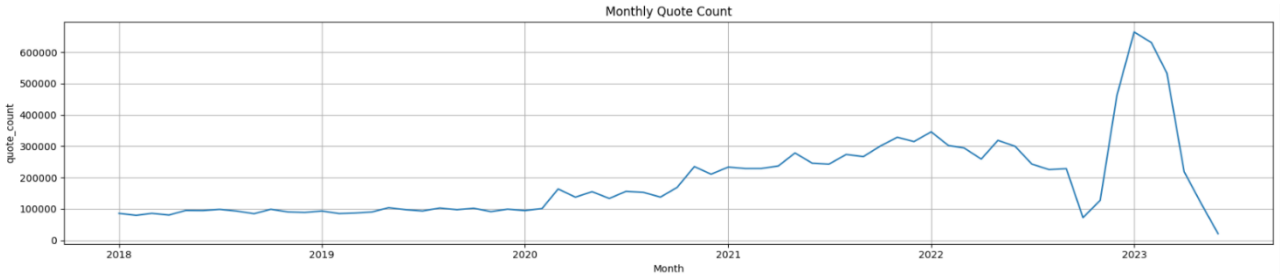

Рисунок 3.7 – Графік зміни кількості цитат у твітах

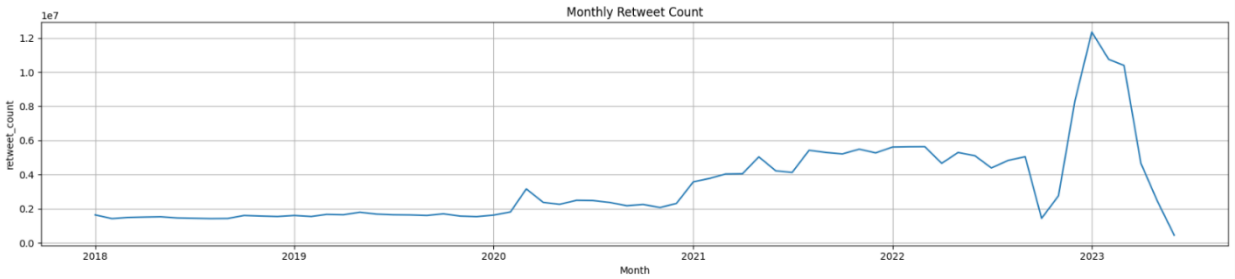

Рисунок 3.8 – Графіки зміни кількості ревітів у твітах

3.2 Обробка та візуалізація даних

У першу чергу, дані по курсу біткоїна було обмежено початком 2018 року, щоб мати консистентний набір даних для аналізу. Далі, була проведена візуалізація обох датасетів та обчислено відсоткові зміни ціни біткоїна з дня на день.

Виявлено численні викиди в даних з Twitter, що свідчило про необхідність провести нормалізацію. Дані були нормалізовані на добову кількість твітів, і 1% максимальних значень було відсіяно для забезпечення консистентності. Після цього було здійснено повторну візуалізацію.

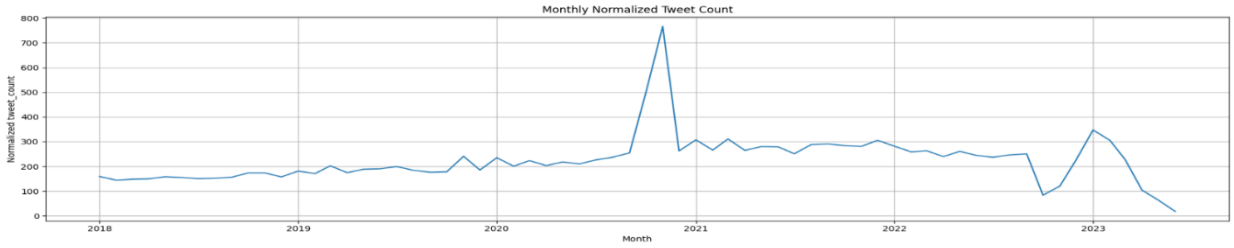

Рисунок 3.9 – Нормалізований графік кількості твітів

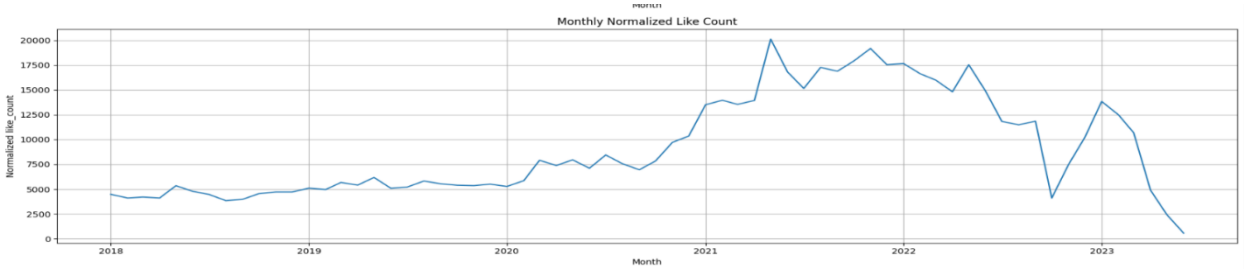

Рисунок 3.10 – Нормалізований графік кількост лайків у твітах

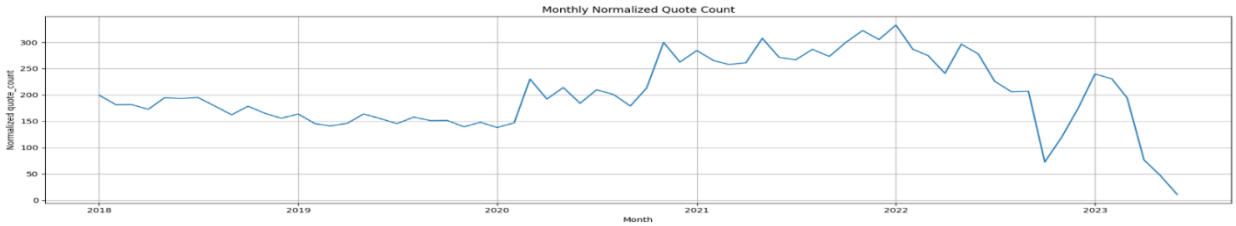

Рисунок 3.11 – Нормалізований графік кількості цитат у твітах

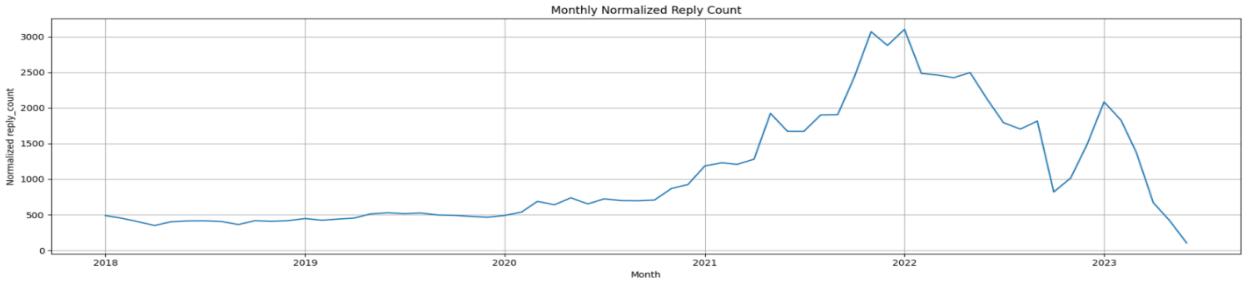

Рисунок 3.12 – Нормалізований графік кількості відповідей у твітах

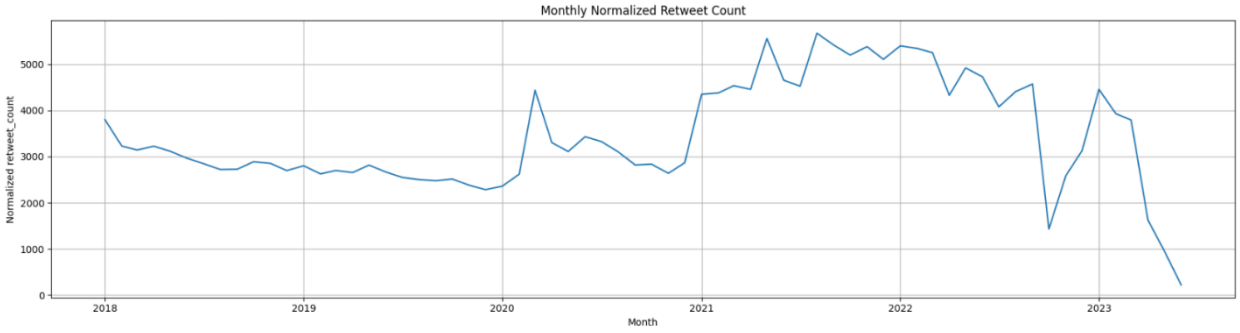

Рисунок 3.13 – Нормалізований графік кількості ревітів у твітах

3.3 Аналіз сезонності

Щоб визначити можливу сезонність у кількості твітів, було обраховано середні значення для кожного місяця на протязі років. Ці дані були візуалізовані разом із середнім трендом ціни біткоїна.

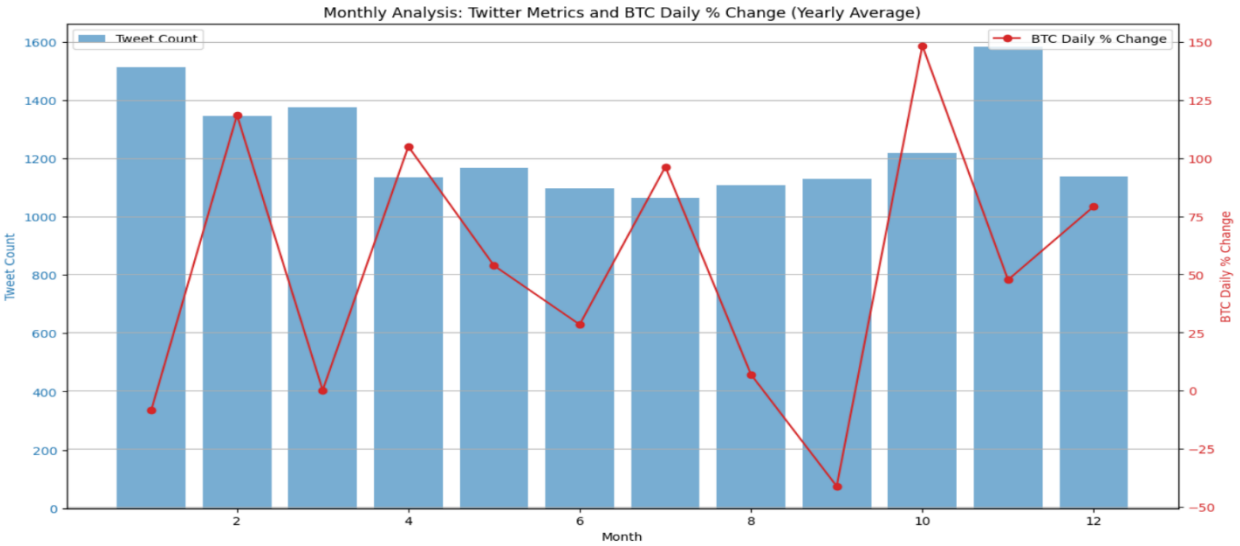

Рисунок 3.14 – Графік середніх значень кількості твітів та ціни Біткоіна по місячно

Результати вказують на наявність явної сезонності у кількості твітів. Кількість твітів стабільно знижується влітку та збільшується в листопаді та зимою. Ці зміни корелюють із змінами в ціні біткоїна, яка також має тенденцію до зниження влітку та збільшення в листопаді.

При аналізі метрик твітів можна помітити подібні шаблони, що підтверджують наявність сезонності в данних.

