

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

«На правах рукопису»
УДК [004.8'912::336.76:659.3:159.942]

До захисту допущено:
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Системи і методи штучного інтелекту»
зі спеціальності 122 «Комп'ютерні науки»
на тему: «Дослідження впливу новинних повідомлень на ринок
криптовалют з використанням методів штучного інтелекту»**

Виконав:
студент II курсу, групи КІ-31мп
Дорошенко Сергій Володимирович _____

Науковий керівник:
доцент кафедри штучного інтелекту, к.ф.-м.н., доцент,
Пишнограсєв Іван Олександрович _____

Рецензент:
доцент кафедри обчислювальної техніки, к.т.н., доцент,
Болдак Андрій Олександрович _____

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

Студент: _____

Київ – 2024 року

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

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«29» серпня 2024 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Дорошенку Сергію Володимировичу

1. **Тема дисертації** «Дослідження впливу новинних повідомлень на ринок криптовалют з використанням методів штучного інтелекту», науковий керівник дисертації Пишнограєв Іван Олександрович, доцент, к.ф.-м.н., затверджені наказом по ННПСА від «07» листопада 2024 р. №5001-с.
2. **Термін подання студентом дисертації:** 13.12.2024.
3. **Об'єкт дослідження:** ринок криптовалют, його поведінка під впливом зовнішніх факторів, включаючи новинні повідомлення.
4. **Вихідні дані:** набір історичних даних про ціни криптовалют та новинних повідомлень, пов'язаних з ринком криптовалют.

5. Перелік завдань, які потрібно розробити:

- дослідити актуальність обраної теми та вплив новинних повідомлень на криптовалютний ринок.
- дослідити та проаналізувати існуючі методи прогнозування цін на криптовалюту, зокрема із врахуванням новинних даних.
- вибрати оптимальну архітектуру системи для обробки новинних потоків та прогнозування змін на ринку криптовалют.
- навчити моделі машинного навчання, інтегрувавши новинні повідомлення як зовнішні змінні, та провести експерименти.
- проаналізувати результати роботи системи та визначити вплив емоційного забарвлення новин на точність прогнозів.
- порівняти результати запропонованої системи з існуючими моделями прогнозування криптовалютного ринку, які враховують або не враховують новинні дані та їх емоційне забарвлення, а також моделі, що базуються виключно на історичних цінових даних. Порівняння виконано за точністю прогнозування під різними конфігураціями моделі.
- оформити пояснювальну записку.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- презентація порядку виконання роботи.

7. Орієнтовний перелік публікацій: заплановано апробування результатів дослідження на міжнародних та всеукраїнських конференціях.

8. Дата видачі завдання: 30.08.2024.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми МД	02.09.2024 – 06.09.2024	Виконано
2	Ознайомлення зі структурою МД згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	09.09.2024 – 13.09.2024	Виконано
3	Ознайомлення з ДСТУ 3008-2015 та стандартами ЄСПД	16.09.2024 – 29.09.2024	Виконано
4	Проведення дослідження за темою МД під керівництвом наукового керівника	30.09.2024 – 13.10.2024	Виконано
5	Завершення роботи над першим варіантом частини МД	14.10.2024 – 17.11.2024	Виконано
6	Проведення роботи над експериментальною частиною МД	18.11.2024 – 01.12.2024	Виконано
7	Проведення роботи над програмним продуктом	02.12.2024 – 08.12.2024	Виконано
8	Оформлення МД та аналіз отриманих результатів	09.12.2024 – 13.12.2024	Виконано

Студент

Сергій ДОРОШЕНКО

Науковий керівник

Іван ПИШНОГРАЄВ

РЕФЕРАТ

Магістерська дисертація: 109 сторінок, 17 рисунків, 17 таблиць, додаток, 35 посилань.

Об'єкт дослідження – ринок криптовалют, його поведінка під впливом зовнішніх факторів, включаючи новинні повідомлення.

Предмет дослідження – вплив новинних повідомлень на ціни основних криптовалют, зокрема через їх емоційне забарвлення та структурні характеристики.

Мета роботи – створення системи прогнозування змін ринку криптовалют на основі аналізу новинних потоків із використанням методів штучного інтелекту.

Дослідження включає аналіз новинних повідомлень і історичних даних про криптовалюту, розробку методології обробки текстових даних, побудову моделей машинного навчання та оцінку їхньої ефективності. Новизна роботи полягає у врахуванні унікальних характеристик ринку криптовалют, таких як мінливість цін і залежність від соціальних трендів. Використання новин як зовнішніх факторів підвищило точність прогнозів.

Запропонована система продемонструвала високу ефективність і враховує емоційні та структурні впливи на ринок.

КРИПТОВАЛЮТИ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ ЦІН, АНАЛІЗ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ НОВИН, ОБРОБКА ТЕКСТОВИХ ДАНИХ

ABSTRACT

Master's thesis: 109 pages, 17 figures, 17 tables, appendix, 35 references.

Research object – the cryptocurrency market and its behavior under the influence of external factors, including news messages.

Subject of research – the impact of news messages on the prices of major cryptocurrencies, particularly through their emotional tone and structural features.

Purpose of the work – to develop a system for forecasting cryptocurrency market changes based on news analysis using artificial intelligence methods.

The study includes the analysis of news messages and historical data, the development of a methodology for textual data processing, the construction of machine learning models, and the evaluation of their performance. The novelty lies in considering the unique characteristics of the cryptocurrency market, such as price volatility and dependence on social trends. News-based external factors significantly improved prediction accuracy.

The proposed system showed high effectiveness and accounted for emotional and structural impacts on the market.

CRYPTOCURRENCIES, MACHINE LEARNING, PRICE FORECASTING,
EMOTIONAL TONE ANALYSIS OF NEWS, TEXTUAL DATA PROCESSING

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ОГЛЯД ДОСЛІДЖЕНЬ ВПЛИВУ НОВИНИХ ПОВІДОМЛЕНЬ НА РИНОК КРИПТОВАЛЮТ	11
1.1 Вплив новин на фінансові ринки	11
<i>1.1.1 Загальна характеристика новин як фактору впливу на ринки</i>	<i>11</i>
<i>1.1.2 Типи новин</i>	<i>12</i>
<i>1.1.3 Механізми впливу новин на настрої інвесторів і динаміку цін.</i>	<i>14</i>
<i>1.1.4 Особливості впливу новин на криптовалютні ринки.</i>	<i>15</i>
1.2 Методи аналізу впливу новин на криптовалютний ринок	16
<i>1.2.1 Класичні підходи: економетричні методи, регресійні моделі</i>	<i>16</i>
<i>1.2.2 Методи машинного навчання для аналізу тексту та прогнозування</i>	<i>19</i>
1.2.2.1 Аналіз текстових даних.....	19
1.2.2.2 Методи регресії.....	21
<i>1.2.3 Використання NLP для аналізу новинних повідомлень.</i>	<i>22</i>
<i>1.2.4 Застосування нейронних мереж для прогнозування цін криптовалют.</i>	<i>24</i>
.....	24
Висновки до розділу 1	25
РОЗДІЛ 2 МЕТОДИ ТА ПІДХОДИ ДО ПРОГНОЗУВАННЯ ДИНАМІКИ ЦІН КРИПТОВАЛЮТ	28
2.1 Постановка задачі.....	28
2.2 Існуючі підходи вирішення задачі.....	29
2.2.1 <i>ARIMA</i>	29
2.2.2 <i>Рекурентні нейронні мережі</i>	30
2.2.3 <i>LSTM</i>	32
2.2.4 <i>GRU</i>	34
2.3 Опис методу для кодування назв та опису новин	37
2.4 Фінальна архітектура системи.....	38
2.5 Метрики задачі авторегресії.....	40
Висновки до розділу 2	42

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ЕКСПЕРЕМЕНТІВ	44
3.1 Аналіз набору даних.....	44
3.1.1 <i>Історичні дані Bitcoin</i>	44
3.1.2 <i>датасет з крипто-новинами.....</i>	46
3.2 Результати.....	47
3.2.1 <i>Навчання з екзогенними фічами, без фіч з новинами.</i>	47
3.2.2 <i>Навчання з екзогенними фічами, разом з фічами з новинами.</i>	49
3.2.3 <i>Навчання без екзогенних фіч та без фіч з новинами.</i>	52
3.2.4 <i>Навчання без екзогенних фіч та з фічами з новинами.</i>	55
3.3 Платформа, мова програмування та бібліотеки	58
Висновки до розділу 3	59
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ	61
4.1 Опис ідеї стартап-проекту.....	62
4.2 Технологічний аудит стартап-проекту.....	64
4.3 Аналіз ринкової стратегії стартап-проекту	72
4.4 Розробка маркетингової стратегії стартап-проекту	77
Висновки до розділу 4	81
ВИСНОВКИ.....	83
ПЕРЕЛІК ПОСИЛАНЬ	85
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ.....	88

ВСТУП

Криптовалютний ринок сьогодні є одним із найбільш динамічних і непередбачуваних секторів фінансової системи. Його унікальність полягає у високій волатильності, глобальному охопленні та особливій чутливості до зовнішніх чинників. Значну роль серед них відіграють новини, які здатні істотно впливати на настрої інвесторів і змінювати цінові тренди. У міру зростання популярності криптовалют і їх інтеграції в міжнародні економічні процеси питання про те, як інформація формує динаміку ринку, стає дедалі важливішим.

Особливість криптовалютного ринку полягає у його залежності від різноманітних новин: регуляторних змін, економічних прогнозів, технічних інновацій, соціальних трендів тощо. Поширення таких повідомлень відбувається дуже швидко завдяки соціальним мережам і спеціалізованим онлайн-платформам. Це створює великі обсяги даних, які вимагають автоматизованого аналізу для розуміння ключових тенденцій.

Традиційні підходи, засновані на фундаментальному чи технічному аналізі, часто не враховують вплив інформаційного поля, а отже, можуть бути недостатньо ефективними. У таких умовах виникає потреба у впровадженні сучасних технологій обробки даних, зокрема методів штучного інтелекту. Ці інструменти дозволяють працювати з великими масивами інформації, швидко обробляти текстові повідомлення, аналізувати їх зміст і тональність, а також прогнозувати наслідки для ринкової динаміки.

Метою цього дослідження є аналіз впливу новинних повідомлень на ринок криптовалют із використанням інструментів штучного інтелекту. Особлива увага приділяється сучасним підходам до аналізу текстових даних, таким як методи обробки природної мови та моделі глибокого навчання. У ході роботи буде зібрано репрезентативний набір даних, що включає новини про криптовалюту, а також історичні дані про ціни. Ці дані будуть використані

для розробки моделей, здатних визначати тональність новин і оцінювати їхній вплив на ринкові показники.

На основі отриманих результатів буде здійснено аналіз ключових факторів, що впливають на короткострокові зміни цін криптовалют. Це дозволить не лише оцінити ефективність застосованих моделей, але й сприятиме кращому розумінню механізмів взаємодії інформаційних потоків і ринкової динаміки. Отримані висновки можуть бути корисними для інвесторів, аналітиків і регуляторів, які прагнуть ухвалювати більш обґрунтовані рішення в умовах сучасного інформаційного середовища.

РОЗДІЛ 1 ОГЛЯД ДОСЛІДЖЕНЬ ВПЛИВУ НОВИНИХ ПОВІДОМЛЕНЬ НА РИНОК КРИПТОВАЛЮТ

1.1 Вплив новин на фінансові ринки

1.1.1 Загальна характеристика новин як фактору впливу на ринки

Новини є ключовим джерелом інформації для учасників фінансових ринків, оскільки вони можуть миттєво змінювати настрої інвесторів і впливати на ціни активів. У сучасній економіці новини розглядаються як критичний елемент, що формує очікування інвесторів, змінює споживчі настрої та сприяє перерозподілу ресурсів на ринках [1]. Вплив новин визначається їх змістом, формою подання та масштабом поширення, що дозволяє їм відігравати роль потужного каталізатора ринкових змін.

Одним із важливих аспектів впливу новин є їхня швидкість поширення. Завдяки сучасним технологіям інформація стає доступною в режимі реального часу, що зменшує затримку між виникненням новини та реакцією ринку. Зокрема, електронні платформи, такі як Bloomberg або Reuters, забезпечують миттєве поширення інформації, дозволяючи трейдерам швидко адаптувати свої стратегії [2]. У цьому контексті зростає важливість автоматизованих систем аналізу новин, які дають змогу оперативно оцінювати їхній зміст та передбачати можливі наслідки для ринкових цін.

Важливо зазначити, що вплив новин на фінансові ринки значною мірою залежить від їхньої природи. Наприклад, макроекономічні дані, такі як зміни процентних ставок або рівень інфляції, можуть мати довгостроковий вплив на ринок. У той час як корпоративні новини, наприклад, звіти про прибутки чи анонси злиттів і поглинань, часто викликають короткострокові коливання цін на відповідні активи [3]. Ефективність новин у зміні ринкової динаміки також пов'язана з їхньою достовірністю та очікуванням аудиторії. Інвестори часто порівнюють отриману інформацію з попередніми прогнозами, що створює додатковий рівень складності в аналізі їхнього впливу.

Іншим важливим аспектом є тональність новин. Дослідження показують, що позитивні новини зазвичай підвищують рівень впевненості інвесторів, сприяючи зростанню цін на активи, тоді як негативні повідомлення здатні викликати паніку та зниження вартості активів [4]. У цьому контексті розвиток технологій аналізу тональності, таких як методи обробки природної мови (NLP), дозволяє більш точно визначати потенційний вплив новин на ринок.

Нарешті, варто зазначити, що новини є не лише джерелом інформації, але й інструментом маніпуляції. Наприклад, фальшиві новини можуть викликати штучні коливання на ринку, які використовуються для отримання вигоди певними гравцями [5]. Це підкреслює важливість розробки та впровадження інструментів для перевірки достовірності новин та оцінки їхнього впливу на ринки.

Таким чином, новини є фундаментальним фактором, що формує поведінку фінансових ринків, і їхній вплив залежить від швидкості поширення, природи, достовірності та тональності. Ефективне використання новин у торгівлі та інвестуванні вимагає поєднання класичних підходів і сучасних технологій аналізу даних.

1.1.2 Типи новин

Різні типи новин мають специфічний вплив на фінансові ринки, залежно від їхнього змісту, джерела та контексту. До основних типів відносяться економічні, регуляторні, соціальні новини, а також інші події, що впливають на ринкові настрої.

Економічні новини. Ця категорія включає інформацію про макроекономічні показники, такі як зміни ВВП, рівень безробіття, інфляція, процентні ставки та зовнішньоекономічний баланс. Вони мають безпосередній вплив на формування інвестиційних стратегій і оцінку ризиків. Наприклад,

оголошення про зростання ВВП часто сприймається як позитивний сигнал для інвесторів, сприяючи зростанню фондових індексів. Водночас новини про підвищення процентних ставок можуть викликати зниження акцій, оскільки це збільшує витрати на кредитування [1].

Регуляторні новини. До цієї категорії відносяться законодавчі ініціативи, постанови урядів та центральних банків, а також політичні рішення, які можуть впливати на ринки. Наприклад, введення нових податків або регуляторних обмежень на торгівлю криптовалютами може призводити до різких змін у їхній вартості. Відомо, що заяви регуляторів про можливу заборону криптовалютних операцій або нові правила щодо їхнього використання часто супроводжуються суттєвою волатильністю [2].