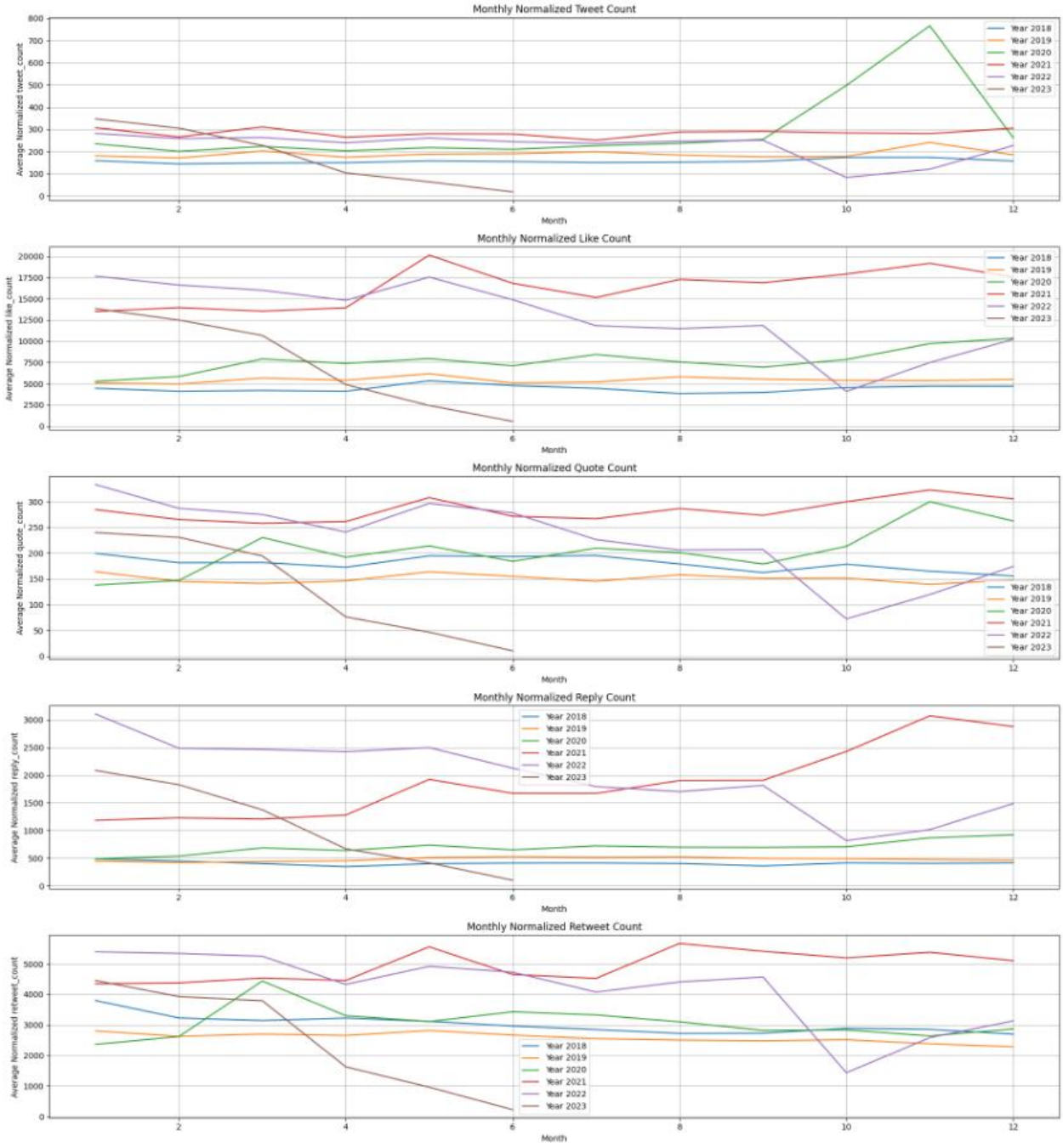

Рисунок 3.15 – Графік середніх значень всіх метрик по твітам по місячно

3.4 Математичне підтвердження сезонності

Для математичного підтвердження наявності сезонності було вирішено використовувати два методи:

1. Модель SARIMA — це популярна модель для прогнозування часових рядів з сезонною компонентою. SARIMA розширює модель ARIMA, додаючи сезонні компоненти.

2. Модель регресії з використанням dummy змінних — цей підхід передбачає створення dummy (або індикаторних) змінних для кожного місяця та включення їх в регресійну модель разом із іншими пояснювальними змінними. Таким чином, ми можемо перевірити статистичну значимість кожного місяця щодо цілі прогнозу.

Обидві моделі підтвердили наявність сезонності у даних та її важливість для прогнозування.

Давайте розглянемо результати дослідження сезонності нормованої кількості твітів.

1. SARIMA

Розглянемо результати для моделі SARIMA:

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4880	0.289	1.686	0.092	-0.079	1.055
ma.L1	-0.8562	0.255	-3.357	0.001	-1.356	-0.356
ma.S.L12	-1.0000	0.171	-5.849	0.000	-1.335	-0.665
sigma2	8792.7275	1.94e-05	4.52e+08	0.000	8792.727	8792.728

Рисунок 3.16 – Результати дослідження на сезонність за допомогою моделі SARIMA

ar.L1 (лагова авторегресійна компонента 1-го порядку): коефіцієнт складає 0.4880, що свідчить про позитивний авторегресійний зв'язок між поточним

значенням і попереднім. Значення P>|z| (0.092) трохи вище порогового значення 0.05, тому ця компонента може бути не такою важливою для моделі.

ma.L1 (лагова ковзаюча середня 1-го порядку): коефіцієнт -0.8562 свідчить про сильний негативний зв'язок. Значення P>|z| (0.001) є меншим за 0.05, тому ця компонента є статистично значущою.

ma.S.L12 (сезонна ковзаюча середня з лагом 12): коефіцієнт -1.0000 показує, що є сильний негативний сезонний вплив на ряд. Значення P>|z| (0.000) свідчить про те, що ця компонента є важливою для моделі.

sigma2: це оцінка дисперсії помилок. Велике значення показує велику варіабельність у ряду даних.

2. Регресійна модель з даммі-перемінними

OLS Regression Results

Dep. Variable:	normalized_tweet_count	R-squared (uncentered):	0.763
Model:	OLS	Adj. R-squared (uncentered):	0.715
Method:	Least Squares	F-statistic:	16.06
Date:	Tue, 10 Oct 2023	Prob (F-statistic):	1.83e-13
Time:	11:29:53	Log-Likelihood:	-409.51
No. Observations:	66	AIC:	841.0
Df Residuals:	55	BIC:	865.1
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
month_2	224.2022	53.573	4.185	0.000	116.839	331.565
month_3	229.2359	53.573	4.279	0.000	121.873	336.599
month_4	189.1840	53.573	3.531	0.001	81.821	296.547
month_5	194.7346	53.573	3.635	0.001	87.371	302.098
month_6	182.6733	53.573	3.410	0.001	75.310	290.036
month_7	212.7368	58.687	3.625	0.001	95.126	330.347
month_8	221.5510	58.687	3.775	0.000	103.941	339.161
month_9	225.6805	58.687	3.846	0.000	108.070	343.291
month_10	243.4078	58.687	4.148	0.000	125.797	361.018
month_11	316.4679	58.687	5.393	0.000	198.857	434.078
month_12	227.4367	58.687	3.875	0.000	109.826	345.047

Omnibus:	22.798	Durbin-Watson:	1.221
Prob(Omnibus):	0.000	Jarque-Bera (JB):	34.849
Skew:	1.303	Prob(JB):	2.71e-08
Kurtosis:	5.426	Cond. No.	1.10

Рисунок 3.17 – Результати дослідження на сезонність за допомогою регресійної моделі з даммі-перемінними

Для кожного місяця створена даммі-змінна. Коефіцієнт для кожної даммі-перемінної показує відхилення середньої кількості твітів у цьому місяці від січня (оскільки січень використовується як базовий місяць).

Наприклад, **month_2** має коефіцієнт 224.2022, що свідчить про те, що у лютому кількість твітів, у середньому, на 224.2022 більше, ніж у січні. При цьому P>|t| є меншим за 0.05, що підтверджує статистичну значущість цього зв'язку.

Всі даммі-перемінні мають значення P>|t| менше за 0.05, що свідчить про їх статистичну значущість. Особливо виділяється **month_11** з коефіцієнтом 316.4679, що свідчить про найбільше збільшення кількості твітів у листопаді порівняно з січнем.

Результати дослідження підтверджують наявність сезонності у нормованій кількості твітів. SARIMA показує важливість сезонної ковзаючої середньої компоненти. Регресійна модель із даммі-перемінними вказує на те, що майже кожен місяць має значуще відхилення від базового (січня). Ці знахідки можуть бути корисними для розробки стратегій маркетингу та прогнозування поведінки ринку.

3.5 Аналіз кореляційних зв'язків між метриками твітів і ціною біткоїна

У контексті аналізу впливу соціальних мереж на рух цін на криптовалютних біржах, було вирішено зосередитись на корреляційних зв'язках метрик по твітам щодо біткоїна та його ціною на ринку. З урахуванням потенційної великої волатильності біткоїна, для нормалізації даних ми конвертували денні зміни ціни в відсотках відносно попереднього дня. Додатково, для глибшого аналізу було розраховано кумулятивну сумму та добуток цих відсоткових змін.

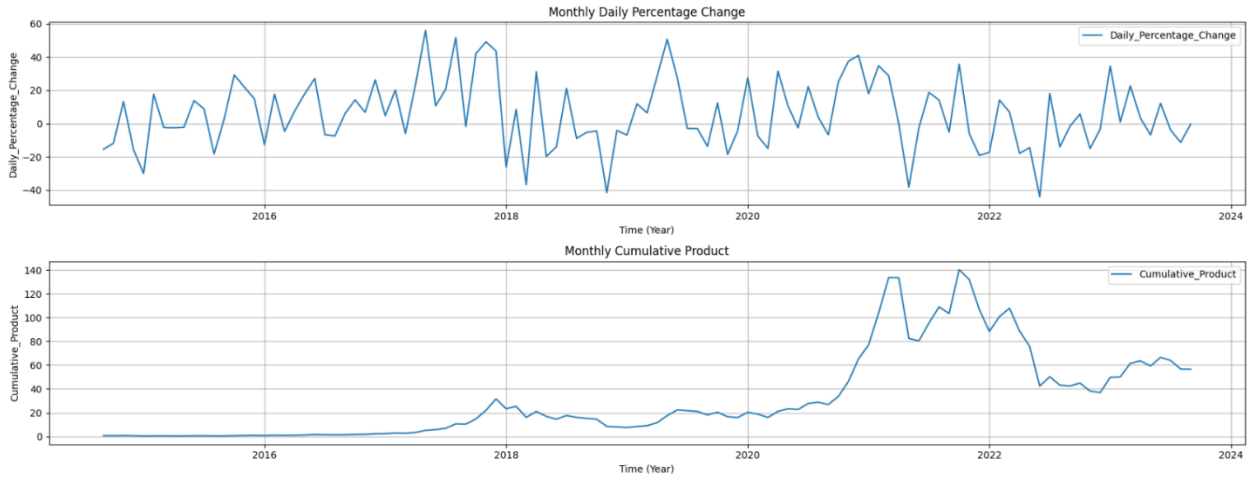

Рисунок 3.18 – Графіки процентної зміни ціни біткоіна та її кумулятивного добутку

Для аналізу кореляцій ми використовуємо таблицю, в якій показано взаємозв'язок між різними метриками твітів і двома основними індикаторами ціни біткоїна: кумулятивним добутком та кумулятивною сумою.

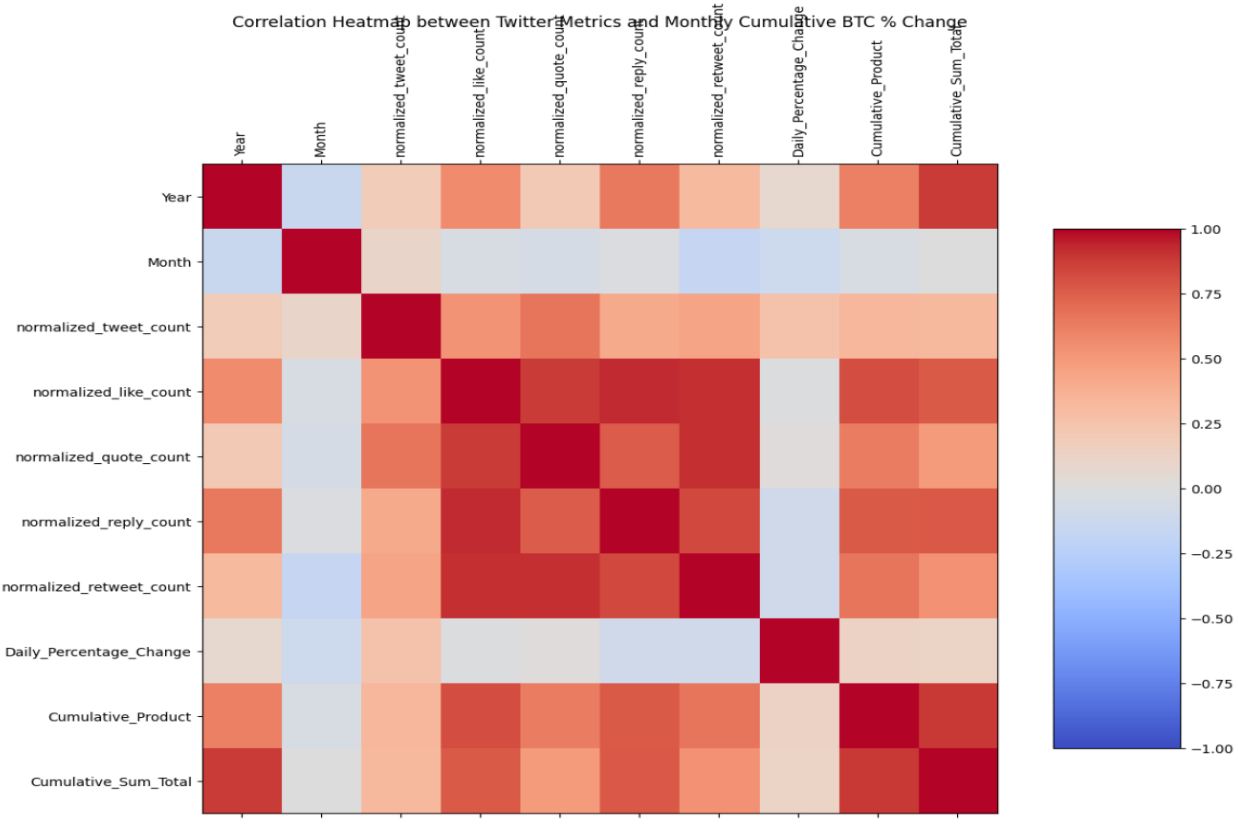

Рисунок 3.19 – Корреляційна теплова мапа

Попередній аналіз кореляційних зв'язків між метриками твітів і ціною біткоїна дозволив нам глибше зрозуміти, як саме інформаційні потоки у соціальних мережах впливають на ринкові коливання. В ході цього аналізу ми звернули увагу на те, що деякі метрики твітів чітко корелюють із змінами ціни біткоїна. Це виявлення відкриває додаткові можливості для прогнозування ринкової поведінки на основі даних з соціальних мереж.