Соціальні новини. Цей тип новин включає події, що стосуються соціальної стабільності, громадської думки та глобальних трендів. Наприклад, новини про соціальні протести, зміни урядів чи навіть значущі культурні події можуть впливати на ринки. Для криптовалют особливе значення мають події, які формують довіру або недовіру громадськості до децентралізованих фінансових систем [3].

Інші новини. До цієї категорії належать новини про технологічні інновації, природні катаклізми, військові конфлікти та інші події, що не входять до зазначених категорій, але мають потенційно значущий вплив. Наприклад, технологічні прориви, пов'язані з блокчейном або криптографією, можуть стимулювати зростання інтересу до криптовалют. Водночас природні катаклізми чи пандемії можуть знижувати інвестиційну активність, створюючи додаткові ризики [4].

Розуміння типів новин і їхнього потенційного впливу є критично важливим для аналізу фінансових ринків, зокрема криптовалютного, оскільки дозволяє інвесторам оперативніше адаптувати свої стратегії.

1.1.3 Механізми впливу новин на настрої інвесторів і динаміку цін.

Механізми впливу новин на фінансові ринки охоплюють широкий спектр взаємодій між інформацією, яка надходить, та реакцією учасників ринку. У цьому контексті важливо зрозуміти, як новини трансформуються в емоційні та раціональні реакції, що в підсумку впливають на ціни активів.

Емоційні реакції. Новини можуть викликати миттєві емоційні реакції серед інвесторів, які, у свою чергу, формують попит та пропозицію на ринку. Наприклад, позитивні новини, такі як партнерство великих компаній з криптовалютними проєктами, можуть стимулювати масовий інтерес до купівлі криптовалют. Водночас негативні новини, як-от злом великих криптобірж, призводять до панічних продажів і падіння цін [1].

Раціональні реакції. Інша категорія реакцій пов'язана з раціональним аналізом новинного контенту. Інвестори аналізують потенційні довгострокові наслідки новин для конкретних активів чи ринку загалом. Наприклад, повідомлення про нові регуляторні вимоги може змусити інвесторів передбачити зменшення ліквідності та зростання витрат на ведення бізнесу, що негативно позначається на ринку [2].

Роль алгоритмічної торгівлі. Новини дедалі частіше впливають на ринки через алгоритмічну торгівлю. Багато трейдерів використовують автоматизовані системи, які аналізують новинні потоки в режимі реального часу. Такі системи реагують швидше за людину, тому ціни активів можуть змінюватися ще до того, як більшість інвесторів обробить інформацію [3].

Соціальні механізми. Соціальні мережі та платформи новин стають важливим каналом поширення інформації, що впливає на поведінку ринку. Наприклад, у випадку криптовалют новини, які поширюються через Twitter або Reddit, можуть створювати колективні настрої серед інвесторів, що впливають на формування трендів [4].

Розуміння цих механізмів дозволяє аналітикам краще прогнозувати ринкову динаміку та розробляти стратегії, що враховують як емоційні, так і раціональні аспекти поведінки інвесторів.

1.1.4 Особливості впливу новин на криптовалютні ринки.

Криптовалютні ринки мають унікальні характеристики, які значно відрізняють їх від традиційних фінансових ринків, і це відображається на впливі новин на динаміку цін. По-перше, ринок криптовалют функціонує 24/7, що створює постійні можливості для реагування на інформаційні потоки. Новини, які надходять у будь-який час доби, можуть моментально вплинути на настрої інвесторів і викликати значні коливання цін. Така безперервність ринку змушує інвесторів оперативно оцінювати інформацію та ухвалювати рішення, що підсилює роль емоційного компонента в торгівлі криптовалютами.

По-друге, криптовалютний ринок є значно більш волатильним, ніж традиційні фінансові ринки, через порівняно низьку капіталізацію і високу спекулятивність. Ця волатильність підсилює вплив новин, оскільки навіть відносно невеликі за значенням події можуть викликати значні зміни цін. Наприклад, новини про хакерські атаки на біржі, запровадження нових регуляцій або прийняття криптовалют великими компаніями можуть спричинити раптові зміни вартості активів. Також слід враховувати, що відсутність централізованого регулювання підвищує рівень інформаційної асиметрії, що сприяє швидкому поширенню як достовірних, так і недостовірних даних.

По-третє, важливою особливістю є значна роль соціальних мереж та онлайн-форумів у формуванні ринкових трендів. Багато інвесторів на криптовалютному ринку є новачками або частиною спільнот, які орієнтуються

на думки лідерів громадської думки. Наприклад, твіти відомих персон, таких як Ілон Маск, часто мають безпосередній вплив на динаміку цін криптовалют, як це було у випадку з Dogecoin. Такі події підкреслюють важливість аналізу не лише традиційних новинних платформ, але й соціальних медіа як джерел інформації, що впливає на ринок.

Ще однією важливою особливістю є глобальний характер криптовалютного ринку. На відміну від локалізованих традиційних ринків, криптовалюти обертаються по всьому світу. Це означає, що новини, які стосуються однієї країни чи регіону, можуть мати глобальний вплив. Наприклад, регуляторні зміни у великих економіках, таких як США чи Китай, мають тенденцію викликати значні зміни на світовому ринку криптовалют, незалежно від того, де знаходяться основні гравці.

Наостанок, слід зазначити, що інформація про нові технології та інновації у світі блокчейну також може значно впливати на криптовалютний ринок. Оголошення про впровадження нових функцій, таких як оновлення мережі або запуск нових протоколів, часто спричиняють зростання ціни відповідних криптовалют. Однак, через складність технологічних змін, багато інвесторів можуть неправильно інтерпретувати інформацію, що знову ж таки веде до волатильності.

1.2 Методи аналізу впливу новин на криптовалютний ринок

1.2.1 Класичні підходи: економетричні методи, регресійні моделі

Економетричні методи та регресійні моделі широко використовуються для аналізу впливу новин на фінансові ринки, включаючи криптовалютний. Їх основна мета — кількісно оцінити взаємозв'язок між змінними, такими як тональність новин, частота публікацій, і ключовими показниками ринку, зокрема ціною, обсягами торгівлі чи волатильністю. До найбільш популярних

підходів належать методи лінійної та логістичної регресії, авторегресійні моделі (ARIMA), а також VAR (векторна авторегресія).

Лінійна регресія є основним інструментом для оцінки залежностей між незалежними змінними X_1, X_2, \dots, X_n і залежною змінною Y , яка може бути, наприклад, зміною ціни криптовалюти. Рівняння лінійної регресії має вигляд:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon, \quad (1.1)$$

де Y — залежна змінна, X_1, X_2, \dots, X_n — незалежні змінні, $\beta_0, \beta_1, \dots, \beta_n$ — коефіцієнти регресії, а ε — залишкова похибка. Наприклад, у роботі [6] оцінювався вплив кількості позитивних новин про Bitcoin на його ціну за допомогою цього методу.

Логістична регресія застосовується для аналізу двійкових результатів, наприклад, чи призведе новина до зростання або падіння ринку. Вона описується рівнянням:

$$P(Y = 1|X) = \frac{1}{1 + e^{-\beta_0 - \beta_1 X_1 - \dots - \beta_n X_n}}, \quad (1.2)$$

де $P(Y = 1|X)$ — ймовірність позитивного результату за наявності змінних X_1, X_2, \dots, X_n . Дослідження [7] показало, як можна використовувати логістичну регресію для оцінки ймовірності росту ціни криптовалюти на основі тональності новин.

ARIMA (AutoRegressive Integrated Moving Average) є популярною моделлю для прогнозування часових рядів. Вона враховує залежності між минулими значеннями ряду та похибками прогнозу. Рівняння для ARIMA (p, d, q) має вигляд:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}, \quad (1.3)$$

де p — порядок авторегресії, d — кількість диференціювань, q — порядок ковзного середнього, ε_t — випадкова похибка. У дослідженні [8] ARIMA використовувалася для моделювання волатильності криптовалютного ринку після виходу новин про регуляторні зміни.

VAR (векторна авторегресія) моделює взаємозалежність між кількома змінними, наприклад, обсягами новин, їхньою тональністю та ціною активів. Це дозволяє одночасно аналізувати динаміку кількох ринкових показників. У роботі [9] метод VAR застосовувався для дослідження впливу соціальних медіа на криптовалютний ринок.

Для оцінки якості побудованих моделей використовуються такі показники, як коефіцієнт детермінації R^2

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}, \quad (1.4)$$

де Y_i — фактичні значення, \hat{Y}_i — прогнозовані значення, \bar{Y} — середнє значення залежної змінної. Значення R^2 , близьке до 1, свідчить про високу якість моделі [10].

Класичні методи мають обмеження, зокрема нездатність моделювати складні нелінійні взаємозв'язки, властиві криптовалютному ринку. Проте їхня простота, прозорість і доступність роблять їх цінними інструментами для попереднього аналізу.

1.2.2 Методи машинного навчання для аналізу тексту та прогнозування

1.2.2.1 Аналіз текстових даних

Обробка тексту є фундаментальним етапом для роботи з новинними даними у фінансовому аналізі. Основні задачі цього етапу включають векторизацію тексту, аналіз тональності та визначення ключових тем.

Векторизація тексту. Текстові дані перетворюються у числові представлення, придатні для роботи з алгоритмами машинного навчання. Основні методи включають:

1. Bag of Words (BoW): Формується матриця частот слів у текстах. Наприклад, якщо у нас є три документи, BoW виглядатиме як:

Документ 1: {Bitcoin, price, increase},

Документ 2: {Ethereum, regulation, ban}

Вектор для кожного документа відображатиме кількість згадок кожного слова [11].

2. TF-IDF (Term Frequency-Inverse Document Frequency): Формула TF-IDF для кожного слова t в документі d :

$$TF - IDF(t, d) = TF(t, d) \cdot IDF(t), \quad (1.5)$$

де:

$$TF(t, d) = \frac{\text{Кількість згадок } t \text{ у } d}{\text{Загальна кількість слів у } d}, \quad (1.6)$$

$$IDF(t) = \log \frac{\text{Загальна кількість документів}}{\text{Кількість документів із } t}, \quad (1.7)$$

3. Word Embeddings (Word2Vec, GloVe): Замість частот вектори слів

представляють їхній контекст у багатовимірному просторі. Наприклад, слова «Bitcoin» і «Ethereum» можуть бути розташовані ближче у векторному просторі через спільний контекст [12].

Замість традиційної векторизації (TF-IDF чи Bag of Words), сучасні методи пропонують підходи на основі контекстуального розуміння текстів. Основні з них:

1. Sentence Embeddings: У цьому підході текст перетворюється у вектор, що представляє весь зміст речення чи документа. Технології, такі як Universal Sentence Encoder (USE) або SBERT, використовуються для обчислення векторів речень. Формула для подібності між двома векторами:

$$\text{textCosineSimilarity}(u, v) = \frac{u \cdot v}{|u||v|}, \quad (1.8)$$

де u і v — вектори речень.

2. Transformers: Архітектури на основі трансформерів (наприклад, BERT, RoBERTa) забезпечують контекстуальне розуміння текстів. Для кожного слова обчислюється вектор, який залежить від контексту, в якому це слово використовується [3]. Формула уваги:

$$\text{textAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1.9)$$

де Q , K , V — матриці запитів, ключів і значень відповідно, d_k — розмірність ключів.

3. Аналіз тональності. Аналіз тональності дозволяє визначати, чи є новина позитивною, негативною або нейтральною. Основні підходи включають:

Лексиконний аналіз: Використання словників, таких як Loughran McDonald, які класифікують слова за тональністю [13].

Алгоритми машинного навчання: Класифікатори, такі як Naive Bayes, Logistic Regression, або моделі NLP (BERT, RoBERTa) [14].

Приклад використання логістичної регресії для класифікації тональності:

$$P(Y = 1|X) = \frac{1}{1+e^{-\beta_0-\beta_1X_1-\dots-\beta_nX_n}}, \quad (1.10)$$

де $P(Y = 1|X)$ — ймовірність позитивної тональності.

1.2.2.2 Методи регресії

Регресія використовується для прогнозування кількісних значень, наприклад, зміни ціни криптовалюти після появи новини.

Лінійна регресія: Проста у реалізації, але обмежена для нелінійних залежностей. Формула регресії:

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \dots + \beta_nX_n + \varepsilon, \quad (1.11)$$

де Y — прогнозована змінна, X_i — вхідні змінні, β_i — коефіцієнти, ε — випадкова похибка [15].

Гرادієнтне підсилення (Gradient Boosting): Алгоритм поетапно оптимізує похибку моделі. Формула для оновлення дерева:

$$F_m(x) = F_{m-1}(x) + \gamma \cdot \operatorname{argmin}_\gamma \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)), \quad (1.12)$$

де $F_m(x)$ — поточна модель, L — функція втрат, h_m — нове дерево, γ — коефіцієнт навчання [16].

Моделі, такі як багат шаровий перцептрон (MLP), використовуються для регресійних задач. Формула для одного нейрона:

$$z = \sigma(\sum_{i=1}^n w_i x_i + b), \quad (1.13)$$

де z — вихід нейрона, w_i — ваги, x_i — вхідні значення, b — зміщення, σ — активаційна функція [17].

1.2.3 Використання NLP для аналізу новинних повідомлень.

Розвиток сучасних методів обробки природної мови (NLP, Natural Language Processing) надає широкі можливості для всебічного дослідження інформаційних потоків, які впливають на криптовалютний ринок. Зокрема, технології NLP дозволяють оперативно аналізувати великі обсяги новинних повідомлень, виявляти в них ключові теми, суб'єкти, об'єкти та визначати тональність чи емоційне забарвлення текстів. У контексті криптовалютного ринку це дозволяє інвесторам, аналітикам та автоматизованим торговим системам робити висновки щодо очікуваних тенденцій курсу, рівня довіри до певних монет, оцінювати загальну динаміку інформаційного поля та зв'язок між інформаційними імпульсами та волатильністю цін.

Основні завдання NLP в контексті аналізу криптовалютних новин:

Лематизація та токенізація текстів: початковий етап підготовки даних, коли непотрібні символи, пунктуація та стоп-слова вилучаються, а слова нормалізуються до базових форм [18].

Розпізнавання сутностей (Named Entity Recognition, NER): визначення імен власних, назв криптовалют, бірж, компаній, ключових постатей або регуляторних органів, що впливають на ринок [19].

Визначення настрою (Sentiment Analysis): оцінювання загального емоційного тону новини (позитивний, нейтральний, негативний) для подальшої кількісної оцінки інформаційного фону [20].

Тематичне моделювання (Topic Modeling): групування повідомлень за смисловими темами з використанням методів на кшталт LDA (Latent Dirichlet Allocation) [21].

Виявлення ставлення до ризику (Risk Attitude Analysis): оцінка того, наскільки новинний контент посилює або послаблює інтерес інвесторів до ризикових операцій.

Формалізація аналізу настрою: Одним з базових підходів є обчислення інтегральної оцінки настрою новини на основі суми настроєвих оцінок слів. Нехай новина складається з N слів, кожному слову w_i відповідає значення настрою $S(w_i)$ з певного лексикону. Тоді інтегральний настрій новини S_{total} можна оцінити як середнє арифметичне:

$$S_{total} = \frac{1}{N} \sum_{i=1}^N S(w_i). \quad (1.14)$$

Однак, у реальній практиці використовують більш складні моделі, зокрема нейронні мережі на основі трансформерів (наприклад, BERT), які враховують контекст слів у реченні [22].

Використання NLP у поєднанні з методами машинного навчання дозволяє покращити точність прогнозування цінних змін. Наприклад, у дослідженні [20] було показано, що поєднання аналізу тональності твітів та новин з моделями прогнозування динаміки біткоїна може суттєво підвищити точність передбачення. Аналогічно, у працях [18, 19] продемонстровано, що аналіз інформаційного поля навколо регуляцій криптовалютного ринку, згадок про великі біржі чи інституційних інвесторів допомагає визначати ступінь впливу новин на формування очікувань учасників ринку.

Крім того, усвідомлення інформаційного тла дозволяє автоматизувати торговельні стратегії. Наприклад, у роботі [22] було використано нейронні мережі для класифікації новин за типами та визначення потенційних сигналів купівлі або продажу. Це підтверджує, що NLP-інструментарій здатен

слугувати ефективним допоміжним засобом для учасників криптовалютного ринку.

1.2.4 Застосування нейронних мереж для прогнозування цін криптовалют.

На відміну від традиційних статистичних методів, таких як ARIMA чи GARCH, нейронні мережі здатні автоматично виявляти складні нелінійні залежності між даними без необхідності ручного створення ознак. Крім того, вони дозволяють враховувати широкий спектр вхідної інформації, включаючи технічні індикатори, макроекономічні фактори, текстові дані, сентимент і інші сигнали з новинних джерел та соціальних мереж.

Основні підходи при застосуванні нейронних мереж для прогнозування цін криптовалют:

Рекурентні нейронні мережі (RNN) та LSTM/GRU: Ранній досвід застосування нейронних мереж до часового ряду цін криптовалют включав використання RNN-архітектур (Recurrent Neural Networks), зокрема LSTM (Long Short-Term Memory) або GRU (Gated Recurrent Unit) [23, 24]. Такі архітектури добре підходять для аналізу послідовностей, оскільки враховують часову залежність та дозволяють моделі "запам'ятовувати" важливу інформацію з минулих значень.

Конволюційні нейронні мережі (CNN) для часових рядів: Хоча CNN широко використовуються для аналізу зображень, їх можна застосувати і до часових рядів, "розглядаючи" вікно історичних цін як набір фрагментів даних. CNN можуть автоматично виявляти локальні шаблони у вхідних сигналах. Поєднання CNN із рекурентними шарами чи трансформерами може підвищити точність прогнозів [25].

Трансформери (Transformers): Останніми роками трансформерні моделі (наприклад, BERT, GPT, а також спеціалізовані для часових рядів варіації моделей) набирають популярності у фінансовому прогнозуванні. На основі механізму уваги (Attention), трансформери ефективно обробляють послідовні дані, не покладаючись суто на рекурентну структуру [26]. Завдяки цьому моделі можуть з більшим успіхом враховувати довгострокові залежності у цінових рядах та реагувати на раптові ринкові шоки.

Об'єднання ринкових даних із новинним та соціальним середовищем: Використання нейронних мереж дає можливість інтегрувати різнорідні дані (мульти-модальні дані), включаючи:

Часові ряди цін та обсягів торгів; Технічні індикатори (сковзні середні, індекси відносної сили, MACD тощо); Новинний та соціальний сентимент, отриманий за допомогою методів NLP; Інформацію про ключові події, регуляторні зміни чи заходи безпеки, які може бути закодовано за допомогою векторних репрезентацій.

За допомогою глибоких нейронних мереж інвестори та трейдери можуть отримувати більш надійні сигнали для прийняття торговельних рішень. Попередні дослідження демонструють, що використання LSTM- та трансформер-мереж здатне підвищити точність прогнозу цін біткоїна, етереуму та інших популярних токенів. Це особливо корисно для алгоритмічного трейдингу та формування портфелів із керованим ризиком. Крім того, інтеграція нейронних моделей із сигналами новинного середовища (сентимент, події) допомагає адаптуватися до швидких змін стану ринку.

Висновки до розділу 1

У цьому розділі було здійснено ґрунтовний огляд досліджень впливу новинних повідомлень на динаміку криптовалютного ринку, а також

розглянуто основні методи та підходи до аналізу цього впливу. Результати попередніх досліджень свідчать про те, що новини, залежно від свого типу, змісту, тональності та достовірності, можуть суттєво змінювати настрої інвесторів, впливаючи на попит і пропозицію активів, рівень волатильності й формування ринкових трендів. Важливими є як макроекономічні та регуляторні новини, так і повідомлення соціального та технологічного характеру, оскільки усі ці чинники можуть формувати інформаційне тло, котре безпосередньо позначається на поведінці інвесторів та динаміці цін криптовалют. Розглянуті приклади демонструють, що криптовалютний ринок є більш чутливим і волатильним порівняно з традиційними фінансовими ринками, а також характеризується безперервною торгівлею та активною роллю соціальних мереж у поширенні інформації.

Огляд методів аналізу впливу новин охопив як класичні економетричні підходи та регресійні моделі, так і сучасні інструменти машинного навчання та обробки природної мови (NLP). Класичні методи забезпечують прозорість і простоту, проте мають обмежену здатність враховувати складні нелінійні та контекстуальні залежності. У свою чергу, машинне навчання та нейронні мережі, включно з LSTM, CNN і трансформерами, дають змогу моделювати більш складну структуру даних і поєднувати ринкові сигнали з результатами аналізу текстових повідомлень, їхньої тональності та контексту. Використання NLP відіграє ключову роль у розумінні семантики та настрою новин, що у поєднанні з часовими рядами та технічними індикаторами дозволяє підвищувати точність прогнозування динаміки криптовалют.

Разом з тим, у літературі спостерігаються певні проблеми та прогалини. Зокрема, не повною мірою вирішені питання щодо інтеграції різних типів новин та джерел інформації, недостатньо вивчено вплив фальшивих новин та проблеми інформаційної асиметрії. Дотепер складні моделі нерідко залишаються «чорними скриньками», що ускладнює інтерпретацію результатів і прийняття зважених рішень. Отже, виникає необхідність у розробці нових підходів, які б об'єднували сучасні інструменти NLP, здатність

моделювати мульти-модальні дані й можливість інтерпретації результатів. Такий підхід дозволить не лише підвищити точність прогнозів, а й сприятиме кращому розумінню того, які саме новини та чому впливають на ринок криптовалют, відкриваючи нові перспективи для ефективного управління ризиками та формування торговельних стратегій.

РОЗДІЛ 2 МЕТОДИ ТА ПІДХОДИ ДО ПРОГНОЗУВАННЯ ДИНАМІКИ ЦІН КРИПТОВАЛЮТ

2.1 Постановка задачі

Формальна постановка задачі прогнозування курсу валюти з урахуванням екзогенних та ендогенних змінних виглядає наступним чином:

Вхідні дані:

1. Ендогенні змінні — часовий ряд $y_{t=1}^T$, де $y_t \in \mathbb{R}$ — значення курсу валюти в момент часу t . Ендогенні змінні формуються виключно на основі минулих значень курсу валюти.

2. Екзогенні змінні — набір часових рядів $X_{t=1}^{(j)T}$, де $X_t^{(j)} \in \mathbb{R}$, $j = 1, 2, \dots, m$. Кожна змінна $X_t^{(j)}$ є значенням зовнішніх факторів (наприклад, новинних індикаторів, макроекономічних показників, обсягів торгів, індексів тощо) в момент часу t . Загальний вектор екзогенних змінних позначається як $X_t \in \mathbb{R}^m$.

Цільовою змінною є прогнозоване значення курсу валюти \hat{y}_{t+h} на момент часу $t+h$, де h — горизонт прогнозування.

Задача прогнозування полягає у побудові моделі $f(\cdot)$, що апроксимує функціональну залежність між ендогенними та екзогенними змінними:

$$\hat{y}_{t+h} = f(y_t, y_{t-1}, \dots, y_{t-k+1}, X_t, X_{t-1}, \dots, X_{t-k+1}; \theta), \quad (2.1)$$

де: k — довжина історичного вікна для побудови прогнозу,

θ — параметри моделі, які оцінюються в процесі навчання,

y_{t-i} — ендогенні змінні, що відображають динаміку курсу валюти,

X_{t-i} — екзогенні змінні, що враховують вплив зовнішніх факторів.

Критерій оптимізації:

Оптимальні параметри θ^* визначаються шляхом мінімізації функції втрат $L(\cdot)$, яка враховує відхилення прогнозованого значення \hat{y}_{t+h} від фактичного y_{t+h} :

$$\theta^* = \operatorname{argmin}_{\theta} (1/N) \sum_{i=1}^N L(y_{t+h}^{(i)}, \hat{y}_{t+h}^{(i)}), \quad (2.2)$$

де: N — кількість спостережень у тренувальному наборі,

$L(\cdot)$ — функція втрат.

2.2 Існуючі підходи вирішення задачі

2.2.1 ARIMA

ARIMA (Автогерегресивна інтегрована модель з ковзним середнім) є класичною економетричною моделлю, яка використовується для аналізу та прогнозування часових рядів. Вона моделює залежність поточного значення ряду від його минулих значень та помилок прогнозу [27]. Модель ARIMA ефективна для короткострокових прогнозів при наявності сталих тенденцій і сезонних коливань в даних. Проте для її успішного застосування необхідно, щоб дані були стаціонарними, тобто без явних трендів чи сезонності.

Модель ARIMA описується трьома основними параметрами:

p — порядок автогерегресії (AR), який визначає кількість попередніх значень часового ряду, які використовуються для прогнозу.

d — ступінь інтеграції (I), який визначає кількість різниць, необхідних для досягнення стаціонарності ряду.

q — порядок ковзного середнього (MA), який визначає кількість попередніх залишків (помилки прогнозу), що використовуються в моделі [28].

Математично модель ARIMA представлена як:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}, (2.3)$$

де: Y_t — значення часового ряду на момент часу t ,

$Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$ — попередні значення часового ряду,

ϵ_t — залишок (помилка) на момент часу t ,

$\phi_1, \phi_2, \dots, \phi_p$ — коефіцієнти автогорегресії,

$\theta_1, \theta_2, \dots, \theta_q$ — коефіцієнти ковзного середнього.

Процес побудови моделі ARIMA

Перетворення часового ряду. Модель ARIMA вимагає, щоб вхідний часовий ряд був стаціонарним. Якщо ряд не є стаціонарним (наприклад, має тренд або сезонні коливання), застосовують диференціацію, тобто обчислюють різниці між послідовними значеннями.

Вибір параметрів (p , d , q). Параметри моделі ARIMA визначаються за допомогою методів, таких як аналіз автокореляційної функції (ACF) та часткової автокореляційної функції (PACF), або ж шляхом оптимізації (наприклад, за допомогою методу максимального правдоподібності).

Оцінка параметрів. Після вибору параметрів ARIMA, модель оцінює відповідні коефіцієнти ϕ_1, \dots, ϕ_p та $\theta_1, \dots, \theta_q$ для автогорегресії та ковзного середнього[29].

2.2.2 Рекурентні нейронні мережі

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) є класом архітектур штучних нейронних мереж, які спеціально розроблені для роботи з послідовними даними, такими як текст, аудіо, відео або часові ряди [30]. На відміну від звичайних нейронних мереж, які обробляють вхідні дані як

незалежні, RNN враховують контекст попередніх елементів у послідовності, що робить їх особливо корисними для задач, де порядок даних є важливим.

Особливості RNN: Головною відмінністю рекурентних мереж є використання механізму зворотного зв'язку, який дозволяє інформації циркулювати в межах мережі. Це досягається через додавання рекурентного з'єднання, завдяки якому вихід прихованого шару на одному часовому кроці використовується як вхід для цього ж шару на наступному кроці. Математично це можна описати як:

$$h_t = f(W_h h_{t-1} + W_x x_t + b), \quad (2.4)$$

де: h_t — стан прихованого шару на поточному часовому кроці t ;

h_{t-1} — стан прихованого шару на попередньому кроці;

x_t — вхідні дані на часовому кроці t ;

W_h і W_x — матриці ваг для попереднього стану та поточного входу відповідно;

b — вектор зміщення;

f — нелінійна активаційна функція (зазвичай \tanh або ReLU).

Цей підхід дозволяє мережі зберігати контекст попередніх кроків і враховувати його під час обробки нових даних.

Навчання RNN: Навчання рекурентних мереж здійснюється за допомогою алгоритму зворотного поширення помилки через час (Backpropagation Through Time, BPTT). Цей метод є модифікацією стандартного алгоритму зворотного поширення, який враховує вплив кожного стану на наступні часові кроки.

Однак навчання RNN має кілька проблем, серед яких виділяють:

Витік градієнта: значення градієнта стає надто малим, що заважає оновленню ваг на ранніх кроках послідовності.

Вибух градієнта: градієнт може ставати надто великим, що призводить до нестабільності навчання.

Для вирішення цих проблем використовуються спеціальні архітектури, такі як LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) [31], які застосовують механізми керування потоком інформації через додаткові гейти.

2.2.3 LSTM

Архітектура LSTM.

LSTM є одним із найпопулярніших варіантів RNN, завдяки здатності ефективно обробляти довгі послідовності даних. Основою LSTM є блок, який складається з кількох гейтів:

Гейт забування (forget gate): визначає, яку частину інформації потрібно видалити зі стану[32].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2.5)$$

де: f_t — вектор гейта забування на часовому кроці t ;

σ — сигмоїдна функція активації;

W_f — матриця ваг для гейта забування;

h_{t-1} — вихід попереднього прихованого стану;

x_t — вхідний вектор на часовому кроці t ;

b_f — вектор зміщення для гейта забування. Гейт введення (input gate): вирішує, яку нову інформацію додати до стану.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2.6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (2.7)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (2.8)$$

де: i_t — вектор гейта введення;

\tilde{C}_t — вектор нової інформації для стану комірки;

C_t — оновлений стан комірки;

W_i, W_c — матриці ваг для гейта введення та нового стану;

b_i, b_c — вектори зміщень;

\odot — покомпонентне множення.

Гейт виводу (output gate): обирає, яку частину інформації передати на вихід.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (2.9)$$

$$h_t = o_t \odot \tanh(C_t), \quad (2.10)$$

де: o_t — вектор гейта виводу;

h_t — вихід прихованого стану на часовому кроці t ;

W_o — матриця ваг для гейта виводу;

b_o — вектор зміщення.

Ця структура(рис. 1) дозволяє LSTM зберігати важливу інформацію протягом тривалих проміжків часу та ігнорувати несуттєву.