3.6 Маркування вхідних даних

Для забезпечення максимальної точності та релевантності даних, на початковому етапі ми вручну пролейблували акаунти у Twitter, виділяючи тип акаунта. Категорії, на які ми звертали увагу, включали:

- **influencer**: впливові особистості у сфері криптовалют;
- **founder**: засновники проектів або компаній у криптовалютному просторі;
- **analysis/development**: акаунти, що займаються аналізом ринку або розробкою криптовалютних технологій;
- **irrelevant**: акаунти, не пов'язані з криптовалютами;
- **NFT**: акаунти, що фокусуються на невзаємозамінних токенах та суміжних технологіях;
- **news**: новинні акаунти, що публікують інформацію про криптовалюти;
- **wallet**: акаунти, пов'язані з криптовалютними гаманцями;
- **other**: інші акаунти, які не підпадають під вищезгадані категорії;
- **trader**: трейдери або інвестори у криптовалютному просторі;
- **foundation**: фонди та організації, пов'язані з криптовалютами;
- **validator**: акаунти, що беруть участь у валідації транзакцій або мережевих операціях;
- **scam**: акаунти з ознаками шахрайства або маніпуляцій.

	username	chain	label
1	jack	BTC	influencer
2	billbarhydt	BTC	wallet
3	TO	BTC	influencer
4	howardlindzon	BTC	influencer
5	nvk	BTC	influencer
6	woonomic	BTC	influencer
7	muneeb	BTC	founder
8	robustus	BTC	founder
9	nlw	BTC	influencer
10	udiWertheimer	BTC	influencer
11	stevefink	BTC	influencer
12	liron	BTC	influencer
13	mhluongo	BTC	influencer
14	JBSDC	BTC	other
15	scottmelker	BTC	trader
16	evoskuil	BTC	influencer
17	saifedean	BTC	influencer
18	brucefenton	BTC	founder
19	YahooFinance	BTC	news
20	novogratz	BTC	founder
21	joevezz	BTC	founder
22	maxkeiser	BTC	influencer
23	alistairmilne	BTC	influencer
24	jimmysong	BTC	influencer
25	CharlieShrem	BTC	founder
26	Mandrik	BTC	influencer
27	lopp	BTC	founder
28	level39	BTC	influencer

Рисунок 3.20 – Приклад розмітки акантів твітеру

Це дозволило нам збагатити наші дані контекстуальною інформацією, що значно підвищує якість та точність нашого подальшого аналізу та прогнозування.

3.7 Логіка скорингу зібраних акаунтів у Twitter

В рамках нашого дослідження ми розробили методику оцінки акаунтів у Twitter, з метою визначення їх впливовості та надійності. Ця методика базується на комплексному підході до аналізу різних характеристик акаунтів. Нижче представлено детальний опис логіки цієї системи скорингу.

1. **Збір і попередня обробка даних.**

Дані для кожного акаунта містять інформацію про верифікацію, кількість підписників, кількість підписок, кількість твітів та дату створення акаунта.

2. **Розрахунок індивідуальних оцінок.**

Оцінки визначаються за кожною характеристикою: верифікація акаунта, кількість підписників та підписок, кількість твітів, вік акаунта. Ці оцінки нормалізуються для уніфікації масштабу.

3. **Встановлення вагів для критеріїв.**

Верифікація (verified): Вага – 5%. Підтвердження офіційності аккаунту є важливим показником його автентичності, але не завжди корелює з впливовістю та якістю контенту.

Кількість підписників (followers_count): Вага – 20%. Цей показник відіграє ключову роль у визначенні впливовості аккаунту, адже велика кількість підписників часто свідчить про високу зацікавленість аудиторії. Співвідношення кількості підписників до кількості підписок (follow_rate): Вага – 10%. Цей показник дозволяє оцінити баланс між аудиторією аккаунту та його власними інтересами.

Кількість твітів (tweet_count): Вага – 10%. Активність аккаунту у мережі, вимірювана кількістю публікацій, вказує на його залученість та регулярність взаємодії з підписниками.

Вік акаунту (created_at): Вага – 10%. Старіші аккаунти можуть бути сприйняті як більш надійні, оскільки вони мали більше часу для розвитку та здобуття довіри.

Оцінка за міткою (label_score): Вага – 35%. Цей показник включає специфічні характеристики аккаунту, такі як тематика контенту та інші особливості, що не враховуються в стандартних критеріях.

Оцінка за ланцюгом (chain_score): Вага – 10%. Враховується роль аккаунту в певних мережевих структурах або групах, вказуючи на його позицію та вплив у певному контексті.

4. **Розрахунок загального скору.**

Фінальний скор для кожного акаунта розраховується як зважена сума індивідуальних оцінок. Це дозволяє отримати комплексну оцінку, яка враховує як кількісні, так і квалітативні аспекти аккаунтів.

В результаті було отримано такий датасет:

username	score
CathyYu220314	0.2630717394853572
RoamDAO	0.23138959817065113
NeerajKAfood	0.23006755799696466
Nitego29	0.22998104994534493
GabsCrypto	0.23137404570654913
Artnaz_eth	0.23009147107169783
mwaddip	0.258830226356273
timmy_sangwoo	0.17867516684863702
DefiDiogenesYFI	0.2286349607424699
Wooster_Han92	0.17914442410383818
Mothalamp	0.22948340083577795
SKYGODZ_ANIME	0.22835093604337708
rithvikra0	0.22798135059053615
Trrstnn	0.2585608394890038
CardanoNewsNow	0.258188811536683
VeronikaSimms	0.22739308595560367
GendoCrypto	0.22722407553803084
thealchememist	0.2556189528655433
FR3UD_	0.22561124788830186
iamjunpark	0.17576798952423472
0xzuberg	0.22567840258789393
thehasheddude	0.22676834209031813
SamWilsn	0.22456178588187584
DrLiesenfelt	0.2236025733760765
SaintSat0shi	0.22351942518193313

Рисунок 3.21 – Скорінг акантів твітеру

Ця методологія скорингу є важливою для нашого дослідження, оскільки вона дозволяє ідентифікувати та оцінити найбільш впливові та надійні акаунти, які мають потенціал впливати на ринкові показники криптовалют, зокрема

біткоїна. Наступним кроком у нашому дослідженні буде використання цих даних для побудови моделі прогнозування ціни біткоїна на основі активності в Twitter.

3.8 Аналіз сентименту твітів у контексті ринку біткоїна

Важливою частиною нашого дослідження є аналіз емоційного забарвлення, або сентименту, твітів, що стосуються біткоїна. Цей аналіз допомагає оцінити загальний настрій у соціальних мережах та його потенційний вплив на ринкові коливання.

1. **Використання бібліотеки NLTK:**
   - для аналізу сентименту ми використовуємо бібліотеку Natural Language Toolkit (NLTK), яка є стандартним інструментом для обробки природної мови у Python. Вона дозволяє нам визначати емоційну полярність тексту.
2. **Обчислення оцінок полярності:**
   - кожен твіт аналізується на предмет емоційного змісту. Ми визначаємо загальну полярність кожного повідомлення, що варіюється від негативного до позитивного.
3. **Структурування даних:**
   - після обчислення оцінок полярності, дані структуруються для подальшого аналізу. Ми фокусуємося на ідентифікаторі твіта, його тексті, а також розрахованій оцінці полярності.

Цей етап дозволяє нам оцінити, як публічні реакції та настрої, виражені в твітах, можуть впливати на динаміку цін на біткоїн. Розуміння цих взаємозв'язків є ключовим для створення ефективних моделей прогнозування ринкових тенденцій.

3.9 Етап прогнозування: використання LSTM моделі для аналізу історичних даних ціни біткоїна

На першому етапі нашого дослідження ми вирішили зосередитися на використанні моделі Long Short-Term Memory (LSTM) для прогнозування ціни біткоїна. LSTM є варіацією рекурентних нейронних мереж (RNN), які ефективно використовуються для аналізу часових рядів та прогнозування. Вони здатні вловлювати довгострокові залежності у даних, що є ключовим для розуміння динаміки ринку криптовалют.

На цьому етапі наш основний акцент - дослідити, наскільки ефективно LSTM може прогнозувати ціну біткоїна, використовуючи виключно історичні дані цін. Ми зосереджуємося на параметрі "Close", який відображає ціну закриття біткоїна за кожен день. Цей підхід дозволить нам встановити базову лінію для точності прогнозування, перш ніж інтегрувати дані з соціальних мереж.

Процес моделювання включає декілька ключових кроків:

1. Збір та підготовка даних: вибірка історичних даних цін біткоїна, очищення та нормалізація даних для забезпечення консистентності ввідних даних для моделі.
2. Створення LSTM моделі: розробка архітектури LSTM мережі, яка включає налаштування параметрів, таких як кількість шарів, нейронів у кожному шарі, та функції активації.
3. Тренування та валідація моделі: навчання моделі на тренувальному наборі даних і оцінка її ефективності на валідаційному наборі.
4. Тестування та аналіз прогнозів: оцінка якості прогнозів моделі на тестовому наборі даних, аналіз помилок прогнозування та зіставлення результатів із фактичними цінами.

Цей підхід дозволить нам зрозуміти потенціал LSTM у прогнозуванні цін на біткоїн, що стане фундаментом для подальшого включення даних з Twitter у нашу прогнозну модель.

3.10 Реалізація та результати LSTM моделі для прогнозування ціни біткоїна

У цій частині дослідження ми застосували модель Long Short-Term Memory (LSTM) для прогнозування ціни біткоїна. Наша мета полягала у виявленні ефективності LSTM у прогнозуванні цінових показників, використовуючи історичні дані. Нижче представлено опис процесу реалізації моделі та аналіз її результатів.

Реалізація LSTM моделі:

1. Підготовка даних:
   - ми завантажили історичні дані цін біткоїна та пов'язані дані з Twitter, після чого об'єднали ці набори даних;
   - вибрані основні признаки для аналізу: Open, High, Low, Volume, а цільовим показником була ціна закриття (Close);
   - дані були нормалізовані для підготовки до подачі в LSTM мережу.
2. Створення та навчання моделі:
   - модель LSTM складалася з послідовних шарів LSTM і Dense, з включенням шарів Dropout для запобігання перенавчанню;
   - модель була скомпільована з оптимізатором adam та функцією втрат mean_squared_error;
   - проведено тренування моделі на 100 епохах з розміром пакету 32.
3. Прогнозування та аналіз результатів:
   - було здійснено прогнозування цін та порівняння їх із реальними цінами.
   - результати показали:
     середньоквадратична помилка (MSE): 517801.33;
     середня абсолютна помилка (MAE): 514.41;
     Корінь середньоквадратичної помилки (RMSE): 719.58.
4. Візуалізація:
   - було побудовано графіки, які відображають реальні та прогнозовані ціни біткоїна, а також динаміку помилок на етапах обучення та валідації.

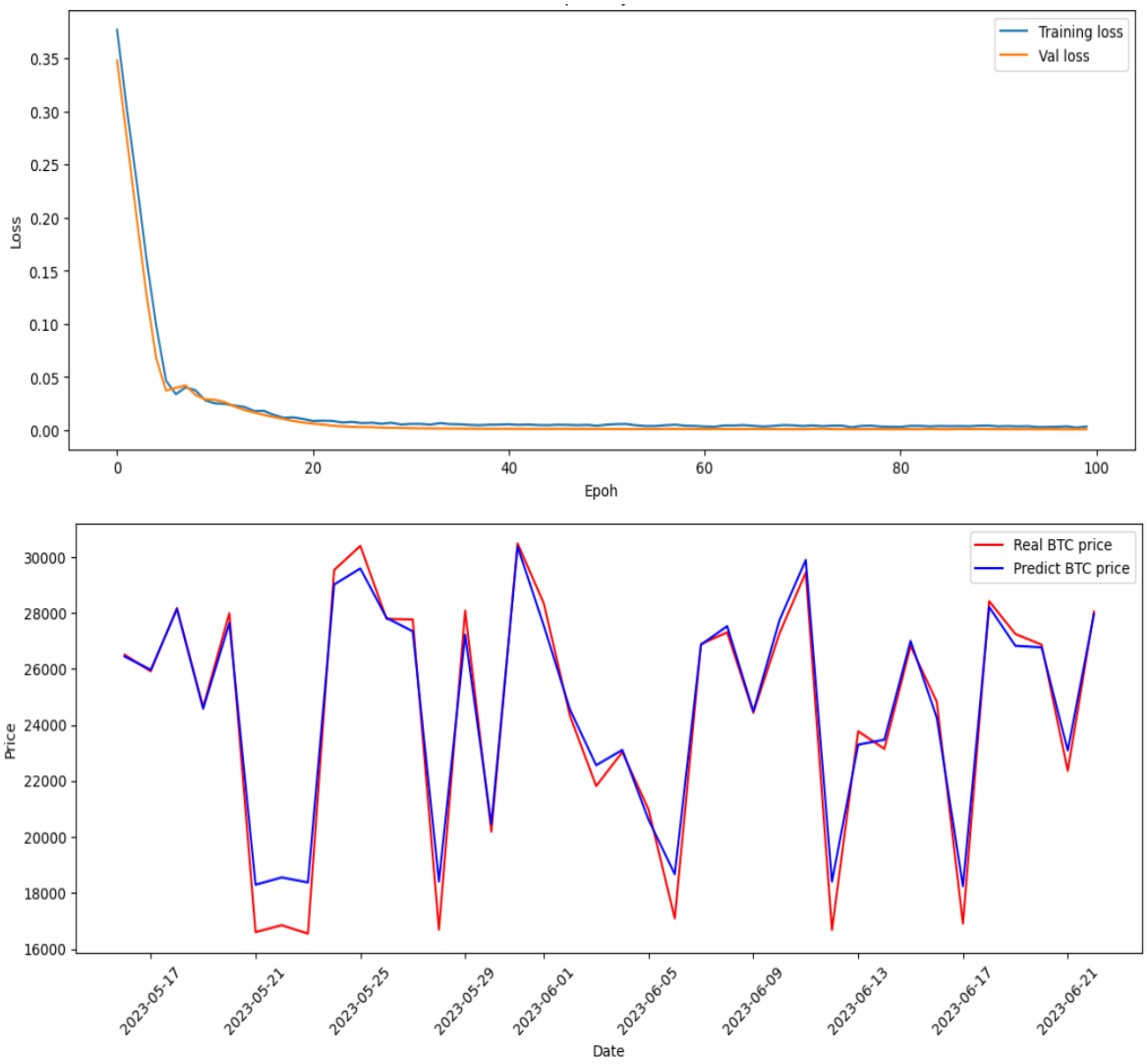

Рисунок 3.22 – Результати прогнозування моделі навченої на історичних даних

Висновки з аналізу:

- модель LSTM показала помірну точність у прогнозуванні цін біткоїна, що підтверджується помилками MSE та MAE;
- величина помилок, особливо RMSE, вказує на певні труднощі при прогнозуванні високо волатильних ринкових умов, характерних для криптовалют;
- незважаючи на ці виклики, LSTM здатна вловлювати складні шаблони у часових рядах, що робить її перспективною для подальших експериментів із залученням додаткових даних, зокрема з соціальних мереж.