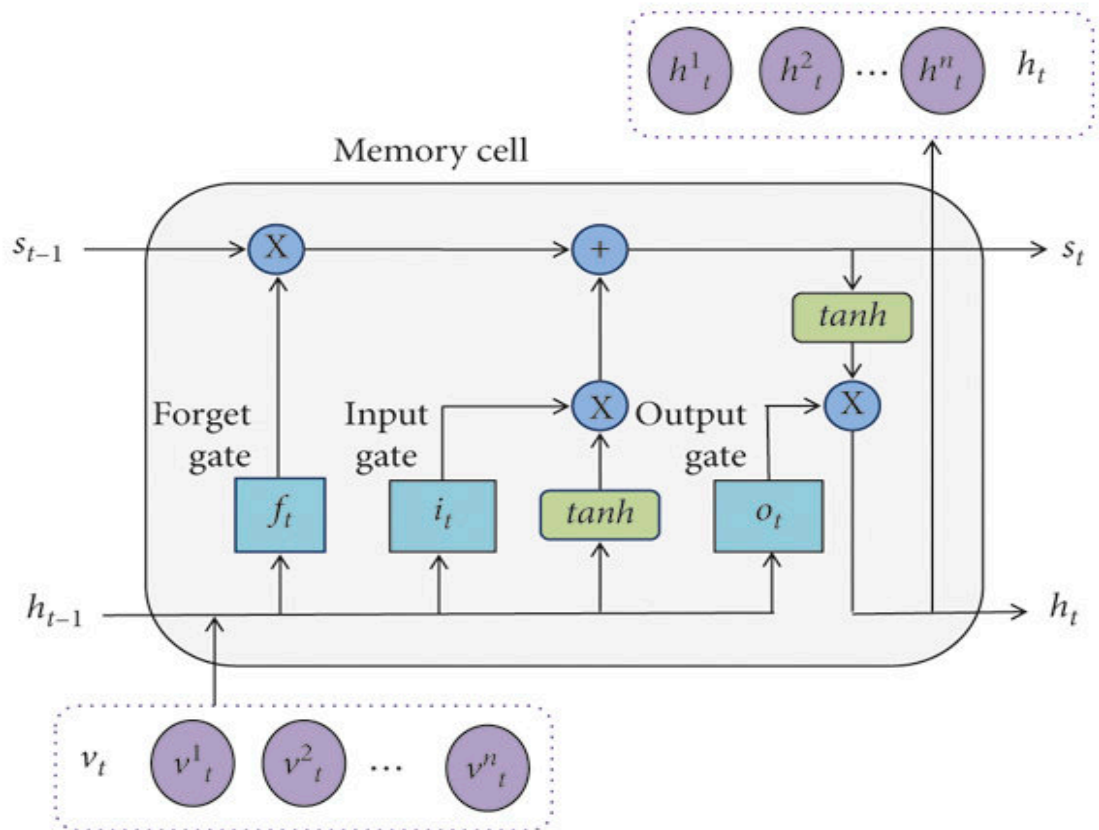


Рисунок 1 – Архітектура LSTM

2.2.4 GRU

GRU є іншим типом рекурентних нейронних мереж, схожих на LSTM, але з меншою кількістю параметрів і складніших обчислень [33]. Вона має менше блоків для зберігання та оновлення інформації, що дозволяє значно зменшити час тренування моделі, зберігаючи при цьому ефективність.

GRU також може використовуватися для прогнозування на основі новин та інших екзогенних змінних, обробляючи тимчасові залежності в даних з меншою складністю порівняно з LSTM.

GRU має два основних компоненти: оновлювальний затвор та скидний затвор.

Оновлювальний затвор (Update Gate). Оновлювальний затвор контролює, яка частина інформації з попереднього стану буде передана в поточний стан. Якщо затвор приймає значення близьке до 1, це означає, що більшість попередньої інформації залишається, а якщо близьке до 0 — більша частина інформації забувається.

Скидний затвор (Reset Gate). Скидний затвор визначає, яку частину попереднього стану потрібно "забути" при обчисленні нового стану. Якщо значення цього затвору близьке до 1, то попередній стан не має великого впливу на новий стан, а якщо близьке до 0 — попередній стан майже повністю забувається.

Для кожного часу t , стан GRU можна обчислити за наступними формулам:

1. Оновлювальний затвор:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (2.11)$$

де: z_t — вектор оновлювального затвору на час t ,

W_z — матриця ваг для оновлювального затвору,

h_{t-1} — попередній стан,

x_t — вхідні дані на час t ,

σ — сигмоїдальна функція активації,

b_z — зміщення для оновлювального затвору.

2. Скидний затвор:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \quad (2.12)$$

де: r_t — вектор скидного затвору на час t ,

W_r — матриця ваг для скидного затвору,

b_r — зміщення для скидного затвору.

3. Обчислення нової кандидатної пам'яті:

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h), \quad (2.13)$$

де: \tilde{h}_t — кандидатний вектор для нового стану,

W_h — матриця ваг для обчислення нової пам'яті,

b_h — зміщення для нової пам'яті.

4. Оновлення стану:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t, \quad (2.14)$$

де: h_t — новий стан на час t ,

z_t — вектор оновлювального затвору,

h_{t-1} — попередній стан,

\tilde{h}_t — кандидатний вектор пам'яті.

Опис роботи:

1. На кожному кроці часу GRU отримує на вхід x_t та попередній стан h_{t-1} .
2. Оновлювальний затвор визначає, яка частина попереднього стану h_{t-1} буде збережена в новому стані.
3. Скидний затвор визначає, яку частину попереднього стану потрібно забути.
4. Кандидатний стан обчислюється на основі поточного вхідного значення та попереднього стану, враховуючи параметри скидного затвору.
5. Новий стан обчислюється як комбінація попереднього стану та нового кандидату, зваженого на основі значень оновлювального затвору.

На рис. 2 зображено загальну архітектуру GRU.

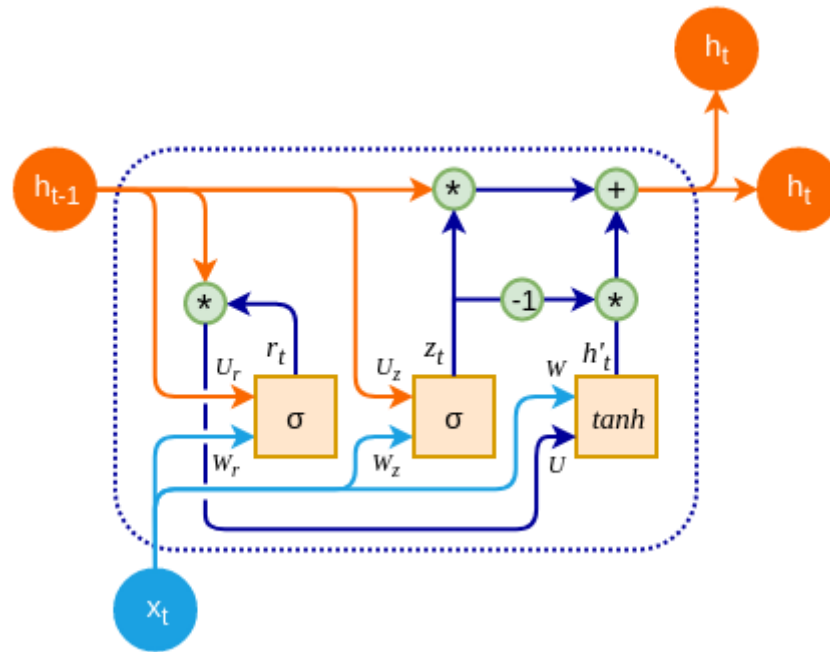


Рисунок 2 – Архітектура GRU

2.3 Опис методу для кодування назв та опису новин

Для кодування наративної роботи кожної з новин використовувалась трансформерна архітектура all-MiniLM-L6-v2.

all-MiniLM-L6-v2 — це одна з моделей трансформерів, розроблена компанією Microsoft, яка входить до родини моделей на основі BERT (Bidirectional Encoder Representations from Transformers). Ця модель спеціалізується на задачах обробки тексту, таких як класифікація тексту, порівняння схожості текстів, та отримання векторних представлень (ембедінгів) текстів[34]. Вона є меншою за розміром порівняно з іншими великими моделями, такими як BERT, але має високу ефективність, що дозволяє застосовувати її для реальних задач, де важлива швидкість обробки.

Модель all-MiniLM-L6-v2 була оптимізована для отримання компактних векторних представлень текстів, які зберігають більшість їх семантичної

інформації. Вона використовує трансформери і має 6 шарів (L6) в архітектурі, що забезпечує хорошу балансованість між швидкістю та якістю.

Як працює all-MiniLM-L6-v2 для перетворення назв і описів новин:

Токенізація тексту: Першим кроком є поділ тексту на токени, що є основними одиницями обробки для трансформерів. У випадку з новинами токенизуються як назви новин, так і їх описи. Токенізація дозволяє моделі працювати з окремими словами або навіть частинами слів (наприклад, за допомогою субсловних одиниць).

Векторизація (отримання ембеддінгів): Після токенизації, кожен токен перетворюється на векторну репрезентацію, що має фіксовану довжину (наприклад, 256 або 512 елементів). Ці вектори є числовими уявленнями слів, що зберігають семантику. Модель all-MiniLM-L6-v2 генерує контекстно-залежні ембеддінги, тобто, значення кожного токена визначається в залежності від оточуючих його слів.

Агрегація ембеддінгів: Для створення векторного представлення цілих документів (наприклад, заголовків та описів новин) модель виконує агрегацію векторів окремих токенів. Зазвичай це робиться за допомогою простих методів, таких як середнє або максимальне значення по всіх токенах, або за допомогою спеціальних методів агрегації, що беруть до уваги позицію токенів.

Отримання фіксованого вектору: На виході модель надає фіксоване числове представлення для кожної новини (заголовок + опис), яке можна використовувати для подальших задач, таких як порівняння схожості між новинами, класифікація чи кластеризація новин за змістом.

2.4 Фінальна архітектура системи

На рисунку 5 представлено схему побудови кінцевої архітектури системи прогнозування ціни криптовалют на основі текстових та кількісних даних.

Система складається з фази підготовки та фази застосування. На етапі підготовки відбувається інтеграція та трансформація [35] різнорідних даних: нарративної інформації з новин (заголовки та описи), ендогенних ринкових ознак (історичні ціни, обсяги торгів) та екзогенних чинників (наприклад, тональність новин чи соціальні сигнали). Модель all-MiniLM-L6-v2 перетворює нарративні дані у векторні уявлення, які зберігають семантику та контекст. Ендогенні та екзогенні ознаки, які мають числову природу, нормалізуються та агрегуються для подальшої обробки.

На етапі навчання рекурентна нейронна мережа (наприклад, архітектура на основі LSTM або GRU) приймає на вхід послідовність попередніх періодів, де кожен період представлений вектором ознак: векторизовані новинні тексти від трансформерної моделі, числові ринкові індикатори та екзогенні фактори. Мережа навчається розпізнавати приховані патерни та залежності між вхідними даними, враховуючи часову динаміку та контекст новинного фону. Таким чином, формується модель, здатна інтегрувати вербальну та кількісну інформацію, щоб прогнозувати майбутню вартість активу.

На фазі застосування нові дані (нарративні повідомлення, ринкові характеристики та екзогенні чинники) подаються на вхід навченої системи. Модель all-MiniLM-L6-v2 перетворює отриманий текст у вектори, а рекурентна мережа, враховуючи історичні патерни та нові сигнали, генерує прогноз ціни для наступного моменту часу. Якщо надходять нові дані про ринкові умови або релевантні новини, модель можна оновити (донавчити), що покращить точність передбачення внаслідок адаптації до актуального інформаційного середовища та поточних ринкових тенденцій.

Отже, запропонована архітектура системи забезпечує комплексний підхід до прогнозування, поєднуючи глибинні моделі обробки природної мови з методами моделювання часових рядів. Це дозволяє отримати прогнози з урахуванням не лише історичних числових даних, але й контексту та змісту новинних повідомлень, підвищуючи якість та інформативність фінальних результатів.

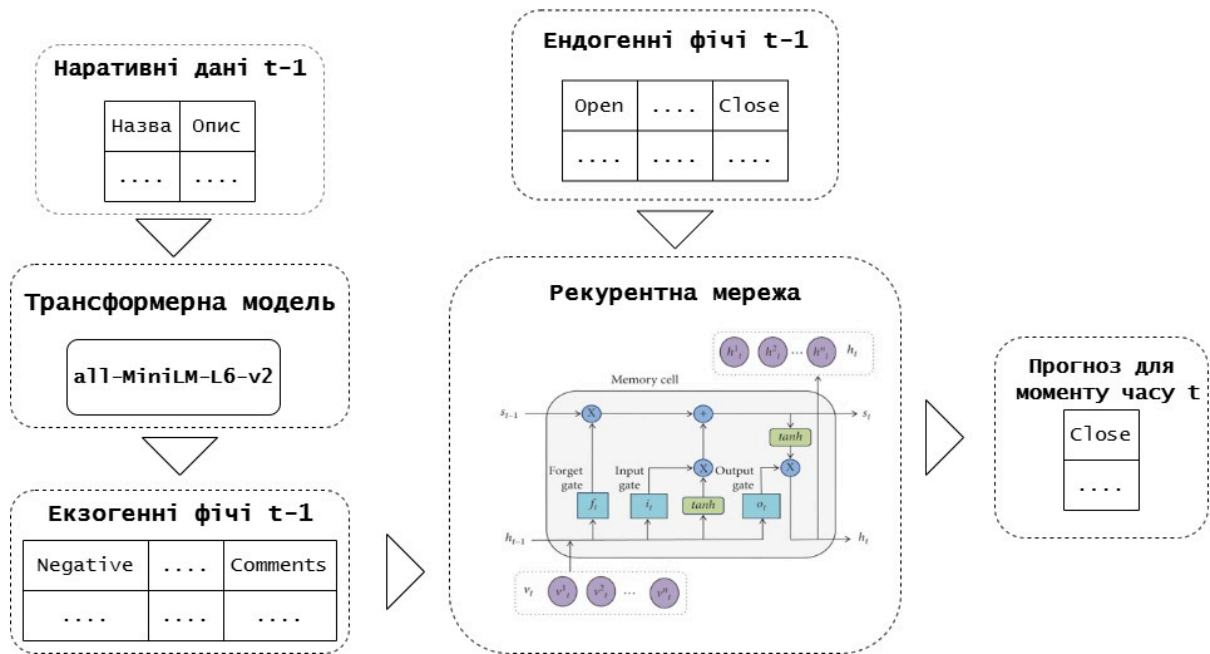


Рисунок 3 – Послідовність генерації прогнозу

2.5 Метрики задачі авторегресії

MAE (Mean Absolute Error) - середнє абсолютне відхилення. Це середнє значення абсолютних різниць між прогнозованими і фактичними значеннями.

Формула для розрахунку MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.15)$$

де: y_i - фактичне значення,

\hat{y}_i - прогнозоване значення,

n - кількість спостережень.

MSE (Mean Squared Error) - середня квадратична похибка. Це середнє значення квадратів різниць між фактичними і прогнозованими значеннями. Формула для розрахунку MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.16)$$

MSE надає більше значення більшим відхиленням і є корисним для оцінки точності моделей.

RMSE (Root Mean Squared Error) - корінь середньої квадратичної похибки. Це квадратний корінь з MSE. Формула для розрахунку RMSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (2.17)$$

RMSE дозволяє виміряти величину середньої похибки в тих самих одиницях вимірювання, що й сама величина даних.

MAPE (Mean Absolute Percentage Error) - середня абсолютна відносна похибка. Це середнє значення відносних абсолютних похибок між прогнозованими і фактичними значеннями. Формула для розрахунку MAPE:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (2.18)$$

MAPE виражає точність моделі як відсоток і дозволяє порівнювати прогнози на різних наборах даних.

R² (Coefficient of Determination) - коефіцієнт детермінації. Це метрика, що вимірює, яка частина варіації залежної змінної пояснюється моделлю. Формула для розрахунку R²:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (2.19)$$

де \bar{y} - середнє фактичне значення. R^2 варіюється від 0 до 1, де значення 1 означає, що модель повністю пояснює варіацію в даних, а 0 - що модель не пояснює жодної варіації.

Метрики MAE, MSE та RMSE є чутливими до масштабу даних, тобто їх значення залежать від одиниць вимірювання. Вони добре підходять для порівняння моделей, які оцінюються на одному й тому ж наборі даних, але стають менш ефективними при аналізі наборів із різними масштабами. MAE менш схильна до впливу аномальних значень порівняно з MSE та RMSE, оскільки останні підсилюють ефект великих похибок через піднесення до квадрату. RMSE має перевагу над MSE завдяки своїй шкалі, яка відповідає шкалі вихідних даних, що спрощує його інтерпретацію[36].

Метрика MAPE, на відміну від MAE, виражається у відсотках, тому не залежить від масштабу даних. Це робить її корисною для порівняння моделей на наборах даних із різними одиницями вимірювання. Однак MAPE має певні недоліки: вона стає невизначеною, коли $Y_t = 0$ $Y_t = 0$, і показує надмірно великі значення, якщо Y_t близький до нуля. Крім того, вона залишається вразливою до впливу екстремальних значень. Щоб уникнути цих проблем, використовується WMAPE, яка нормалізує похибки за допомогою суми фактичних значень, що дозволяє зменшити вплив цих недоліків.

Висновки до розділу 2

У цьому розділі було здійснено ґрунтовний огляд основних методів і підходів, застосовуваних до прогнозування динаміки цін криптовалют з урахуванням внутрішніх (ендогенних) та зовнішніх (екзогенних) чинників. Розглянуто формальну постановку задачі, яка дозволяє чітко визначити цільову змінну, обрати оптимальний горизонт прогнозування та підібрати

релевантні ознаки. Традиційні економетричні моделі, такі як ARIMA, забезпечують базову основу для аналізу часових рядів, проте мають суттєві обмеження в умовах високої нестабільності та нелінійності криптовалютного ринку. Рекурентні нейронні мережі, зокрема архітектури LSTM та GRU, запропонували більш ефективне моделювання складних залежностей у даних завдяки здатності утримувати інформацію протягом довгих часових інтервалів. Цей підхід став особливо корисним у ситуаціях, де ринкові параметри змінюються під впливом інформаційного фону. Використання трансформерної моделі all-MiniLM-L6-v2 для кодування новин у векторні представлення дало змогу інтегрувати значення текстових повідомлень безпосередньо в процес прогнозування цін, підвищуючи рівень інформативності та враховуючи семантику новинного середовища.

Застосування відповідних метрик оцінювання, таких як MAE, MSE, RMSE, MAPE та R^2 , дозволяє кількісно виміряти точність прогнозів, виявити сильні та слабкі сторони обраних моделей і оптимізувати їх для досягнення кращих результатів. Таким чином, розглянуті підходи та методи створюють надійну основу для поєднання часових рядів, новинних повідомлень та сучасних інструментів машинного навчання, що формує передумови для розробки гнучких, чутливих до контексту і більш точних прогнозних моделей у подальших дослідженнях.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ЕКСПЕРЕМЕНТІВ

У цьому розділі наведено опис вхідних даних, які використовувалися для навчання моделей, а також детальний опис самих моделей, параметрів найкращих з них та результатів їх застосування. Крім того, буде представлено опис модулів програмного продукту, розробленого в ході виконання дослідження.

3.1 Аналіз набору даних

3.1.1 Історичні дані Bitcoin

Для дослідження розробленого підходу використовувався датасет з історичним курсом Bitcoin з 2017-09-23 по 2024-11-30, датасет має наступні параметри:

Timestamp. Це тимчасова мітка, яка відображає момент часу, коли була зафіксована інформація про торг. Зазвичай цей стовпець містить дані у вигляді дати та часу, наприклад, "2023-09-23 00:00:00". Цей стовпець допомагає визначити часову послідовність даних і є ключовим для аналізу тимчасових рядів.

Open. Це ціна, за якою інструмент (наприклад, акція або криптовалюта) відкривається на початку певного торгового періоду. Цей стовпець вказує на першу ціну, за якою була здійснена угода під час певного інтервалу часу (наприклад, за одну хвилину, день або тиждень).

High. Це максимальна ціна, досягнута інструментом протягом певного торгового періоду. Вона відображає найвищу ціну, за якою здійснювались угоди на ринку.

Low. Це мінімальна ціна, досягнута інструментом під час певного торгового періоду. Це найнижча ціна, за якою здійснювались угоди на ринку в межах певного періоду.

Close. Це ціна, за якою інструмент закривається на кінець певного торгового періоду. Зазвичай це остання ціна угоди, яка була виконана в кінці періоду (наприклад, в кінці дня або на момент закриття ринку).

Volume. Це обсяг торгів, який вказує на кількість одиниць інструменту, що були продані чи куплені протягом певного періоду часу. Цей стовпець допомагає визначити активність на ринку і є важливим індикатором для аналізу трендів.

На рис. 4 зображена історія значень величини Close, яка прогнозується в дослідженні.

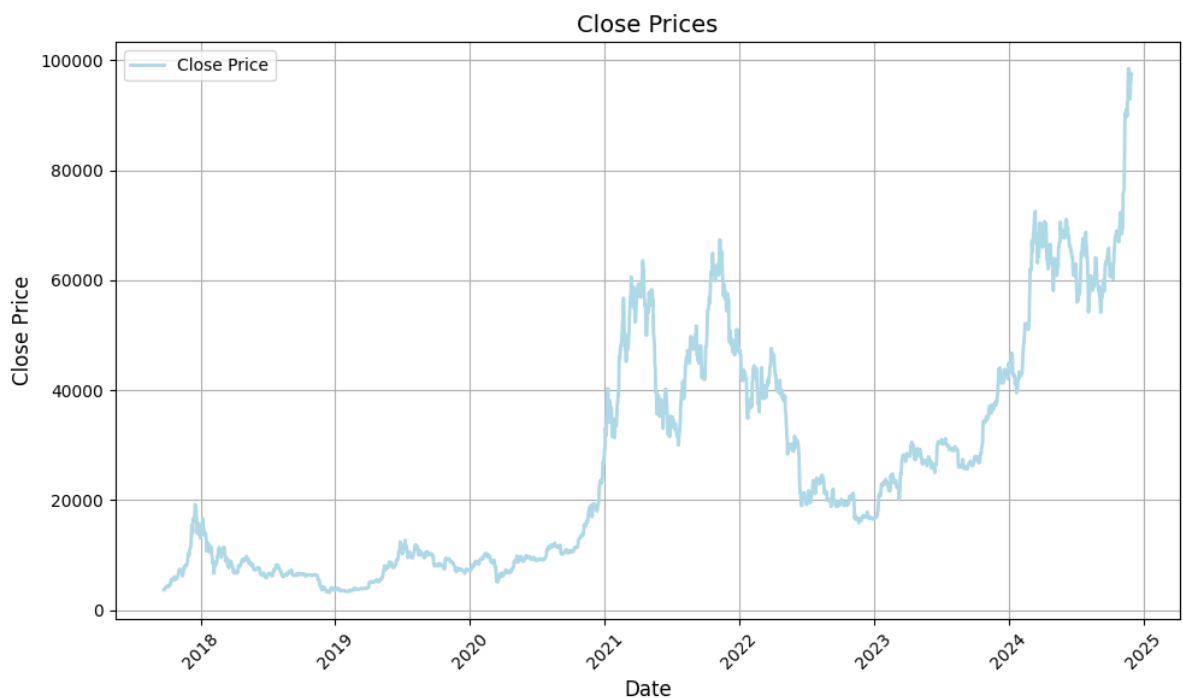


Рисунок 4 – Історичні значення Close

Приклад записів в датасеті зображений на рис. 5.

	Timestamp	Open	High	Low	Close	Volume
0	2012-01-01 10:01:00	4.58	4.58	4.58	4.58	0.000000
1	2012-01-01 10:02:00	4.58	4.58	4.58	4.58	0.000000
2	2012-01-01 10:03:00	4.58	4.58	4.58	4.58	0.000000
3	2012-01-01 10:04:00	4.58	4.58	4.58	4.58	0.000000
4	2012-01-01 10:05:00	4.58	4.58	4.58	4.58	0.000000
...
6721276	2024-11-29 23:57:00	97425.00	97425.00	97425.00	97425.00	0.011609
6721277	2024-11-29 23:58:00	97460.00	97497.00	97460.00	97497.00	0.000720
6721278	2024-11-29 23:59:00	97495.00	97515.00	97492.00	97512.00	1.696408
6721279	2024-11-30 00:00:00	97510.00	97510.00	97482.00	97482.00	0.007763
6721280	NaT	57854.00	57864.00	57835.00	57835.00	1.353466

6721281 rows × 6 columns

Рисунок 5 – Приклад записів в датасеті

3.1.2 датасет з крипто-новинами

Крім цього, маємо датасет з крипто-новинами, що має наступні характеристики:

Negative – флаг негативного тону або настрою новини.

Positive – флаг позитивного тону або настрою новини.

Important – флаг важливості новини.

Liked – флаг того що більшість взаємодій з новиною позитивна.

Disliked – флаг того що більшість взаємодій з новиною негативна.

Lol – флаг гумористичності новини

Toxic – флаг того що новина отримала сувору критику

Saved – флаг наявності великої кількості збережень новини користувачами.

Comments – флаг наявності великої кількості коментарів.

Також до даного датасету входять поля Label та Description, які були згодом закодовані трансформерною моделлю.

3.2 Результати

3.2.1 Навчання з екзогенними фічами, без фіч з новинами.

Спершу було проведено експерименти по навчанню трьох моделей з використанням фіч екзогенних 'Open', 'High', 'Low', 'Volume'.

SARIMA:

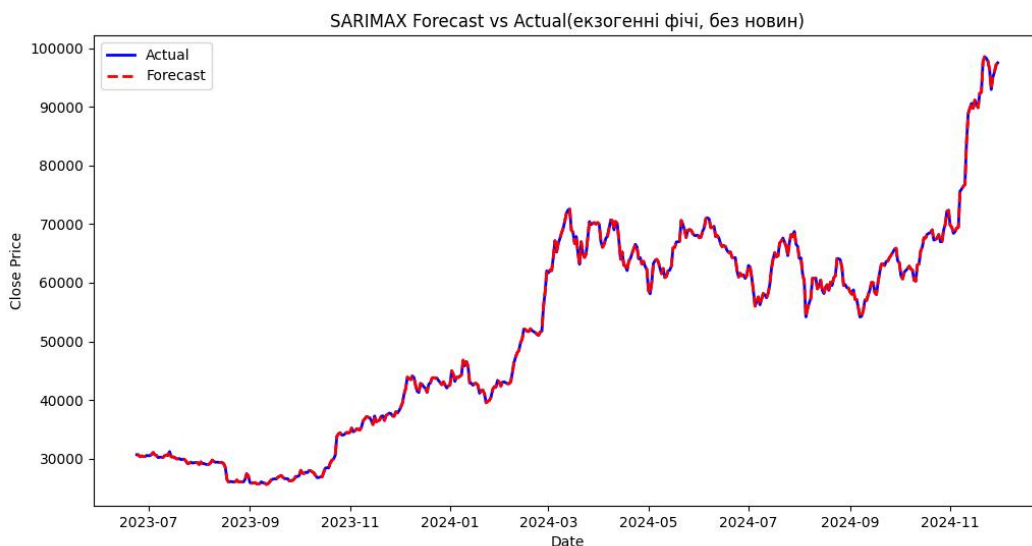


Рисунок 6 – Результати прогнозу моделі SARIMA на вибірці з екзогенними фічами, без новин

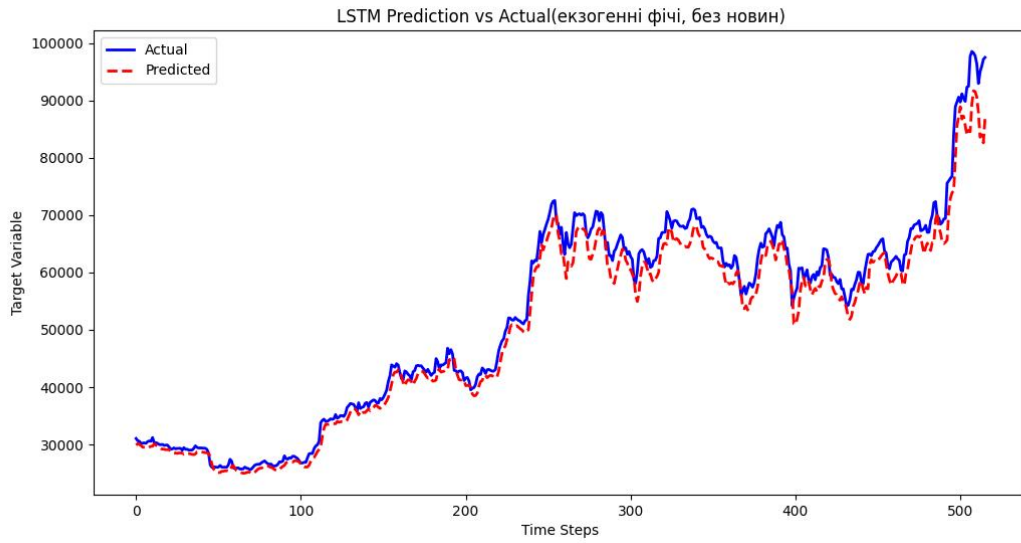
LSTM:

Рисунок 7 – Результати прогнозу моделі LSTM на вибірці з екзогенними фічами, без новин

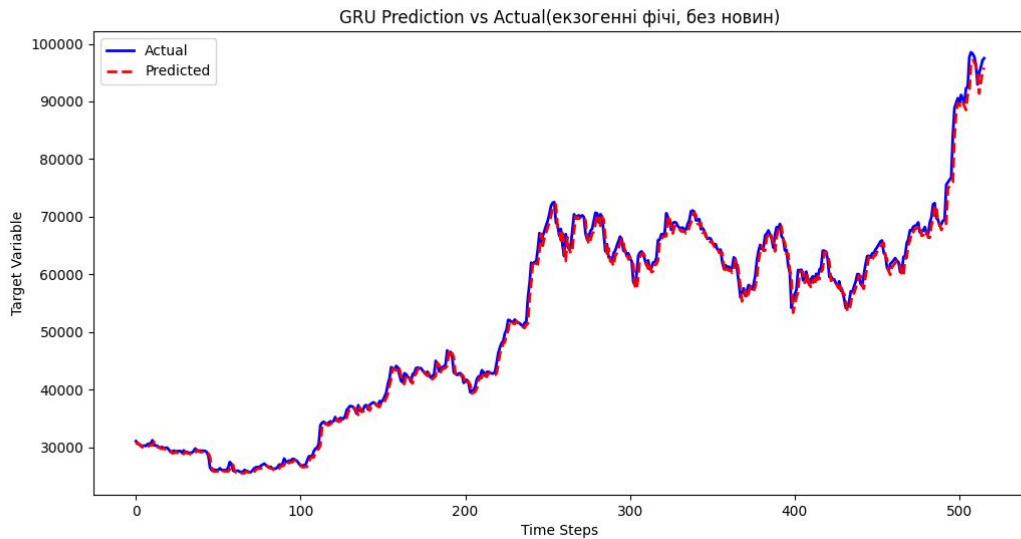
GRU:

Рисунок 8 – Результати прогнозу моделі GRU на вибірці з екзогенними фічами, без новин

3.2.2 Навчання з екзогенними фічами, разом з фічами з новинами.