3.11 Розрахунок середньозважених показників та групування твітів за датою

Для детальнішого аналізу взаємозв'язку між активністю у Twitter та ринковими показниками біткоїна, ми провели розрахунок середньозважених значень ключових метрик твітів, групуючи їх за датами. Цей підхід дозволив нам врахувати вплив кожного твіта в контексті його значущості та авторитетності акаунта, який його опублікував.

1. Обрахунок важливості твітів:
    - ми використовували показники, такі як кількість ретвітів, вражень і лайків, разом із розрахованими оцінками сентименту. Кожен з цих показників множився на вагу (скоринг) відповідного акаунта, що дозволило нам врахувати авторитетність та впливовість акаунта.
2. Середньозважені показники (для кожної дати ми розрахували середньозважене значення для наступних показників):
    - кількість ретвітів;
    - кількість вражень;
    - кількість лайків;
    - оцінки полярності.

Ці значення були обчислені шляхом сумування добутків кожного показника на вагу акаунта, поділеного на суму всіх ваг акаунтів, які здійснили публікації цього дня.

3. Групування за датами:
    - дані були агреговані за кожною датою, що дало можливість зробити висновки про загальний настрій та активність у соціальних мережах протягом певного періоду.

Цей підхід забезпечує глибше розуміння того, як окремі твіти та загальна активність у Twitter впливають на ринкову динаміку біткоїна. Агрегування даних за датами дозволяє спостерігати тренди та взаємозв'язки у часі, що є ключовим для розробки точних прогнозних моделей.

3.12 Інтеграція даних із Twitter для покращення прогнозування ціни біткоїна

Після аналізу базової моделі LSTM, яка використовувала виключно історичні дані цін біткоїна, ми вирішили розширити наш підхід, інтегруючи дані з Twitter. Мета цього етапу полягає у вивченні впливу соціальних сигналів на точність прогнозування цін на криптовалюту.

Реалізація розширеної LSTM моделі:

1. Об'єднання даних: ми здійснили інтеграцію історичних цінових даних з даними Twitter, включаючи такі показники, як середня кількість ретвітів, вражень, вподобань, а також полярність твітів.
2. Підготовка та нормалізація даних: вибрані признаки були нормалізовані для підвищення ефективності навчання моделі.
3. Тренування моделі: модель LSTM була адаптована для роботи з додатковими фічами з Twitter. Модель включала кілька шарів LSTM та Dense, а також шари Dropout.
4. Прогнозування та аналіз:було проведено прогнозування з використанням розширеної моделі та зіставлення цих прогнозів з реальними цінами.

Результати показали значне зменшення помилок порівняно з базовою моделлю:

- Середньоквадратична помилка (MSE): 67296.80;
- Середня абсолютна помилка (MAE): 204.97;
- Корінь середньоквадратичної помилки (RMSE): 259.42.

5. Візуалізація результатів: графіки показують порівняння між реальними та прогнозованими цінами, а також динаміку помилок протягом етапів навчання та валідації.

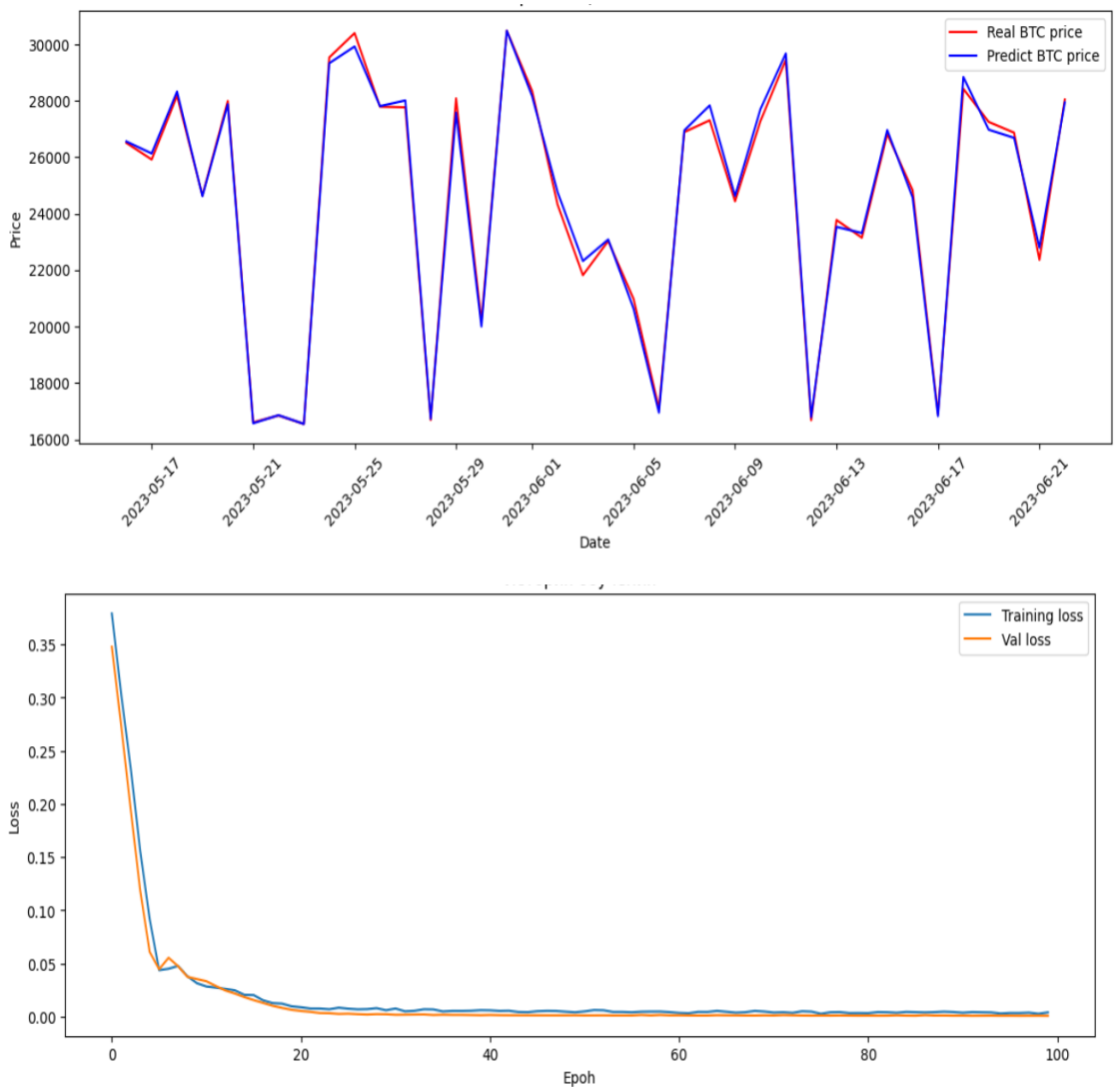

Рисунок 3.23 – Результати прогнозування моделі навченої на історичних даних та даних з твітеру

Отже: інтеграція даних із Twitter здійснила помітний вплив на точність прогнозів, що підтверджується зниженням показників помилок.

Ці результати свідчать про значний потенціал використання соціальних медіа як додаткового джерела інформації для прогнозування ринкових тенденцій у криптовалютному просторі.

3.13 Порівняльний аналіз двох моделей LSTM для прогнозування ціни біткоїна

У цьому розділі ми зосереджуємося на порівнянні двох моделей LSTM, які були розроблені та протестовані у попередніх частинах нашого дослідження. Перша модель базувалася виключно на історичних цінових даних біткоїна, тоді як друга – інтегрувала дані з Twitter. Основна мета порівняння – виявити вплив соціальних медіа на точність прогнозування.

Основні висновки з порівняння:

1. Точність прогнозування:
- перша модель показала середньоквадратичну помилку (MSE) на рівні 517801.33, середню абсолютну помилку (MAE) 514.41, та корінь середньоквадратичної помилки (RMSE) 719.58;
- друга модель, з даними з Twitter, покращила результати: MSE знизилася до 67296.80, MAE - до 204.97, RMSE - до 259.42;
- це свідчить про значно кращу ефективність другої моделі у прогнозуванні цін біткоїна.
2. Аналіз впливу даних з Twitter:
- інтеграція даних з Twitter, включаючи інформацію про ретвіти, вподобання, враження та полярність твітів, виявилася значущою для точності прогнозів;
- соціальні сигнали з Twitter, схоже, надають важливу контекстуальну інформацію, яка допомагає моделі краще розуміти ринкову поведінку та реакції.
3. Візуальний аналіз:
- графіки прогнозованих та реальних цін підтверджують вищевказані висновки. Вони демонструють, що прогнози другої моделі ближчі до фактичних цін порівняно з прогнозами першої моделі.
4. Імплікації для подальшого дослідження:
- результати показують, що врахування даних з соціальних мереж може бути важливим у контексті прогнозування ринків криптовалют;

- це відкриває шлях для подальших досліджень із більш детальним аналізом різних видів соціальних сигналів та їх взаємодії з іншими економічними індикаторами.

Загальний висновок: це порівняння підкреслює значення інтеграції різноманітних джерел даних у прогнозуванні фінансових ринків. Включення соціальної інформації з Twitter значно покращило якість прогнозів цін на біткоїн, підтверджуючи, що сучасні методи аналізу даних можуть забезпечити більш глибоке розуміння ринкових тенденцій.

5. Кореляція з кумулятивним добутком:

- найвища кореляція: здається, що найбільшу кореляцію з кумулятивним добутком мають normalized_like_count (0.818323), normalized_reply_count (0.765775), та normalized_retweet_count (0.660323). Це може вказувати на те, що активність користувачів у вигляді вподобань, відповідей та ретвітів має більший зв'язок з ціною біткоїна, ніж просто кількість твітів.

Висновки до розділу 3

Сильні взаємозв'язки: сильні кореляції між метриками активності в соціальних мережах (особливо like, reply, та retweet) і кумулятивним добутком ціни біткоїна можуть свідчити про те, що підвищена соціальна активність корелює зі збільшенням вартості біткоїна.

Рік як індикатор: позитивна кореляція з роком може відображати довгостроковий тренд зростання ціни біткоїна впродовж часу.

Місяць менш значущий: низька та негативна кореляція з місяцем може вказувати на те, що короткострокові сезонні фактори менш важливі для змін в ціні біткоїна.

Обмеження та додаткові аспекти:

Не лінійний зв'язок: варто врахувати, що кореляція не обов'язково означає причинно-наслідковий зв'язок. Окрім того, деякі взаємозв'язки можуть бути не лінійними.

Зовнішні фактори: взаємозв'язки можуть також бути впливнуті зовнішніми факторами, які не включені в цей аналіз, наприклад, глобальні економічні події, політична нестабільність, або зміни в регуляціях криптовалют.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

На сьогоднішній день все більше стартапів перетворюють ідеї пов’язані із нейромережевими технологіями у працюючі бізес моделі. Інтерес інветорів до стартапів, що працюють із нейронними мережами зростає з кожним роком. Все більше і більше з’являється прикладів успішного застосування штучного інтелекту у найрізноманітніших галузях діяльності людини, значно зростає кількість корпорацій що впроваджують системи прогнозування із застосуванням нейромереж у свою операційну діяльність.

Окрім цього, нейронні мережі знайшли широке застосування в прогнозуванні фінансових ринків. Нова і перспектина технологія швидко привернула увагу венчурних інвесторів. У цьому розділі розглянута розробка стартап проекту системи прогнозування фінансових ринків.

4.1 Опис ідеї проекту

Результативність застосування традиційних методів прогнозування фінансових активів, наприклад акцій, облігацій, та валюти, які вільно продаються і купуються на біржах, можна назвати недостатньою для потреб сучасного ринку. Це пов’язано із тим, що інсвестиції на фондовому ринку тісно пов’язані із Інтернетом і залежні від інформаційного середовища. Для підвищення точності прогнозування доцільно застосувати таку модель, що не тільки базується на кореляціях факторів та особливостях часового ряду, а й тісно пов’язана з декількома джерелами даних.