Оскільки на одну і ту ж дату ми можемо мати кілька нових, в якості нових екзогенних фіч брались середні значення фіч по новинам на певну дату. Загалом це 'Open', 'High', 'Low', 'Volume', 'negative', 'positive', 'important', 'liked', 'disliked', 'lol', 'toxic', 'saved', 'comments'.

SARIMA:

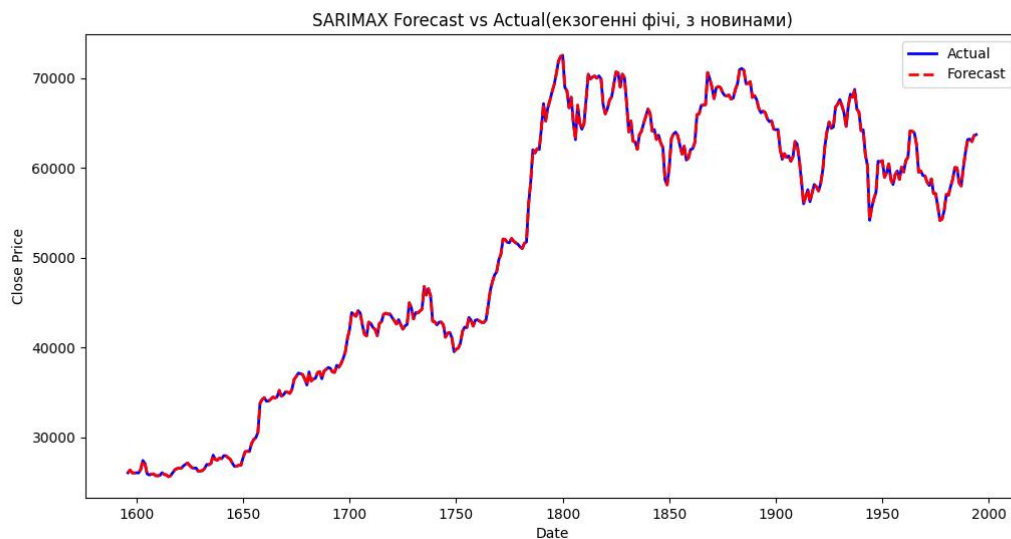


Рисунок 9 – Результати прогнозу моделі SARIMA на вибірці з екзогенними фічами, з новинами

LSTM:

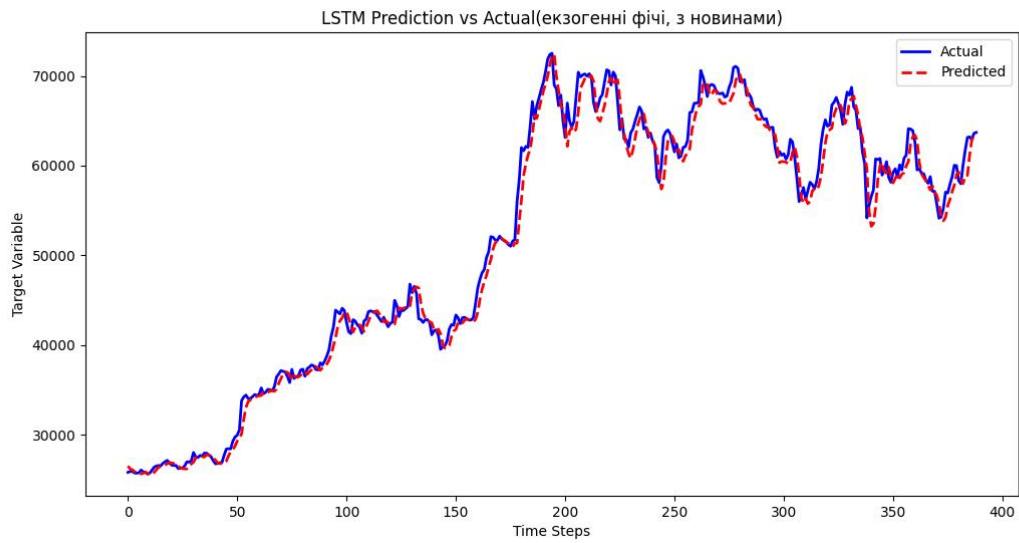


Рисунок 10 – Результати прогнозу моделі LSTM на вибірці з екзогенними фічами, з новинами

GRU:

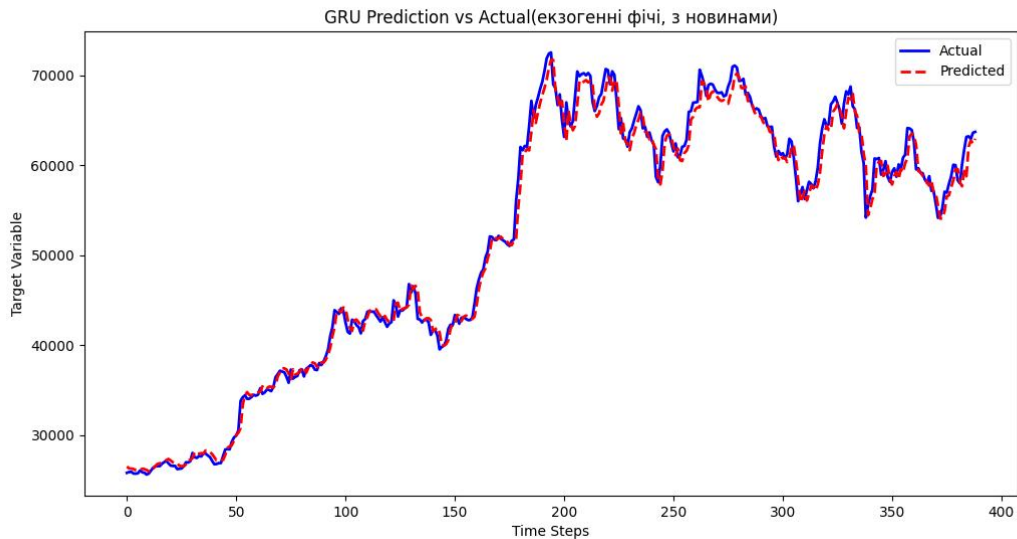


Рисунок 11 – Результати прогнозу моделі GRU на вибірці з екзогенними фічами, з новинами

В наступній таблиці записані результати прогнозів кожної з моделей, як можна побачити, фічі з крипто-новинами дали невеликий приріст для моделей SARIMAX та LSTM.

Таблиця 3.1 – Результати прогнозів на екзогенних фічах

<i>Model</i>	<i>MAE</i>	<i>MSE</i>	<i>RMSE</i>	<i>MAPE</i>	<i>R²</i>
SARIMAX (trade data only)	3,10212	59,06057	7,685088	0,004725	0,999999811
LSTM (trade data only)	2261,273	8932192	2988,677	3,993651	0,97119808
GRU (trade data only)	916,4118	1832765	1353,796	1,675784	0,994090237
SARIMAX (trade data and cryptonews)	2,685787	15,83402	3,979198	0,004601	0,999999929
LSTM (trade data and cryptonews)	1081,457	2319582	1523,017	2,007434	0,989006227
GRU (trade data and cryptonews)	920,7078	1662759	1289,48	1,709516	0,992119269

Можемо зробити висновок, що оскільки якість моделі з екзогенними фічами самої валюти надто велика, для оцінки впливу фіч з новинами, потрібно прибрати екзогенні фічі.

3.2.3 Навчання без екзогенних фіч та без фіч з новинами.

На даному етапі використовуємо тільки значення **Close**.

SARIMA:

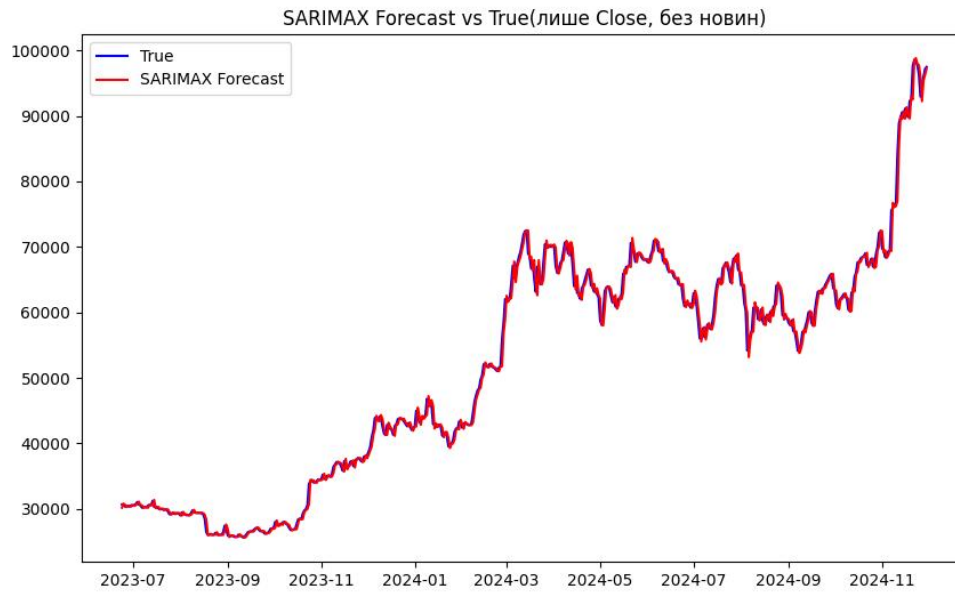


Рисунок 12 – Результати прогнозу моделі SARIMA на вибірці без екзогенних фіч та без новин

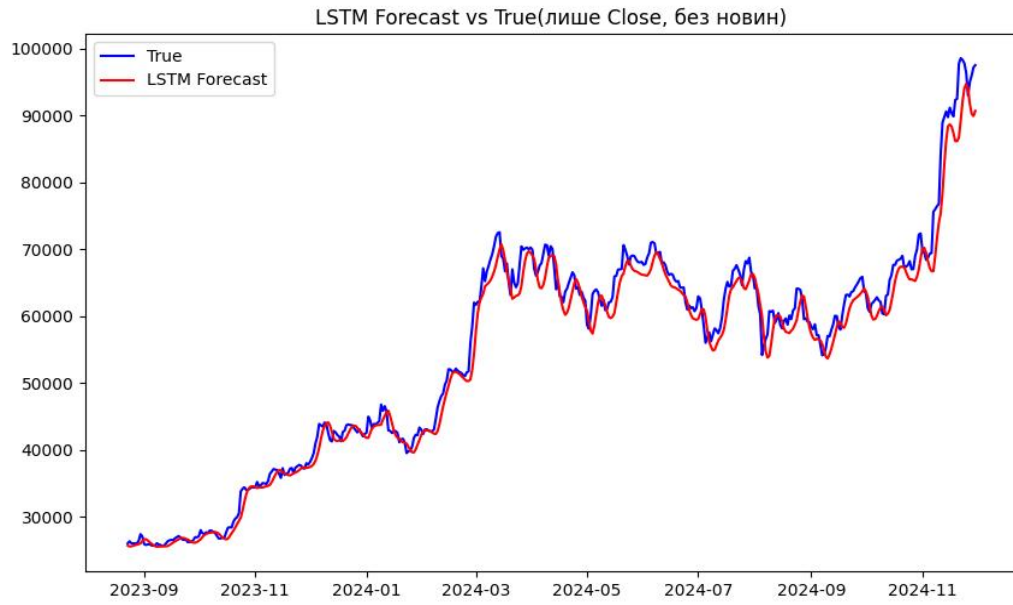
LSTM:

Рисунок 13 – Результати прогнозу моделі LSTM на вибірці без екзогенних фіч та без новин

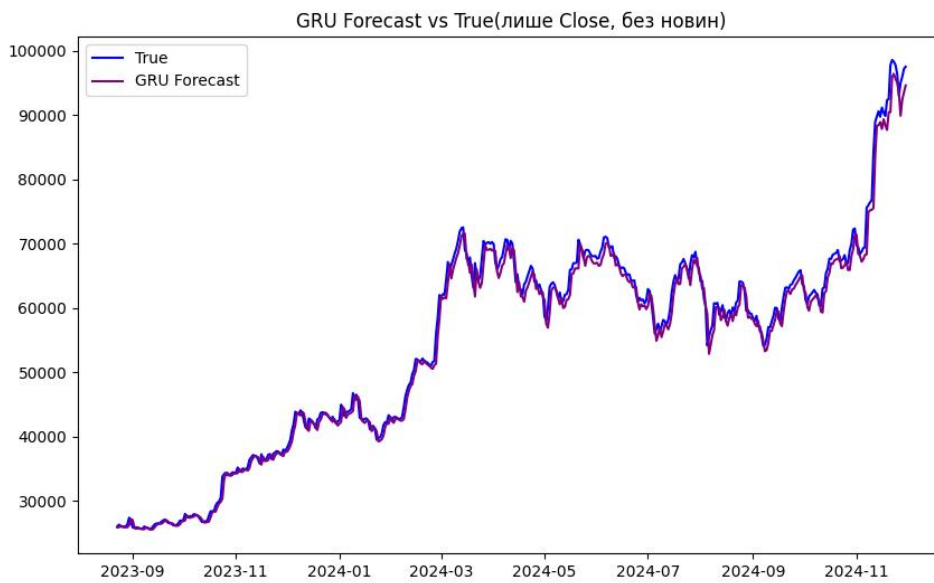
GRU:

Рисунок 14 – Результати прогнозу моделі GRU на вибірці без екзогенних фіч та без новин

3.2.4 Навчання без екзогенних фіч та з фічами з новинами.