Сучасні системи прогнозування не враховують комплексно кількісні та якісні фактори, що впливають на зміну курсів акцій або враховують у векторному вигляді, який обмежує можливості представлення вхідних даних до нейромережі. Такі системи прогнозування зазвичай мають точність 53-58% вірного прогнозу.

Цієї точності замало для того, щоб використовувати такі системи як повноцінний інструмент для економічного аналізу та прогнозування. Отже, ідеєю старттаппроекту є створення і розповсюдження системи прогнозування фінансових ринків із більшою ніж у конкурентів точністю прогнозування тренду (табл. 4.1).

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1. Фінансова аналітика	Дешевша, порівняно з аналогами, краща точність прогнозування тренду
	2. Торгівля на біржах	Можливість отримувати безпосередній прибуток за рахунок точності прогнозування

Застосування зрозумілої математичної моделі спрощує реалізацію, а доведення кращих показників прогнозування може сприяти розповсюдженню системи. Безумовно, на ринку є аналоги. Для порівняння розробленої системи проведемо аналіз потенційних техніко-економічних переваг ідеї.

Метою аналізу техніко-економіних переваг є чітке виокремлення технічних і маркетингових особливостей розробленого продукту:

- дослідження характеристик і властивостей розробленої системи;
- дослідження конкурентів, товарів-аналогів, товарів-замінників та загальної ситуації на ринку де буде комерціалізуватимс стартам;
- проведення порівняльного аналізу слабких, нейтральних та сильних характеристик розробленої системи;

Визначимо сильні, слабкі та нейтральних характеристик ідеї проекту (табл.4.2).

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W(слабка сторона)	N(нейтральна сторона)	S(сильна сторона)
		Мій проект	Neural Builder 2015	Neural Shell	Neural Trader			+
1.	Вартість ПЗ	Низька	Висока	Висока	Висока			+
2.	Доступність	Низька	Висока	Висока	Середня		+	
3.	Кроссплатформленість	Так	Так	Ні	Ні			+
4.	Підтримка	+	-	+	+		+	

З таблиці можна зрбити висновок, що розроблена система є конкурентоспроможною.

4.2 Технологічний аудит проекту

Метою технічного аудиту є визначення переліку технологій, за допомогою який реалізована система і їх аналіз. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл.4.3):

- за допомогою якої технологією розролена система згідно ідеї проекту;
- чи існують у відкритому доступі ці технології, чи їх потрібно додатково

розробляти або купувати;

- чи має доступ розробник до описаних технологій.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Прогнозування цін акцій	Рекурентність нейронної мережі. Визначення курсу акцій	Висока	Висока
Обраною мовою програмування є JavaScript, використовуються нейронні мережі їх доігою короткостроковою пам`яттю				

За результатами аналізу можна зробити висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано мову програмування JavaScript через її доступність та безкоштовність

4.3 Аналіз ринкових можливостей запуску стартап-проекту.

Метою аналізу ринкових можливостей є виявлення та дослідження обмежень, можливостей та розрахунок конкретних показників реалізації розробленої системи прогнозування фінансових ринків як стартап-проекту.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап проекту

№п/п	Показники стану ринку(найменування)	Характеристика
1	Кількість систем конкурентів на ринку, одиниць	4
2	Загальний обсяг продаж, грн/м. од.	234000
3	Динаміка ринку	Стагнує
4	Наявність обмеженьдля входу (вказати характер обмежень)	Нема
5	Специфічні вимоги до стантартизації та сертифікації	Нема
6	Середнє значення рентабельності в галузі(або ринку)%	18%(відповідає середній річний ставці депозиту в гривні)

Проаналізувавши результати, можна зробити висновок що проект є придатним до інвестицій, оскільки його рентабельність перевищує відсоток депозиту.

За результатами порівняння що було наведене у табл. 4.4 було зроблено висновок, що ринок є придатним для розповсюдження системи як стартап-проекту.

Після цього були досліджені потенційні категорії клієнтів, їх особливості та затверджено перелік вимог до системи для кожної категорії клієнтів (табл. 4.5)

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проект

№п/п	Потреба, що формує ринок систем прогнозування	Цільова аудиторія та потенційні клієнти	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачі до товару
1	Програмне забезпечення для прогнозування фінансових ринків	Компанії, що займаються фінансовою аналітикою, трейдери, керівництво публічних корпорацій	Цільові групи клієнтів мають різні доходи і потенційний прибуток від застосування системи, різні очікування та мотивація	Зручний інтерфейс, наявність інструкції з користування, кросплатформенність

Після дослідження ринкового середовища: складено таблиці факторів, що сприяють реалізації розробленої системи як стартап-проекту, та факторів, що йому заважають (табл. 4.6, 4.7).

Таблиця 4.6 – Фактори загроз

№п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок систем з кращими характеристиками та моделю надання послуг	Вийти на ринок зосередивши увагу на власні переваги системи. Покращення якісних характеристик системи прогнозування. Обрати цільову аудиторію
2	Зміна потреб користувачів	Клієнту потрібна буде система з більшою точністю прогнозування новими функціями	Передбачити можливість розширення системи та підвищення точності прогнозування

Таблиця 4.7 – Фактори можливостей

№п/п	Фактор	Зміст можливості	Можлива реація компанії
1	Конкуренція	Відсутність аналогів на українському ринкудля вітчизняного користувача. Нові методи моделювання більш легкі в освоєні праці	Адаптація системи до особливостей потреб на українському ринку . Розширення можливостей, максимальне спрощення
2	Поява альтернативних методів моделювання		

Надалі було проведено дослідження пропозиції: визначили загальні характеристик конкуренції на ринку (табл. 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому виражена дана характеристика	Вплив на діяльність компанії (можливі дії для підвищення конкурентноспроможності)
1. Вказати тип конкуренції - монополія	На ринку присутні декілька постачальників-конкурентів, але їх товар дещо відрізняється від нашої системи	Підтримка якості системи, безперервний розвиток, покращення, вдосконалення, оновлення та підтримка

Закінчення таблиці 4.8

2. За рівнем конкурентної боротьби– міжнародний	Компанії-конкуренти з інших країн	Створення основної системи та прогнозування, щоб легко локалізувати її для використання в інших країнах
3. За галузевою ознакою – внутрішньогалузева	Система може застосовуватися в одній галузі, але в різних її сферах	Постійне вдосконалення системи, прогнозування, що немає прив`язки до сфери, але має до галузі
4. Конкуренція видами товарів:- товарно -видова	Конкуренція між видами систем прогнозування їх особливостями	Створити систему прогнозування, враховуючи, недоліки конкурентів
5. За характером конкурентних переваг: -цінова	Покращення процесу створення програмного продукту, мінімізація витрат на оновлення застосування безперервної антеграції	Використання відкритих та дешевих технологій для побудови системи в порівнянні з системами конкурентів, але тільки якщо ці технології відповідають необхідним критеріям якості
6. За інтенсивністю: -не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах, виставках

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл.4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік безпосередніх конкурентів	Визначити бар`єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили клієнтів	Фактори загроз з боку товарів-замінників
	Neural Builder	Наявність вже існуючих рішень	-	Якість системи та її підтримки оновлення	Більш відомий розробник, що підтримує свою систему
Висновки	На даний момент немає конкурентів на українському ринку	Вихід на український ринок буде легшим через відсутність конкуренції	-	Вимоги клієнтів такі, як зручний інтерфейс, якість програмного продукту	Випустити систему, що буде не гірше, ніж у конкурента, але мати кращу точність прогнозування

За результатами порівняльного дослідження що наведене у табл. 4.9 було зроблено висновок про доцільність виходу на український та міжнародні ринки.
На основі аналізу ситуації на ринках, проведеного в табл. 4.9, а також із урахуванням характеристик розробленої системи (табл. 4.2), вимог потенційних клієнтів до товару (табл. 4.5) та особливостей ринкового середовища (табл. 4.6, 4.7) досліджується та визначається перелік факторів конкурентоспроможності розробленої системи прогнозування фінансових ринків. Аналіз факторів конкурентоспроможності розробленої системи наведено у табл. 4.9

За визначеними факторами конкурентоспроможності (табл. 4.9) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.10).

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг систем-конкурентів у порівнянні з розробленою системою прогнозування						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	15					*		
2	Кросплатформність	20			*				
3	Орієнтованість на кінцевого споживача	7					*		

Кінцевим кроком маркетингового дослідження можливостей реалізації системи прогнозування фінансових ринків як стартап-проекту є побудова SWOT-аналізу (матриці сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі описаних конкурентних та маркетингових загроз та можливостей, та сильних і слабких сторін (табл. 4.11). Список маркетингових загроз та можливостей було складено на основі дослідження факторів загроз та факторів можливостей ринкової ситуації. Маркетингові загрози та можливості є наслідками (прогнозованими результатами) впливу ринкових факторів.

Таблиця 4.11 – SWOT-аналіз стартап-проекту

Сильні сторони: Ціна, орієнтованість на кінцевого споживача, кросплатформеність	*Слабкі сторони:* Складність розповсюджувати продукцію закордоном
Можливості: Відсутність конкуренції на українському ринку	*Загрози:* Зміна основних потреб клієнтів, при відсутності конкуренції необхідно підтримувати інтерес аудиторії до продукту

За результатами SWOT-аналізу було сформовано альтернативи ринкової стратегії (перелік заходів) для виведення розробленої системи як стартап-проекту на український та міжнародні ринки та розрахований оптимальний час їх ринкової реалізації з урахуванням потенційних розробок конкурентів, що можуть бути виведені на ринок (див. табл. 4.9, аналіз потенційних конкурентів). Запропоновані альтернативи були проаналізовані з точки зору часу реалізації та ймовірності отримання ресурсів (табл. 4.12).

Таблиця 4.12 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Приблизна ймовірність отримання ресурсів	Приблизні строки реалізації
1	Безкоштовне розповсюдження версії створеного програмного продукту	85%	12 місяців
2	Створення програмної системи з подальшим платним розповсюдженням (продаж платної ліцензії)	45%	12 місяців

Після аналізу було обрано альтернативу №2.

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим етапом передбачає визначення стратегії охоплення ринку: дослідження цільових груп потенційних клієнтів, які приведенні в табл. 4.13.

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

№п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи(сегменту)	Інтенсивність конкуренції в сегменті	Простота іходу у сегмент
1	Компанії, що займаються фінансовою аналітикою	Готово	Необхідно	Висока	Середня

Визначена цільова група клієнтів: Компанії що займаються фінансовою аналітикою

Дослідивши потенційні групи клієнтів було визначено цільові категорію, для яких буде пропонуватися розроблена система, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу.

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (табл. 4.14).

Таблиця 4.14 — Визначення базової стратегії розвитку

№п/п	Обрана альтернатива розвитку стартап-проекту	Ключові конкурентноспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Визначити потреби сучасного ринку для кожної з груп	Цінова політика, універсальність продукту	Стратегія диференціаціїї

Наступним кроком обрано стратегію конкурентної поведінки (табл.4.15).

Таблиця 4.15 — Визначення базової стратегії конкурентної поведінки

№п/п	Чи є проект першою подібною системою на ринку	Чи буде розробник системи шукати нових споживачів, або забирати існуючих у конкурентів	Чи буде розробник копіювати властивості товару конкурентів	Стратегія конкурентної поведінки
1	Ні	Шукати нових	Ні	Заняття конкурентної ніші

За результатами аналізу вимог клієнтів визначених категорій до розробника стартап-проекту та до продукту (див. Табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.14) та стратегії конкурентної поведінки (табл. 4.6) розроблено стратегію позиціонування (табл. 4.16), що полягає у формуванні

ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.16 — Визначення стратегії позиціонування

№п/п	Вимоги до системи з боку потенційних клієнтів	Основна стратегія розвитку	Ключові конкурентно-спроможні позиції власного стартап-проекту	Вибір основних асоціацій
1	Простота побудова моделі прогнозування, довгий час навчання нейромережі, проте вищі точність прогнозу та інтуїтивно зрозумілий інтерфейс, докладне керівництво користувача	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів, відмінні особливості споживача	Економія часу, зручність застосування, практичність

За результатами дослідження була сформована система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії українському та міжнародному ринках.

4.5 Розроблення маркетингової програми стартап-проекту

Визначено маркетингову концепцію системи прогнозування фінансових ринків із використанням рекурентних нейромереж, яку буде купувати споживач. Основна концепція продукту — письмовий опис фізичних та програмних характеристик системи, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

За результатами дослідження було сформовано трирівневу маркетингову модель системи: ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання:

− 1-й рівень вирішує питання щодо того, засобом вирішення якої потреби і / або проблеми буде система, яка її основна вигода;

− 2-й рівень являє рішення того, як буде реалізована система в реальному ринковому середовищі. Рівень включає в себе якість, властивості, дизайн, упаковку, ціну;

− 3-й рівень визначає додаткові послуги та переваги для клієнта системи, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Після цього визначимо цінові межі, якими необхідно керуватись при встановленні ціни на систему, яке включає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової категорії клієнтів (табл. 4.17). Аналіз проведено експертним методом.

Таблиця 4.18 — Визначення меж встановлення ціни

№п/п	Приблизна вилка вартості товарів-замінників	Приблизний рівень доходів цільової групи клієнтів	Верхня та нижня межі вартості системи
1	800-3000$	3000$+	200-700$

Після цього визначимо оптимальну систему збуту, за допомогою якої буде розповсюджуватися система як сатрап-проект.(табл. 4.18).