SARIMA:

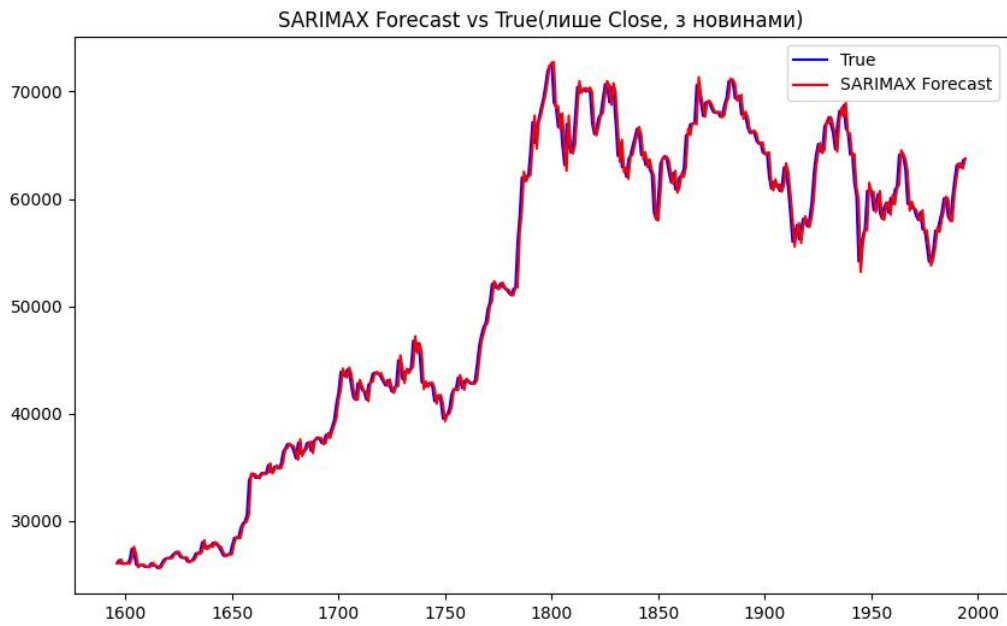


Рисунок 15 – Результати прогнозу моделі SARIMA на вибірці без екзогенних фіч та з новинами

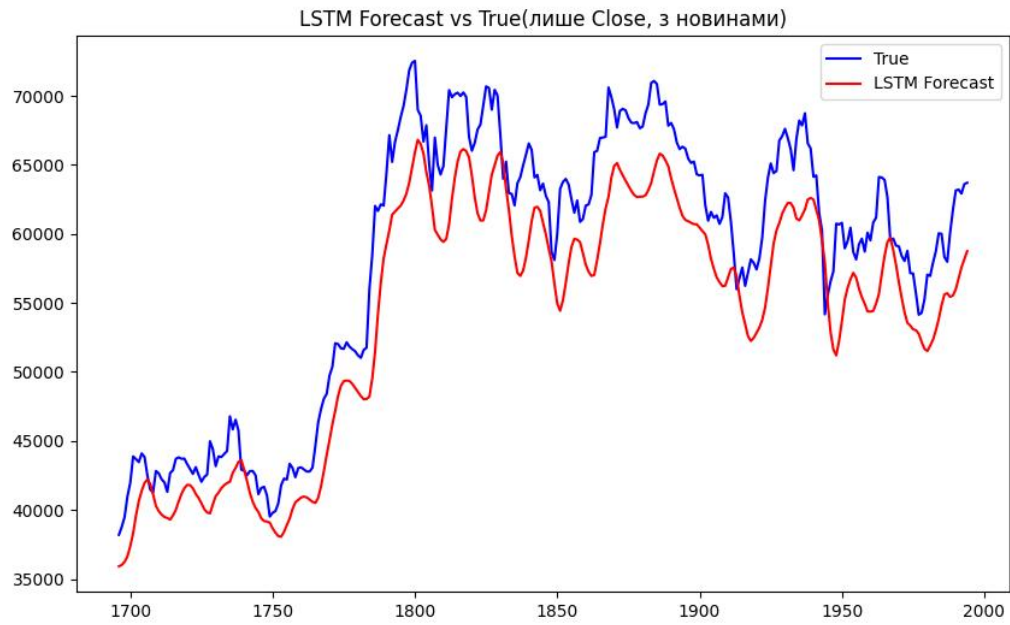
LSTM:

Рисунок 16 – Результати прогнозу моделі LSTM на вибірці без екзогенних фіч та з новинами

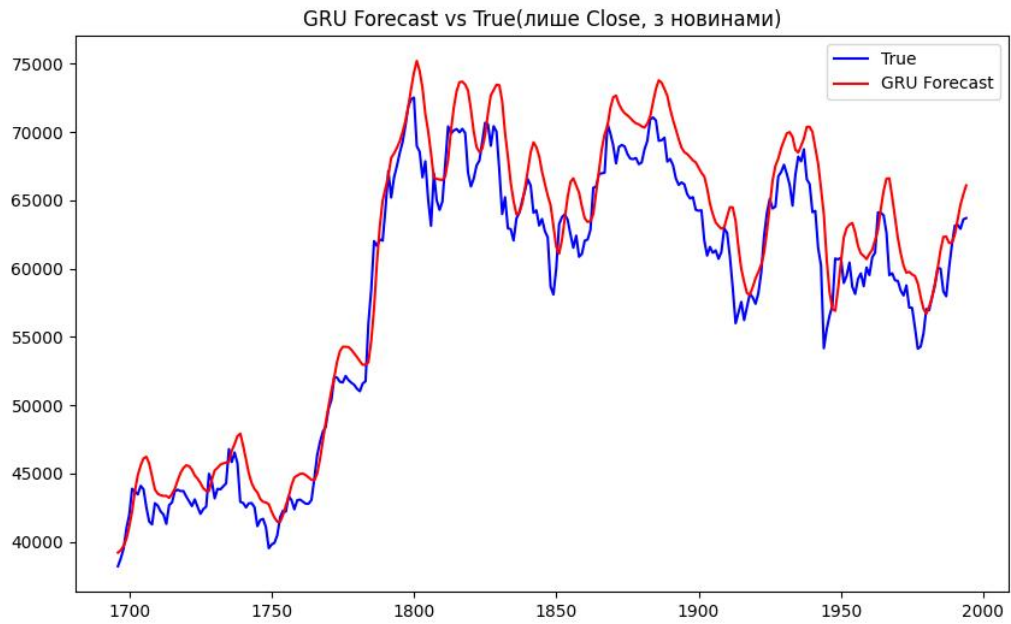
GRU:

Рисунок 17 – Результати прогнозу моделі GRU на вибірці без екзогенних фіч та з новинами

Маємо фінальну таблицю з результатами:

Таблиця 3.2 – Фінальна таблиця з результатами

Model	MAE	MSE	RMSE	MAPE	R ²
SARIMAX_trade_data_only	3,10212	59,06057	7,685088	0,004725	0,9999998
LSTM_trade_data_only	2261,273	8932192	2988,677	3,993651	0,9711981
GRU_trade_data_only	916,4118	1832765	1353,796	1,675784	0,9940902
SARIMAX_trade_data_and_cryptonews	2,685787	15,83402	3,979198	0,004601	0,9999999
LSTM_trade_data_and_cryptonews	1081,457	2319582	1523,017	2,007434	0,9890062
GRU_trade_data_and_cryptonews	920,7078	1662759	1289,48	1,709516	0,9921193
SARIMAX_only_Close	774,0159	1455444	1206,418	1,409356	0,9953488
LSTM_only_Close	1748,644	5957820	2440,865	0,029676	0,9788636
GRU_only_Close	1117,807	2508112	1583,702	0,019068	0,991102
SARIMAX_only_Close_and_cryptonews	517,589	208112,7	583,7949	1,114073	0,9971768
LSTM_only_Close_and_cryptonews	4116,181	21519893	4638,954	0,069494	0,7763467
GRU_only_Close_and_cryptonews	2336,677	8351201	2889,844	0,040302	0,9132071

В табл. 3.2 холодними кольорами відмічені найменші значення кожного стовбця, а гарячими – найбільші.

3.3 Платформа, мова програмування та бібліотеки

Для успішної розробки програмного продукту важливо ретельно вибрати найбільш підходяще середовище та інструменти для реалізації. Це вимагає врахування цілей та обмежень завдання, а також можливостей, які надає кожне конкретне середовище розробки. Для вирішення задач, пов'язаних з класифікацією, кластеризацією та іншими аспектами машинного навчання, оптимальним вибором є мова програмування Python. Python є об'єктно-орієнтованою мовою високого рівня, яка підтримує динамічну типізацію. Широка популярність Python у сфері машинного навчання зумовлена його високою ефективністю при обробці великих обсягів даних, а також наявністю

численних бібліотек, що значно спрощують розробку та реалізацію алгоритмів. У процесі створення цього програмного продукту була обрана версія Python 3.10.5.

Серед основних бібліотек, які використовувались в роботі, можна відзначити: Numpy, Pandas, Sklearn та Matplotlib. Numpy та Pandas використовуються для ефективної роботи з різноманітними даними, зокрема табличними та матричними структурами. Вони дозволяють оптимізувати обробку даних, знижуючи вимоги до пам'яті, завдяки чому програма працює швидше та з меншою витратою ресурсів. Бібліотека Sklearn надає доступ до багатьох попередньо реалізованих моделей машинного навчання, що значно полегшує процес їх застосування. Крім того, Sklearn дозволяє коригувати моделі та налаштовувати їх параметри відповідно до потреб конкретного завдання. Відкритий вихідний код цієї бібліотеки дає можливість вносити додаткові зміни для точнішої адаптації під специфічні вимоги. Бібліотека має BSD ліцензію, що дозволяє використовувати її на комерційній основі. Окрім зазначених бібліотек, були використані й інші допоміжні інструменти, які забезпечують виконання складних математичних операцій та сприяють оптимізації роботи програмного продукту.

Висновки до розділу 3

Для датасету з екзогенними фічами, результуючі моделі SARIMA та LSTM на основі підходу з використанням фіч з новинами, дали незначний приріст, який не говорить про ефективність підходу. Проте, модель SARIMAX з scrptonews без екзогенних фіч отримала приріст по метрикам квадратичної помилки майже в 2 рази, в порівнянні з її версією без крипто-новин. Можна зробити висновок, щоб підхід повністю робочий, проте потребує ретельного калібрування результуючих моделей, та дотримання певних топологічних

особливостей в структурі моделі. Можливо тому в даному випадку, для рекурентних мереж не спостерігалось підвищення ефективності.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартапи набувають усе більшої популярності, особливо у сфері технологій, де інновації та новітні рішення відкривають нові можливості. Стартапи — це підприємства або проекти, що зосереджені на розробці унікального продукту чи послуги, які засновники прагнуть впровадити на ринок. У більшості випадків такі проекти знаходяться на ранніх стадіях розвитку, не мають усталеної бізнес-моделі та значного капіталу для масштабування, тому часто фінансуються засновниками або залученими інвесторами.

Стартапи у сфері штучного інтелекту особливо затребувані, оскільки ці технології дозволяють вирішувати складні задачі інноваційними підходами. Наприклад, у фінансовій сфері штучний інтелект демонструє високі результати в аналізі великих обсягів даних, передбаченні ринкових тенденцій та автоматизації процесів прийняття рішень. Це відкриває нові можливості для аналізу та прогнозування фінансових ринків, зокрема криптовалют.

Розробка стартапу, спрямованого на аналіз впливу новин на ринок криптовалют за допомогою методів штучного інтелекту, може стати вагомим внеском у розвиток інноваційних рішень у сфері фінансових технологій. Такий проект дозволить заповнити нішу на ринку аналітичних інструментів, які враховують емоційний фон новин і їхній вплив на динаміку криптовалют.

Як і будь-який інший стартап, цей проект має як переваги, так і ризики. Серед переваг — можливість застосування новітніх методів глибинного навчання, перспектива швидкого масштабування у разі успіху, а також високий попит на інструменти прогнозування на ринку криптовалют. Ризики включають високу конкуренцію, обмеження у фінансуванні на початковому етапі та необхідність забезпечення точності моделей, щоб уникнути значних помилок у прогнозах.

Таким чином, стартап у сфері аналізу ринку криптовалют за допомогою штучного інтелекту має потенціал не лише стати комерційно успішним, а й значно вплинути на розвиток технологій у фінансовій аналітиці, сприяючи вдосконаленню процесів прийняття рішень трейдерами та інвесторами.

4.1 Опис ідеї стартап-проекту

У межах цього підрозділу проаналізовано та подано у вигляді таблиці:

- зміст ідеї;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач продукту;
- відмінності від існуючих аналогів та заміників.

Інформація, подана у табл. 4.1, дає цілісне уявлення про зміст ідеї, потенційні переваги, а також базові ризики, які слід врахувати при визначенні груп потенційних користувачів.

Таблиця 4.1 – Опис ідеї стартап проекту

<i>Зміст ідеї</i>	Розробка платформи для аналізу ринку криптовалют із використанням методів штучного інтелекту. Ця система автоматизує процес обробки новинних потоків та прогнозує динаміку цін криптовалют, враховуючи тональність і вплив новин.
-------------------	---

Закінчення таблиці 4.1

<i>Напрямки застосування</i>	Фінансова аналітика: допомога трейдерам та інвесторам у прийнятті рішень; Освітня сфера: використання в навчанні методів AI для аналізу ринкових даних; Маркетинг та бізнес-аналітика: оцінка ризиків інвестицій.
<i>Вигоди для користувача</i>	Економія часу на аналіз ринкових даних; Підвищення точності прогнозів; Зменшення впливу людського фактора; Інтеграція з популярними трейдинговими платформами для автоматизації рішень.
<i>Відмінності від існуючих аналогів</i>	Застосування сучасних трансформерних моделей (наприклад, BERT, GPT) для аналізу тональності новин; Інтеграція історичних цін та новинних потоків; Інтуїтивно зрозумілий інтерфейс для користувачів без технічної підготовки.

Аналіз потенційних техніко-економічних переваг ідеї дозволив виявити певну кількість аналогічних рішень на ринку, які використовують штучний інтелект для аналізу криптовалютних даних. Однак наш стартап має низку значущих переваг у порівнянні з ними:

- Вища точність прогнозів завдяки інтеграції сучасних трансформерних моделей ШІ, таких як GPT або BERT, для аналізу тональності новин та їхнього впливу на ринок (сильна сторона).

- Комплексний підхід, який об'єднує аналіз новин, ринкових даних і поведінкових факторів, що робить платформу більш універсальною (сильна сторона).

- Зручність використання, забезпечена інтуїтивно зрозумілим інтерфейсом, який орієнтований як на досвідчених аналітиків, так і на звичайних трейдерів (сильна сторона).

- Високі обчислювальні вимоги через необхідність обробки великих обсягів даних у реальному часі (нейтральна сторона).

Таким чином, у порівнянні з існуючими рішеннями, наш стартап вирізняється високим рівнем інноваційності, інтеграцією кількох джерел даних та орієнтацією на широкий спектр користувачів. Це забезпечує конкурентну перевагу на ринку криптовалютних аналітичних платформ.

4.2 Технологічний аудит стартап-проекту

У межах цього підрозділу проведено аудит технологій, за допомогою яких можливо реалізувати ідею проекту. Визначення технологічної здійсненності передбачає аналіз основних компонентів, що подано у табл. 4.2.

Таблиця 4.2 — Визначення технологічної здійсненності ідеї проекту

<i>№</i>	<i>Ідея стартапу</i>	<i>Технології і реалізація</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
1	Аналіз новин та ринкових даних із використанням штучного інтелекту	Python, Hugging Face, Pandas	Наявна	Доступна
2	Розробка нейронних мереж для аналізу текстових даних	TensorFlow, PyTorch	Наявна	Доступна
3	Попередня обробка та аналіз даних	Pandas, NumPy, Scikit-learn	Наявна	Доступна
4	Візуалізація результатів прогнозів	Matplotlib, seaborn, Plotly	Наявна	Доступна

Обраною технологією реалізації є TensorFlow для побудови моделей штучного інтелекту, які працюватимуть із новинними потоками, у поєднанні з Python для організації всього процесу. Для аналізу ринкових даних використовуються Pandas і NumPy, а для візуалізації — Matplotlib і Seaborn. Уся розробка буде інтегрована у середовище Jupyter Notebook, що забезпечує зручність для тестування і прототипування.