Таблиця 4.19 — Формування системи збуту

№п/п	Особливості ринкової поведінки потенційних клієнтів	Функції збуту, які має виконувати постачальник системи	Глибина каналу збуту	Оптимальний канал збуту
1	Цільові клієнти компанії, які бажають впровадити у своїй роботі сучасні засоби, допоможуть прогнозувати фінансові ринки із кращою точністю	Побудова прямих контактів іх потенційними клієнтами і їх підтримка формування попиту і стимулювання продажів	Один (від виробника одразу споживачу)	Прямий канал збуту до клієнта, мінімізувати збутові витрати. Розвиток нових маркетингових концепцій

Фінальною складовою маркетингової програми є визначення концепції маркетингових комунікацій.

Результатом дослідження стала робоча ринкова програма, що включає в себе концепції системи, збуту, просування та попереднє дослідження ціноутворення на продукт, базується на цінностям та потребах категорій покупців, конкурентні переваги системи, стан та динаміку маркетингового середовища, в межах якого впроваджено системи прогнозування фінансових ринків як стартап-проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 4

В цьому розділі було проведено аналіз розробленої системи прогнозування фінансових ринків у якості стартап-проекту. Варто зауважити, що проект має можливість ринкової комерціалізації, через те, що ринок систем прогнозування потребує якісного та інноваційного продукту для прогнозування фінансові ринки.

Результатом роботи є розроблений стартап-проект, план виходу на ринки програмного забезпечення та маркетингова стратегія. Розроблений стартап-проект доцільно застосувати при комерціалізації розробки.

Висока конкуренція зумовлює необхідність виваженого підходу до просування продукту. Проте кількість учасників фінансових ринків зростає з кожним днем, що зумовлює хороші перспективи для комерціалізації системи. Для впровадження ринкової реалізації проекту була обрана стратегія, яка включає розробку програмної системи із класичною моделлю ліцензування за певну плату.

Можна сказати, що подальший розвиток проекту є доцільним, оскільки кількість потенційних користувачів системи зростає кожного року.

ВИСНОВКИ

У цій магістерській роботі проведено глибоке дослідження взаємодії між активністю користувачів соціальної мережі Twitter та ціною біткоїна. Метою було з'ясувати, чи існує статистично значущий кореляційний зв'язок між показниками соціальної активності та коливаннями ціни на криптовалюту. Для цього було аналізувано великий набір даних, що включав кількість твітів, лайків, цитувань, відповідей, ретвітів, пов'язаних з біткоїном, а також щоденні цінові зміни біткоїна.

Дослідження показало, що існує виразна кореляція між активністю в Twitter і ціновою динамікою біткоїна. Ця кореляція особливо виражена, коли аналізуємо кумулятивні показники, такі як кумулятивна сума та кумулятивний добуток цих параметрів. В ході аналізу розроблено чотири статистичні моделі для прогнозування ціни біткоїна, базуючись на зібраних даних з Twitter.

Крім цього, розробилено систему скорінгу твітів, яка класифікує їх за типами акаунтів та впливом на ціну біткоіну. Було проведено детальний аналіз тональності кожного твіта з використанням сучасних NLP моделей. Найбільш ефективною виявилася модель, що включає в себе історичні дані, сентиментальний аналіз твітів, усереднені метрики та скорінг твітів. Це підкреслює суттєвий вплив соціальних медіа на ціну біткоіна та значно покращує ефективність прогнозування ціни за допомогою ансамблю моделей, побудованого на основі історичних даних та даних з соціальних медіа.

Результати цієї роботи вже отримали визнання у науковому співтоваристві. Наша стаття 'Social Media Impact on the "Cosmos" Blockchain Ecosystem: State and Prospect', співавторами якої є Олійник Б.О. та Петренко А.В., була прийнята до публікації у міжнародному науковому журналі 'Data Science Journal', який індексується у Scopus. Це вказує на актуальність і значущість даного

дослідження, яке відкриває нові можливості для подальшого вивчення впливу соціальних медіа на фінансові ринки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Tschorsch, F., & Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralised Digital Currencies. IEEE Communications Surveys Tutorials, 18(3), pp.20842123.

2. Zhong, F., & Michahelles, F. (2015). Predicting Bitcoin Price Fluctuations Using Social Media Analysis. In Proceedings of the Workshop on Cryptocurrencies and Blockchains for Distributed Systems.

3. Barber, S., Boyen, X., Shi, E., & Uzun, E. (2012). Bitter to Better - How to Make Bitcoin a Better Currency. In Financial Cryptography and Data Security (pp. 399-414). Springer, Berlin, Heidelberg.

4. Tufekci, Z. (2017). Twitter and tear gas: The power and fragility of networked protest. Yale University Press.

5. Howard, P. N., & Hussain, M. M. (2013). Democracy's fourth wave?: Digital media and the Arab Spring. Oxford University Press.

6. Marwick, A., & Lewis, R. (2017). Media manipulation and disinformation online. Data & Society Research Institute.

7. Zohar, A., & Rosenschein, J. S. (2010). Mechanisms for Internet-based voting on a blockchain. In International Conference on Financial Cryptography and Data Security (pp. 34-49). Springer, Berlin, Heidelberg.

8. Kim, T. E., Hong, S., & Kang, P. (2020). Predicting cryptocurrency price bubbles using social media data and epidemic theory. Decision Support Systems, 129,pp. 113234.

9. Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. Communications of the ACM, 59(7),pp. 96-104.

10. Adhami, S., Giudici, G., & Martinazzi, S. (2018). Why do businesses go crypto? An empirical analysis of initial coin offerings. Journal of Economics and Business, 100, pp.64-75.

11. Larimer, D. (2013). Delegated proof-of-stake (DPOS). BitShares White Paper.

12. Woolley, S., & Howard, P. (2016). Political Communication, Computational Propaganda, and Autonomous Agents-Introduction. International Journal of Communication, 10, 9.

13. Tapscott, D., & Tapscott, A. (2016). Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world. Penguin.

14. Bruns, A. (2019). Are filter bubbles real? Polity Press.

15. G. Hinton, S. Nitish, A. Krizhevsky. Improving neural networks by preventing co-adaptation of feature detectors. 2012. arXiv:1207.0580v1. DOI: https://doi.org/10.48550/arXiv.1207.0580

16. Selvaraju R. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. International Journal of Computer Vision. 2020. 128. DOI: 10.1007/s11263-019-01228-7.

17. Simonyan K. Very Deep Convolutional Networks for Large-Scal Image Recognition. 2014. arXiv: 1409.1556v6. DOI: https://doi.org/10.48550/arXiv.1409.1556

18. Моделі нейронних мереж. URL: http://techn.sstu.ru/kafedri/подразделения/1/MetMat/Terin/neiro/neiro.htm (дата звернення: 28.11.2023).

19. Zhou B. Learning Deep Features for Discriminative Localization. 2015. arXiv:1512.04150v1. DOI: https://doi.org/10.48550/arXiv.1512.04150

ДОДАТОК А ПРОГРАМНИЙ КОД

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import torch.nn.functional as F
sns.set(color_codes=True)
df = pd.read_csv('../input/sentiment140/training.1600000.processed
.noemoticon.csv', encoding='latin-1', header = None).iloc[:10000, :]
df.columns=['Sentiment', 'id', 'Date', 'Query', 'User', 'Tweet']
df = df.drop(columns=['id', 'Date', 'Query', 'User'], axis=1)

df['Sentiment'] = df.Sentiment.replace(4,1)
df['tweet'] = df['tweet'].astype('str')

# nltk
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords from
nltk.tokenize import word_tokenize
#Stop Words: A stop word is a commonly used word (such as "the",
" a", "an", "in")
#that a search engine has been programmed to ignore,
#both when indexing entries for searching and when retrieving
them as the result of a search query.
nltk.download('stopwords')
stopword = set(stopwords.words('english'))
print(stopword)
import warnings
warnings.filterwarnings('ignore')
import re
import string
import pickle
urlPattern = r"((http://)[^ ]*|(https://)[^ ]*|( www\.)[^ ]*)"
userPattern = '@[^\s]+'
some = 'amp,today,tomorrow,going,girl'
def process_tweets(tweet):
  # Lower Casing
    tweet = re.sub(r"he's", "he is", tweet)
    tweet = re.sub(r"there's", "there is", tweet)
    tweet = re.sub(r"We're", "We are", tweet)
    tweet = re.sub(r"That's", "That is", tweet)
    tweet = re.sub(r"won't", "will not", tweet)
```

```
tweet = re.sub(r"they're", "they are", tweet)
tweet = re.sub(r"Can't", "Cannot", tweet)
tweet = re.sub(r"wasn't", "was not", tweet)
tweet = re.sub(r"don\x89Ûªt", "do not", tweet)
tweet = re.sub(r"aren't", "are not", tweet)
tweet = re.sub(r"isn't", "is not", tweet)
tweet = re.sub(r"What's", "What is", tweet)
tweet = re.sub(r"haven't", "have not", tweet)
tweet = re.sub(r"hasn't", "has not", tweet)
tweet = re.sub(r"There's", "There is", tweet)
tweet = re.sub(r"He's", "He is", tweet)
tweet = re.sub(r"It's", "It is", tweet)
tweet = re.sub(r"You're", "You are", tweet)
tweet = re.sub(r"I'M", "I am", tweet)
tweet = re.sub(r"shouldn't", "should not", tweet)
tweet = re.sub(r"wouldn't", "would not", tweet)
tweet = re.sub(r"i'm", "I am", tweet)
tweet = re.sub(r"I\x89Ûªm", "I am", tweet)
tweet = re.sub(r"I'm", "I am", tweet)
tweet = re.sub(r"Isn't", "is not", tweet)
tweet = re.sub(r"Here's", "Here is", tweet)
tweet = re.sub(r"you've", "you have", tweet)
tweet = re.sub(r"you\x89Ûªve", "you have", tweet)
tweet = re.sub(r"we're", "we are", tweet)
tweet = re.sub(r"what's", "what is", tweet)
tweet = re.sub(r"couldn't", "could not", tweet)
tweet = re.sub(r"we've", "we have", tweet)
tweet = re.sub(r"it\x89Ûªs", "it is", tweet)
tweet = re.sub(r"doesn\x89Ûªt", "does not", tweet)
tweet = re.sub(r"It\x89Ûªs", "It is", tweet)
tweet = re.sub(r"Here\x89Ûªs", "Here is", tweet)
tweet = re.sub(r"who's", "who is", tweet)
tweet = re.sub(r"I\x89Ûªve", "I have", tweet)
tweet = re.sub(r"y'all", "you all", tweet)
tweet = re.sub(r"can\x89Ûªt", "cannot", tweet)
tweet = re.sub(r"would've", "would have", tweet)
tweet = re.sub(r"it'll", "it will", tweet)
tweet = re.sub(r"we'll", "we will", tweet)
tweet = re.sub(r"wouldn\x89Ûªt", "would not", tweet)
tweet = re.sub(r"We've", "We have", tweet)
tweet = re.sub(r"he'll", "he will", tweet)
tweet = re.sub(r"Y'all", "You all", tweet)
tweet = re.sub(r"Weren't", "Were not", tweet)
tweet = re.sub(r"Didn't", "Did not", tweet)
tweet = re.sub(r"they'll", "they will", tweet)
tweet = re.sub(r"they'd", "they would", tweet)
tweet = re.sub(r"DON'T", "DO NOT", tweet)
tweet = re.sub(r"That\x89Ûªs", "That is", tweet)
tweet = re.sub(r"they've", "they have", tweet)
tweet = re.sub(r"i'd", "I would", tweet)
tweet = re.sub(r"should've", "should have", tweet)
tweet = re.sub(r"You\x89Ûªre", "You are", tweet)
```

```
tweet = re.sub(r"where's", "where is", tweet)
tweet = re.sub(r"Don\x89Ûªt", "Do not", tweet)
tweet = re.sub(r"we'd", "we would", tweet)
tweet = re.sub(r"i'll", "I will", tweet)
tweet = re.sub(r"weren't", "were not", tweet)
tweet = re.sub(r"They're", "They are", tweet)
tweet = re.sub(r"Can\x89Ûªt", "Cannot", tweet)
tweet = re.sub(r"you\x89Ûªll", "you will", tweet)
tweet = re.sub(r"I\x89Ûªd", "I would", tweet)
tweet = re.sub(r"let's", "let us", tweet)
tweet = re.sub(r"it's", "it is", tweet)
tweet = re.sub(r"can't", "cannot", tweet)
tweet = re.sub(r"don't", "do not", tweet)
tweet = re.sub(r"you're", "you are", tweet)
tweet = re.sub(r"i've", "I have", tweet)
tweet = re.sub(r"that's", "that is", tweet)
tweet = re.sub(r"i'll", "I will", tweet)
tweet = re.sub(r"doesn't", "does not", tweet)
tweet = re.sub(r"i'd", "I would", tweet)
tweet = re.sub(r"didn't", "did not", tweet)
tweet = re.sub(r"ain't", "am not", tweet)
tweet = re.sub(r"you'll", "you will", tweet)
tweet = re.sub(r"I've", "I have", tweet)
tweet = re.sub(r"Don't", "do not", tweet)
tweet = re.sub(r"I'll", "I will", tweet)
tweet = re.sub(r"I'd", "I would", tweet)
tweet = re.sub(r"Let's", "Let us", tweet)
tweet = re.sub(r"you'd", "You would", tweet)
tweet = re.sub(r"It's", "It is", tweet)
tweet = re.sub(r"Ain't", "am not", tweet)
tweet = re.sub(r"Haven't", "Have not", tweet)
tweet = re.sub(r"Could've", "Could have", tweet)
tweet = re.sub(r"youve", "you have", tweet)
tweet = re.sub(r"donå«t", "do not", tweet)

tweet = re.sub(r"some1", "someone", tweet)
tweet = re.sub(r"yrs", "years", tweet)
tweet = re.sub(r"hrs", "hours", tweet)
tweet = re.sub(r"2morow|2moro", "tomorrow", tweet)
tweet = re.sub(r"2day", "today", tweet)
tweet = re.sub(r"4got|4gotten", "forget", tweet)
tweet = re.sub(r"b-day|bday", "b-day", tweet)
tweet = re.sub(r"mother's", "mother", tweet)
tweet = re.sub(r"mom's", "mom", tweet)
tweet = re.sub(r"dad's", "dad", tweet)
tweet = re.sub(r"hahah|hahaha|hahahaha", "haha", tweet)
tweet = re.sub(r"lmao|lolz|rofl", "lol", tweet)
tweet = re.sub(r"thanx|thnx", "thanks", tweet)
tweet = re.sub(r"goood", "good", tweet)
tweet = re.sub(r"some1", "someone", tweet)
tweet = re.sub(r"some1", "someone", tweet)
tweet = tweet.lower()
```

```
    tweet=tweet[1:]
    # Removing all URls
    tweet = re.sub(urlPattern,'',tweet)
    # Removing all @username.
    tweet = re.sub(userPattern,'', tweet)
    #remove some words
    tweet= re.sub(some,'',tweet)
    #Remove punctuations
    tweet = tweet.translate(str.maketrans("","",string.punctuation
))
    #tokenizing words
    tokens = word_tokenize(tweet)
    #tokens = [w for w in tokens if
    len(w)>2] #Removing Stop Words
    final_tokens = [w for w in tokens if w not in
    stopword] #reducing a word to its word stem
    wordLemm = WordNetLemmatizer()
    finalwords=[]
    for w in final_tokens:
      if len(w)>1:
        word = wordLemm.lemmatize(w)
        finalwords.append(word)
    return ' '.join(finalwords)


abbreviations = {
    "$" : " dollar ",
    "€" : " euro ",
    "4ao" : "for adults only",
    "a.m" : "before midday",
    "a3" : "anytime anywhere anyplace",
    "aamof" : "as a matter of fact",
    "acct" : "account",
    "adih" : "another day in hell",
    "afaic" : "as far as i am concerned",
    "afaict" : "as far as i can tell",
    "afaik" : "as far as i know",
    "afair" : "as far as i remember",
    "afk" : "away from keyboard",
    "app" : "application",
    "approx" : "approximately",
    "apps" : "applications",
    "asap" : "as soon as possible",
    "asl" : "age, sex, location",
    "atk" : "at the keyboard",
    "ave." : "avenue",
    "aymm" : "are you my mother",
    "ayor" : "at your own risk",
    "b&b" : "bed and breakfast",
    "b+b" : "bed and breakfast",
    "b.c" : "before christ",
    "b2b" : "business to business",
```

```
"b2c" : "business to customer",
"b4" : "before",
"b4n" : "bye for now",
"b@u" : "back at you",
"bae" : "before anyone else",
"bak" : "back at keyboard",
"bbbg" : "bye bye be good",
"bbc" : "british broadcasting corporation",
"bbias" : "be back in a second",
"bbl" : "be back later",
"bbs" : "be back soon",
"be4" : "before",
"bfn" : "bye for now",
"blvd" : "boulevard",
"bout" : "about",
"brb" : "be right back",
"bros" : "brothers",
"brt" : "be right there",
"bsaaw" : "big smile and a wink",
"btw" : "by the way",
"bwl" : "bursting with laughter",
"c/o" : "care of",
"cet" : "central european time",
"cf" : "compare",
"cia" : "central intelligence agency",
"csl" : "can not stop laughing",
"cu" : "see you",
"cul8r" : "see you later",
"cv" : "curriculum vitae",
"cwot" : "complete waste of time",
"cya" : "see you",
"cyt" : "see you tomorrow",
"dae" : "does anyone else",
"dbmib" : "do not bother me i am busy",
"diy" : "do it yourself",
"dm" : "direct message",
"dwh" : "during work hours",
"e123" : "easy as one two three",
"eet" : "eastern european time",
"eg" : "example",
"embm" : "early morning business meeting",
"encl" : "enclosed",
"encl." : "enclosed",
"etc" : "and so on",
"faq" : "frequently asked questions",
"fawc" : "for anyone who cares",
"fb" : "facebook",
"fc" : "fingers crossed",
"fig" : "figure",
"fimh" : "forever in my heart",
"ft." : "feet",
"ft" : "featuring",
```

```
"ftl" : "for the loss",
"ftw" : "for the win",
"fwiw" : "for what it is worth",
"fyi" : "for your information",
"g9" : "genius",
"gahoy" : "get a hold of yourself",
"gal" : "get a life",
"gcse" : "general certificate of secondary education",
"gfn" : "gone for now",
"gg" : "good game",
"gl" : "good luck",
"glhf" : "good luck have fun",
"gmt" : "greenwich mean time",
"gmta" : "great minds think alike",
"gn" : "good night",
"g.o.a.t" : "greatest of all time",
"goat" : "greatest of all time",
"goi" : "get over it",
"gps" : "global positioning system",
"gr8" : "great",
"gratz" : "congratulations",
"gyal" : "girl",
"h&c" : "hot and cold",
"hp" : "horsepower",
"hr" : "hour",
"hrh" : "his royal highness",
"ht" : "height",
"ibrb" : "i will be right back",
"ic" : "i see",
"icq" : "i seek you",
"icymi" : "in case you missed it",
"idc" : "i do not care",
"idgadf" : "i do not give a damn fuck",
"idgaf" : "i do not give a fuck",
"idk" : "i do not know",
"ie" : "that is",
"i.e" : "that is",
"ifyp" : "i feel your pain",
"IG" : "instagram",
"iirc" : "if i remember correctly",
"ilu" : "i love you",
"ily" : "i love you",
"imho" : "in my humble opinion",
"imo" : "in my opinion",
"imu" : "i miss you",
"iow" : "in other words",
"irl" : "in real life",
"j4f" : "just for fun",
"jic" : "just in case",
"jk" : "just kidding",
"jsyk" : "just so you know",
"l8r" : "later",
```

```
"lb" : "pound",
"lbs" : "pounds",
"ldr" : "long distance relationship",
"lmao" : "laugh my ass off",
"lmfao" : "laugh my fucking ass off",
"lol" : "laughing out loud",
"ltd" : "limited",
"ltns" : "long time no see",
"m8" : "mate",
"mf" : "motherfucker",
"mfs" : "motherfuckers",
"mfw" : "my face when",
"mofo" : "motherfucker",
"mph" : "miles per hour",
"mr" : "mister",
"mrw" : "my reaction when",
"ms" : "miss",
"mte" : "my thoughts exactly",
"nagi" : "not a good idea",
"nbc" : "national broadcasting company",
"nbd" : "not big deal",
"nfs" : "not for sale",
"ngl" : "not going to lie",
"nhs" : "national health service",
"nrn" : "no reply necessary",
"nsfl" : "not safe for life",
"nsfw" : "not safe for work",
"nth" : "nice to have",
"nvr" : "never",
"nyc" : "new york city",
"oc" : "original content",
"og" : "original",
"ohp" : "overhead projector",
"oic" : "oh i see",
"omdb" : "over my dead body",
"omg" : "oh my god",
"omw" : "on my way",
"p.a" : "per annum",
"p.m" : "after midday",
"pm" : "prime minister",
"poc" : "people of color",
"pov" : "point of view",
"pp" : "pages",
"ppl" : "people",
"prw" : "parents are watching",
"ps" : "postscript",
"pt" : "point",
"ptb" : "please text back",
"pto" : "please turn over",
"qpsa" : "what happens",
"ratchet" : "rude",
"rbtl" : "read between the lines",
```

```
    "rlrt" : "real life retweet",
    "rofl" : "rolling on the floor laughing",
    "roflol" : "rolling on the floor laughing out loud",
    "rotflmao" : "rolling on the floor laughing my ass off",
    "rt" : "retweet",
    "ruok" : "are you ok",
    "sfw" : "safe for work",
     "sk8" : "skate",
    "smh" : "shake my head",
    "sq" : "square",
    "srsly" : "seriously",
    "ssdd" : "same stuff different day",
    "tbh" : "to be honest",
    "tbs" : "tablespooful",
    "tbsp" : "tablespooful",
    "tfw" : "that feeling when",
    "thks" : "thank you",
    "tho" : "though",
    "thx" : "thank you",
    "tia" : "thanks in advance",
    "til" : "today i learned",
    "tl;dr" : "too long i did not read",
    "tldr" : "too long i did not read",
    "tmb" : "tweet me back",
    "tntl" : "trying not to laugh",
    "ttyl" : "talk to you later",
    "u" : "you",
    "u2" : "you too",
    "u4e" : "yours for ever",
    "utc" : "coordinated universal time",
    "w/" : "with",
    "w/o" : "without",
    "w8" : "wait",
    "wassup" : "what is up",
    "wb" : "welcome back",
    "wtf" : "what the fuck",
    "wtg" : "way to go",
    "wtpa" : "where the party at",
    "wuf" : "where are you from",
    "wuzup" : "what is up",
    "wywh" : "wish you were here",
    "yd" : "yard",
    "ygtr" : "you got that right",
    "ynk" : "you never know",
    "zzz" : "sleeping bored and tired"
}

def convert_abbrev_in_text(tweet):
    t=[]
    words=tweet.split()
    t = [abbreviations[w.lower()] if w.lower() in

abbreviations.ke ys() else w for w in words]
```

```
    return ' '.join(t)


df['processed_tweets'] = df['tweet'].apply(lambda x:
process_tweet s(x))
df['processed_tweets'] = df['processed_tweets'].apply(lambda x:
co nvert_abbrev_in_text(x))
print('Text Preprocessing complete.')
df


#removing shortwords
df['processed_tweets']=f['processed_tweets'].apply(lambda x: "
".j oin([w for w in x.split() if len(w)>3]))


from sklearn.utils import shuffle
df =  df(tweets).reset_index(drop=True)


#tokenization
tokenized_tweet=df['processed_tweets'].apply(lambda x: x.split())

from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer token =
RegexpTokenizer(r'[a-zA-Z0-9]+')
cv = CountVectorizer(stop_words='english',ngram_range =
(1,1),toke nizer = token.tokenize)
text_counts =
cv.fit_transform(tweets['processed_tweets'].values.a stype('U'))
from transformers import
BertTokenizer,BertForSequenceClassificati on,AdamW
In [12]:
tokenizer = BertTokenizer.from_pretrained('bert-base-
uncased',do_l ower_case = True)
input_ids = []
attention_mask = []
for i in text:
    encoded_data =
    tokenizer.encode_plus( i,
    add_special_tokens=True,
    max_length=64,
    pad_to_max_length = True,
    return_attention_mask= True,
    return_tensors='pt')
    input_ids.append(encoded_data['input_ids'])
    attention_mask.append(encoded_data['attention_mask'])
input_ids = torch.cat(input_ids,dim=0)
attention_mask = torch.cat(attention_mask,dim=0)
labels = torch.tensor(labels)
```

```
from torch.utils.data import DataLoader,SequentialSampler,RandomSa
mpler,TensorDataset,random_split
In [15]:
dataset = TensorDataset(input_ids,attention_mask,labels)
train_size = int(0.8*len(dataset))
val_size = len(dataset) - train_size

train_dataset,val_dataset =

random_split(dataset,[train_size,val_s ize])

print('Training Size - ',train_size)
print('Validation Size - ',val_size)

train_dl = DataLoader(train_dataset,sampler =
RandomSampler(train_ dataset),
                      batch_size = 32)
val_dl = DataLoader(val_dataset,sampler =
SequentialSampler(val_da taset),
                      batch_size = 32)


model = BertForSequenceClassification.from_pretrained(
'bert-base-uncased',
num_labels = 2,
output_attentions = False,
output_hidden_states = False)


import random

seed_val = 17
random.seed(seed_val)
np.random.seed(seed_val)
torch.manual_seed(seed_val)
torch.cuda.manual_seed_all(seed_val)

device = torch.device('cuda' if torch.cuda.is_available() else 'cp
u')
model.to(device)

optimizer = AdamW(model.parameters(),lr = 2e-5,eps=1e-8)
In [22]:
from transformers import
get_linear_schedule_with_warmup epochs = 1
total_steps = len(train_dl)*epochs
scheduler =
get_linear_schedule_with_warmup(optimizer,num_warmup_s teps=0,
                                                num_training_steps=tota
l_steps)

def accuracy(preds,labels):
```

```
    pred_flat = np.argmax(preds,axis=1).flatten()
    label_flat = labels.flatten()
    return np.sum(pred_flat==label_flat)/len(label_flat)


def evaluate(dataloader_test):
    model.eval()
    loss_val_total = 0
    predictions,true_vals = [],[]
    for batch in dataloader_test:
        batch = tuple(b.to(device) for b in batch)
        inputs = {
            'input_ids':batch[0],
            'attention_mask': batch[1],
            'labels': batch[2]
        }
        with torch.no_grad():
            outputs = model(**inputs)
        loss = outputs[0]
        logits = outputs[1]
        loss_val_total += loss.item()
        logits = logits.detach().cpu().numpy()
        label_ids = inputs['labels'].cpu().numpy()
        predictions.append(logits)
        true_vals.append(label_ids)
    loss_val_avg = loss_val_total / len(dataloader_test)
    predictions = np.concatenate(predictions,axis=0)
    true_vals = np.concatenate(true_vals,axis=0) return
    loss_val_avg,predictions,true_vals


from tqdm.notebook import tqdm
torch.cuda.empty_cache()
for epoch in tqdm(range(1, epochs+1)):

    model.train()

    loss_train_total = 0

    progress_bar = tqdm(train_dl, desc='Epoch {:1d}'.format(epoch)
, leave=False, disable=False)

    for batch in progress_bar:

        model.zero_grad()

        batch = tuple(b.to(device) for b in batch)

     inputs = {'input_ids':         batch[0],
                  'attention_mask': batch[1],
                  'labels':         batch[2],
                 }
```

```
        outputs = model(**inputs)

        loss = outputs[0]
        loss_train_total += loss.item()
        loss.backward()

        torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)

        optimizer.step()
        scheduler.step()

        progress_bar.set_postfix({'training_loss':
'{:.3f}'.format (loss.item()/len(batch))})

    tqdm.write(f'\nEpoch {epoch}')

    loss_train_avg = loss_train_total/len(train_dl)
    tqdm.write(f'Training loss: {loss_train_avg}')

    val_loss, predictions, true_vals = evaluate(val_dl)
    val_acc = accuracy(predictions, true_vals)
    tqdm.write(f'Validation loss: {val_loss}')
    tqdm.write(f'Accuracy: {val_acc}')


output_dir = './'
model_to_save = model.module if hasattr(model, 'module') else
mode l
model_to_save.save_pretrained(output_dir)
tokenizer.save_pretrained(output_dir)




# Get the twiiter
data import pandas as
pd import requests
!pip install

hfda import hfda

params = {
 "source" : "twitter",
 "coin" : "bitcoin",
 "bin_size" : "T",
 "count_ptr" : 10,
 "start_ptr" : 0,
 "start_datetime" : "2012-05-01T00:00:00Z",
 "end_datetime" : "2021-11-30T00:00:00Z",
}

endpoint_url = "http://api-dev.augmento.ai/v0.1/events/aggregated" r

= requests.get(endpoint_url, params=params).json()
```

```
df_req = pd.DataFrame(r)


# Get the BitStamp data from http://api.bitcoincharts.com/v1/csv/

df = pd.read_csv('Data/bitstampUSD.csv.gz', compression='gzip')
df.rename(columns={
    "1315922016": "Timestamp",
    "5.800000000000": "Price",
    "1.000000000000": "Volume"
}, inplace=True)
df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')
df.set_index("Timestamp", inplace=True)

df['Price_by_Volume'] = df['Price'] * df['Volume']
df_vwap = df.resample('T').sum()
df_vwap['Weighted_Price'] = df_vwap['Price_by_Volume'] /
df_vwap['Volume']

df_ohlc = df['Price'].resample('T').ohlc()
df_ohlc = df_ohlc.join(df_vwap[['Volume',
'Price_by_Volume', 'Weighted_Price']])
df = df_ohlc

close_val = df.close

import numpy as np
from matplotlib import pyplot as plt
```

```
slope = pd.Series(np.gradient(close_val.values),
close_val.index, name='slope')
second_slope = pd.Series(np.gradient(slope.values), slope.index,
name='second_slope')
plt.style.use('seaborn')
plt.subplot(3, 1, 1)
close_val.plot()
plt.subplot(3, 1, 2)
slope.plot()
plt.subplot(3, 1, 3)
second_slope.plot()
df['1_slope'] = slope
df['2_slope'] = second_slope

df = df.dropna()
df['fractal_dim_roll60'] = df.close.rolling(60).apply(lambda x:
hfda.measure(x, 5))
df.to_csv('Data/resampled.csv')

df = pd.read_csv("Data/resampled.csv")
df = df.rename({"Timestamp":"date"}, axis=1)
df.date = pd.to_datetime(df.date)
df = df.set_index('date')
semantic = pd.read_csv('Data/btc_tweets_with_semantic.csv',
usecols=['date', 'semantic'])
semantic = semantic[~(pd.to_datetime(semantic.date,
errors='coerce').isnull())] semantic =
semantic.reset_index()
semantic.date = pd.to_datetime(semantic.date)
semantic = semantic.set_index('date')
semantic = semantic.resample('T').mean()
semantic = semantic.fillna(method='backfill')
semantic = semantic.drop(['level_0','index'], axis=1)

final_df = df.merge(semantic, left_index=True, right_index=True)
idx = pd.date_range("2021-02-05 10:52:00", "2021-11-26
23:59:00", freq="T")
final_df = final_df.reindex(idx,
fill_value=np.nan).fillna(method='backfill')
final_df.to_csv('final_df.csv')
final_df = final_df[['Volume', "Weighted_Price", '1_slope',
'2_slope', 'fractal_dim_roll60', 'semantic']]
final_df.to_csv('final_df_main_features.csv'



import glob
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn
import torch
```

```
import torch.nn as nn
from datetime import datetime


df = pd.read_csv('../input/bitcoin-semantic-fractal-dim-slope12-
vwap-vol/final_df_main_features.csv', index_col=0)


import plotly.graph_objs as go
from plotly.offline import iplot

def plot_dataset(df, title):
    data = []
    value = go.Scatter(
        x=df.index,
        y=df.Weighted_Price,
        mode="lines",
        name="values",
        marker=dict(),
        text=df.index,
        line=dict(color="rgba(0,0,0, 0.3)"),
    )
    data.append(value)

    layout = dict(
        title=title,
        xaxis=dict(title="Date", ticklen=5, zeroline=False),
        yaxis=dict(title="Value", ticklen=5, zeroline=False),
    )

    fig = dict(data=data, layout=layout)
    iplot(fig)


from sklearn.model_selection import train_test_split
def feature_label_split(df, target_col):
    y = df[[target_col]].iloc[1:, :]
    X = df.iloc[:-1. :]  #.drop(columns=[target_col])

    return X, y
```

```
def train_val_test_split(df, target_col, test_ratio):
    val_ratio = test_ratio / (1 - test_ratio)
    X, y = feature_label_split(df, target_col)
    X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=test_ratio, shuffle=False)
    X_train, X_val, y_train, y_val = train_test_split(X_train,
y_train, test_size=val_ratio, shuffle=False)
    return X_train, X_val, X_test, y_train, y_val, y_test

X_train, X_val, X_test, y_train, y_val, y_test =
train_val_test_split(df, 'Weighted_Price', 0.1)


from sklearn.preprocessing import
MinMaxScaler scaler = MinMaxScaler((0, 1))
X_train_arr = scaler.fit_transform(X_train)
X_val_arr = scaler.transform(X_val)
X_test_arr = scaler.transform(X_test)

y_train_arr = scaler.fit_transform(y_train)
y_val_arr = scaler.transform(y_val)
y_test_arr = scaler.transform(y_test)


from torch.utils.data import TensorDataset, DataLoader

batch_size = 128

train_features =
torch.Tensor(X_train_arr).to(torch.device("cuda:0"))
train_targets =
torch.Tensor(y_train_arr).to(torch.device("cuda:0"))
val_features = torch.Tensor(X_val_arr).to(torch.device("cuda:0"))
val_targets = torch.Tensor(y_val_arr).to(torch.device("cuda:0"))
test_features =
torch.Tensor(X_test_arr).to(torch.device("cuda:0"))
test_targets = torch.Tensor(y_test_arr).to(torch.device("cuda:0"))

train = TensorDataset(train_features, train_targets)
val = TensorDataset(val_features, val_targets)
test = TensorDataset(test_features, test_targets)


train_loader = DataLoader(train,
batch_size=batch_size, shuffle=False, drop_last=True)
val_loader = DataLoader(val, batch_size=batch_size,
shuffle=False, drop_last=True)
test_loader = DataLoader(test, batch_size=batch_size,
shuffle=False, drop_last=True)
```

```
test_loader_one = DataLoader(test, batch_size=1, shuffle=False,
drop_last=True)


class LSTMModel(nn.Module):
    def __init__(self, input_dim, hidden_dim,
layer_dim, output_dim, dropout_prob):
        super(LSTMModel, self).__init__()

        # Defining the number of layers and the nodes in each
layer
        self.hidden_dim = hidden_dim
        self.layer_dim = layer_dim

        # LSTM layers
        self.lstm = nn.LSTM(
            input_dim, hidden_dim, layer_dim, batch_first=True,
dropout=dropout_prob
        )

        # Fully connected layer
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        # Initializing hidden state for first input with
        zeros h0 = torch.zeros(self.layer_dim, x.size(0),
self.hidden_dim).requires_grad_().to(device)

        # Initializing cell state for first input with
        zeros c0 = torch.zeros(self.layer_dim, x.size(0),
self.hidden_dim).requires_grad_().to(device)

        # We need to detach as we are doing truncated
backpropagation through time (BPTT)
        # If we don't, we'll backprop all the way to the
start even after going through another batch
        # Forward propagation by passing in the input,
hidden state, and cell state into the model
        out, (hn, cn) = self.lstm(x, (h0.detach(), c0.detach()))

        # Reshaping the outputs in the shape of
(batch_size, seq_length, hidden_size)
        # so that it can fit into the fully connected layer
        out = out[:, -1, :]

        # Convert the final state to our desired output
shape (batch_size, output_dim)
        out = self.fc(out).to(device)

        return out
```

```
class Optimization:
    def __init__(self, model, loss_fn, optimizer):
        self.model = model
        self.loss_fn = loss_fn
        self.optimizer = optimizer
        self.train_losses = []
        self.val_losses = []

    def train_step(self, x, y):
        # Sets model to train mode
        self.model.to(device)
        self.model.train()
        # Makes predictions
        yhat = self.model(x)

        # Computes loss
        loss = self.loss_fn(y, yhat)

        # Computes gradients
        loss.backward()
        # Updates parameters and zeroes
        gradients self.optimizer.step()
        self.optimizer.zero_grad()
        # Returns the loss
        return loss.item()

    def train(self, train_loader, val_loader,
batch_size=64, n_epochs=1, n_features=1):
            model_path = f'{datetime.now().strftime("%Y-%m-%d.%H-
%M-%S")}'

            for epoch in range(1, n_epochs + 1):
                batch_losses = []
                for x_batch, y_batch in train_loader:
                    x_batch = x_batch.view([batch_size, -1,
n_features]).to(device)
                    y_batch = y_batch.to(device)
                    loss = self.train_step(x_batch, y_batch)
                    batch_losses.append(loss)
                training_loss = np.mean(batch_losses)
                self.train_losses.append(training_loss)

                with torch.no_grad():
                    batch_val_losses = []
                    for x_val, y_val in val_loader:
                        x_val = x_val.view([batch_size, -1,
n_features]).to(device)
                        y_val = y_val.to(device)
                        self.model.eval()
```

```
                        yhat = self.model(x_val)
                        val_loss = self.loss_fn(y_val,
yhat).item()
                        batch_val_losses.append(val_loss)
                    validation_loss = np.mean(batch_val_losses)
                    self.val_losses.append(validation_loss)

                print(
                    f"[{epoch}/{n_epochs}] Training loss:
{training_loss:.4f}\t Validation loss: {validation_loss:.4f}"
                )

            torch.save(self.model.state_dict(), model_path)

    def evaluate(self, test_loader, batch_size=1, n_features=1):
        with torch.no_grad():
            predictions = []
            values = []
            for x_test, y_test in test_loader:
                x_test = x_test.view([batch_size, -1,
n_features]).to(device)
                y_test = y_test.to(device)
                self.model.eval()
                yhat = self.model(x_test)
                predictions.append(yhat.cpu().detach().numpy())
                values.append(y_test.cpu().detach().numpy())

        return predictions, values

    def plot_losses(self):
        plt.plot(self.train_losses, label="Training loss")
        plt.plot(self.val_losses, label="Validation loss")
        plt.legend()
        plt.title("Losses")
        plt.show()
        plt.close()


import torch.optim as optim

device = torch.device('cuda:0')

input_dim = len(X_train.columns)
output_dim = 1
hidden_dim = 500
layer_dim = 5
batch_size = 128
dropout = 0.2
n_epochs = 300
learning_rate = 1e-6
weight_decay = 1e-8
```

```
model_params = {'input_dim': input_dim,
                'hidden_dim' : hidden_dim,
                'layer_dim' : layer_dim,
                'output_dim' : output_dim,
                'dropout_prob' : dropout}

model = LSTMModel(**model_params)

loss_fn = nn.MSELoss(reduction="mean")
optimizer = optim.Adam(model.parameters(),
lr=learning_rate, weight_decay=weight_decay)
opt = Optimization(model=model,
loss_fn=loss_fn, optimizer=optimizer)
opt.train(train_loader, val_loader,
batch_size=batch_size, n_epochs=n_epochs,
n_features=input_dim) opt.plot_losses()


predictions, values = opt.evaluate(test_loader_one,
batch_size=1, n_features=input_dim)


def inverse_transform(scaler, df, columns):
    for col in columns:
        df[col] = scaler.inverse_transform(df[col])
    return df


def format_predictions(predictions, values, df_test, scaler):
    vals = np.concatenate(values, axis=0).ravel()
    preds = np.concatenate(predictions, axis=0).ravel()
    df_result = pd.DataFrame(data={"value": vals, "prediction":
preds}, index=df_test.head(len(vals)).index)
    df_result = df_result.sort_index()
    df_result = inverse_transform(scaler, df_result,
[["value", "prediction"]])
    return df_result


df_result = format_predictions(predictions, values,
X_test, scaler)
df_result.plot()
r2_score(df_result)
```