За результатами аналізу встановлено, що технологічна реалізація проекту є повністю можливою завдяки доступності всіх необхідних технологій та їх безкоштовності.

Для подальшої реалізації ідеї стартапу також важливо провести аналіз ринкових можливостей, динаміки ринку та потенційних загроз. Попередні результати наведено у Табл. 4.3.

Таблиця 4.3 – Характеристика вхідних даних та особливостей реалізації проекту.

<i>№</i>	<i>Показники</i>	<i>Характеристика</i>
1	Джерела даних	Новинні портали, API криптобірж, соціальні мережі для аналізу новинних потоків і ринкових даних.
2	Обсяг даних	>500 000 новинних повідомлень та історичних ринкових даних за останні 5 років.
3	Складність обробки	Необхідність попередньої обробки даних: очищення тексту, виділення ключових слів, визначення тональності.
4	Вимоги до апаратного забезпечення	Наявність GPU для обробки великих обсягів даних і тренування моделей.
5	Тривалість навчання моделі	Від кількох годин до кількох днів залежно від обсягу даних та обраної моделі (наприклад, BERT, GPT або LSTM).
6	Оновлюваність даних	Необхідність регулярного оновлення бази новин і ринкових даних для забезпечення актуальності прогнозів.

Після цього було здійснено попередній аналіз цільової аудиторії та потенційних користувачів, сформовано їх основні потреби та вимоги до програмного продукту (Табл. 4.4).

Таблиця 4.4 – Характеристика потенційних користувачів.

<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до продукту</i>
Аналіз новин та ринкових даних	Трейдери, інвестори, фінансові аналітики	Різний рівень підготовки: від новачків до професіоналів; необхідність персоналізації даних	Висока точність прогнозів, можливість автоматизації рішень
Візуалізація результатів	Фінансові установи, консалтингові фірми	Потреба у простому та інтуїтивно зрозумілому інтерфейсі для аналізу великих обсягів даних	Інтуїтивно зрозумілий інтерфейс, адаптація до потреб організацій
Інтеграція з інструментами трейдингу	Розробники наукового ПЗ	Вимога до відкритості API для інтеграції з іншими платформами та розширення функціоналу	Легкість інтеграції, масштабованість, стабільність роботи

У межах аналізу ринкового середовища було визначено фактори потенційних загроз та можливостей, а також розроблено можливі реакції команди стартап-проекту. Результати наведені у Табл. 4.5 та Табл. 4.6.

Таблиця 4.5 – Аналіз факторів потенційних загроз

<i>№</i>	<i>Фактор</i>	<i>Загроза</i>	<i>Можлива реакція</i>
1	Конкуренція на ринку	Велика кількість проектів у сфері фінансових технологій, що використовують штучний інтелект для аналізу криптовалют	Створення унікального аналітичного функціоналу, наприклад, прогнозів з урахуванням глобальних новинних подій.
2	Технологічні обмеження	Збільшення обсягів даних може перевищити доступні обчислювальні ресурси	Використання хмарних сервісів для масштабування потужностей, таких як AWS, Azure або Google Cloud.
3	Залежність від якості даних	Новинні потоки можуть містити багато неточних, фейкових або неструктурованих даних	Впровадження механізмів фільтрації та верифікації даних на етапі попередньої обробки.
4	Регуляторні зміни	Введення обмежень на доступ до криптобіржових API або новинних платформ	Забезпечення підтримки альтернативних джерел даних і регулярний моніторинг законодавчих змін у цій сфері.

Кінець таблиці 4.5

5	Відсутність користувачької довіри	Користувачі можуть сумніватися в точності прогнозів, зроблених системою	Публікація порівнянь результатів прогнозів із реальними ринковими змінами для підвищення довіри.
6	Швидкість змін на ринку криптовалют	Різка зміна ринкових умов може зменшити точність прогнозів системи	Використання методів реального часу для аналізу новин та ринкових даних, що оновлюються щохвилини.

Таблиця 4.6 – Аналіз факторів потенційних можливостей

<i>№</i>	<i>Фактор</i>	<i>Зміст</i>	<i>Реакція</i>
1	Розвиток методів штучного інтелекту	Нові трансформерні моделі та алгоритми ШІ покращують точність аналізу та прогнозів	Впровадження інноваційних алгоритмів та регулярне оновлення використовуваних моделей
2	Зростання популярності криптовалют	Підвищення інтересу до криптовалют серед широкої аудиторії та інвесторів	Розширення функціоналу для різних рівнів користувачів: від новачків до професійних аналітиків

Кінець таблиці 4.6

3	Доступність даних із криптобірж	Розширення API криптобірж для отримання точних і актуальних ринкових даних	Інтеграція з більшою кількістю джерел для підвищення точності аналізу
4	Поширення автоматизації у трейдингу	Автоматизовані системи допомагають трейдерам приймати рішення швидше та ефективніше	Розробка модулів для автоматичного виконання трейдинг-стратегій на основі прогнозів
5	Популяризація фінансової грамотності	Більше людей цікавляться фінансовими технологіями, зокрема аналітикою ринку криптовалют	Створення навчальних матеріалів і модулів для ознайомлення з основами роботи платформи

На основі аналізу було визначено фактори, які впливають на конкурентоспроможність проекту, і результати представлені в Табл. 4.7.

Таблиця 4.7 — Аналіз факторів конкурентоспроможності

<i>№</i>	<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування</i>
1	Унікальність методів аналізу	Використання сучасних трансформерних моделей (BERT, GPT), які забезпечують високу точність аналізу новин і прогнозів.
2	Простота використання	Інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко освоїти платформу без спеціальної технічної підготовки.
3	Висока продуктивність	Здатність швидко обробляти великі обсяги даних завдяки оптимізованим алгоритмам і використанню хмарних обчислень.

На завершення було виконано SWOT-аналіз проекту, який включає сильні та слабкі сторони, а також можливості і загрози, наведені у табл. 4.8.

Таблиця 4.8 — SWOT-аналіз проекту

<i>Сильні сторони</i>	<i>Слабкі сторони</i>
Висока точність прогнозів завдяки сучасним моделям ШІ	Велика кількість неструктурованих даних, які потребують обробки
Доступність технологій для реалізації	Потреба у спеціалізованому обладнанні для тренування моделей
Простота інтеграції з іншими платформами	Обмежений функціонал на ранніх етапах розробки
<i>Можливості</i>	<i>Загрози</i>
Впровадження новітніх методів штучного інтелекту	Зростання конкуренції на ринку платформ для аналізу криптовалют
Розширення аудиторії користувачів за рахунок навчальних модулів	Зміна стандартів або форматів обробки даних
Інтеграція додаткових джерел даних та API криптобірж	Поява нових технологій або сервісів, які можуть витіснити продукт

4.3 Аналіз ринкової стратегії стартап-проекту

Для успішної реалізації стартап-проекту, пов'язаного з аналізом ринку криптовалют за допомогою методів штучного інтелекту, необхідно розробити ефективну ринкову стратегію. Вона включає визначення стратегії охоплення ринку, зокрема аналіз цільових груп потенційних споживачів. Результати такого аналізу наведено в табл. 4.9.

Таблиця 4.9 – Аналіз цільових груп потенційних споживачів

<i>№</i>	<i>Опис профілю цільової групи</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит у межах ЦГ</i>	<i>Інтенсивність конкуренції у сегменті</i>	<i>Простота входу</i>
1	Трейдери та приватні інвестори	Висока	Високий	Сильна	Складно
2	Фінансові установи та аналітичні компанії	Висока	Середній	Середня	Середня
3	Стартапи у галузі фінансових технологій	Середня	Низький	Висока	Середня

На основі проведеного аналізу визначено, що основними цільовими групами споживачів є трейдери та приватні інвестори, фінансові установи та аналітичні компанії, а також стартапи у сфері фінансових технологій. Обрана стратегія охоплення ринку — стратегія диференційованого маркетингу, яка дозволяє орієнтуватися на декілька сегментів ринку.

Для забезпечення ефективності роботи у визначених сегментах було сформовано базову стратегію розвитку проекту (Табл. 4.10).

Таблиця 4.10 — Визначення базової стратегії розвитку

<i>№</i>	<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку</i>
1	Фокус на трейдерах та інвесторах	Диференційований маркетинг	Висока точність прогнозів; Інтуїтивно зрозумілий інтерфейс; Можливість інтеграції з трейдинговими платформами	Розробка функцій автоматизації рішень; маркетингові кампанії для залучення трейдерів
2	Співпраця з фінансовими установами	Корпоративний підхід	Гнучкість налаштувань для потреб компаній; Можливість інтеграції в існуючі системи	Розробка корпоративних рішень; встановлення довгострокових партнерств
3	Розширення функціоналу для стартапів	Стратегія підтримки B2B клієнтів	Інтеграція API; підтримка масштабованості; стабільність системи	Створення стандартизованих API; розробка документації для розробників

Кінець таблиці 4.10

4	Підвищення фінансової грамотності	Освітній маркетинг	Інтерактивні навчальні модулі; інформативна підтримка для новачків	Створення освітніх матеріалів та курсів; організація вебінарів
---	-----------------------------------	--------------------	--	--

Крім того, було проаналізовано стратегію конкурентної поведінки стартапу, результати якої наведені в табл. 4.11.

Таблиця 4.11 — Визначення стратегії конкурентної поведінки

<i>Чи є стартап «першопрохідцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента?</i>	<i>Стратегія конкурентної поведінки</i>
Ні	Компанія буде шукати нових споживачів і частково залучати аудиторію конкурентів	Ні, компанія фокусується на створенні унікального функціоналу та інноваційних рішень	Стратегія диференціації, орієнтована на унікальність продукту

На основі аналізу вимог споживачів, обраної стратегії розвитку та конкурентної поведінки було сформовано стратегію позиціонування продукту стартап-проекту. Ця стратегія передбачає створення позиції, яка вирізняє продукт від інших на ринку. Результати наведено у табл. 4.12.

Таблиця 4.12 — Вимоги до продукту з боку цільової аудиторії

<i>Вимоги до продукту з боку ЦА</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції старту</i>	<i>Вибір асоціацій, які мають сформувати позицію</i>
Висока точність прогнозів; зручний інтерфейс для трейдерів та аналітиків; інтеграція з API криптобірж	Стратегія інновацій	Висока точність і швидкість прогнозів; зручність інтеграції	Інноваційність, надійність, доступність для різних груп користувачів
Актуальність даних; підтримка великих обсягів новин і ринкових показників	Стратегія технологічного лідерства	Можливість обробки великих обсягів даних у реальному часі	Технологічна досконалість, стабільність, ефективність

4.4 Розробка маркетингової стратегії стартап-проекту

У цьому підрозділі розглядається процес формування маркетингової стратегії просування продукту стартап-проекту, пов'язаного з автоматичною класифікацією морфології галактик. Під концепцією продукту мається на увазі опис його функціональних і фізичних характеристик, які задовольняють потреби споживачів, а також переваг, які забезпечують конкурентоспроможність на ринку.

На першому етапі визначено ключові переваги концепції продукту стартапу. Результати наведено в Табл. 4.13.

Таблиця 4.13 — Визначення ключових переваг концепції потенційного продукту

<i>№</i>	<i>Потреба</i>	<i>Вигода, яку пропонує продукт</i>	<i>Ключові переваги перед конкурентами</i>
1	Висока точність аналізу та прогнозів	Використання трансформерних моделей ШІ (BERT, GPT), що забезпечують детальний аналіз новин і ринкових даних	Найкраща точність прогнозів із врахуванням багатofакторних залежностей
2	Зручність інтеграції з іншими системами	Гнучкий API для інтеграції з криптобіржами, трейдинговими платформами та іншими фінансовими інструментами	Інтеграція з провідними платформами без необхідності складного налаштування

Кінець таблиці 4.13

3	Швидкість обробки великих обсягів даних у реальному часі	Автоматична обробка потоків даних у режимі реального часу для миттєвих рішень	Забезпечення реальної продуктивності навіть за високого навантаження
4	Актуальність даних	Регулярне оновлення інформації з кількох джерел, включаючи соціальні мережі, новинні портали та API криптобірж	Постійна актуальність даних для збереження точності прогнозів
5	Зрозумілість і адаптивність інтерфейсу	Інтерфейс, що налаштовується під рівень користувача: від початківця до професіонала	Простота використання для новачків і розширені функції для досвідчених користувачів

Наступним кроком було формування трирівневої маркетингової моделі продукту, яка деталізує його функціональність, властивості та вигоди. Ця модель наведена в Табл. 4.14.

Таблиця 4.14 — Опис трьох рівнів товару

<i>Рівні товару</i>	<i>Сутність та складові</i>
I. Товар за задумом	Програмний продукт для аналізу ринку криптовалют із використанням методів штучного інтелекту.
	Основна функція: автоматизація аналізу новинних потоків, визначення тональності та впливу новин на ринок.
II. Товар у реальному виконанні	Характеристики: точність прогнозів, швидкість обробки даних, зручний інтерфейс, інтеграція з API криптобірж.
	Якість: регулярне оновлення алгоритмів, перевірка результатів на реальних даних, адаптація до змін ринку.
	Марка: назва продукту, яка асоціюється з надійністю, інноваціями та точністю прогнозів.
III. Товар із підкріпленням	До продажу: забезпечення документації, тестових наборів даних, навчальних матеріалів для користувачів.
	Після продажу: технічна підтримка, консультації, регулярне оновлення продукту.
	Захист: ліцензія, патентування алгоритмів аналізу новин.

Останнім кроком у розробці маркетингової стратегії стало визначення концепції маркетингових комунікацій, які базуються на специфіці поведінки

потенційних споживачів та ключових перевагах продукту. Результати наведені в табл. 4.15.

Таблиця 4.15 — Концепції маркетингових комунікацій

<i>Специфіка поведінки ЦА</i>	<i>Канали комунікації</i>	<i>Ключові позиції для позиціювання</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
Використання продукту трейдерами та інвесторами	Соціальні мережі, фінансові форуми, email-розсилки, вебінари	Висока точність прогнозів, швидкість аналізу даних	Підвищення довіри до продукту через демонстрацію результатів	Простота та надійність у прийнятті торгових рішень
Використання продукту фінансовими і установами	Прямі переговори, конференції, професійні виставки	Гнучкість інтеграції, відповідність корпоративним стандартам	Формування інтересу до автоматизації рішень і підвищення ефективності	Технологічна і інновація для оптимізації фінансових процесів
Інтеграція продукту стартапами у фінансових технологіях	Платформи для розробників (GitHub, API форуми), професійні блоги	Легкість інтеграції API, масштабованість рішення	Підвищення обізнаності про можливості інтеграції з іншими платформами	Відкритість і підтримка інноваційних рішень

Кінець таблиці 4.15

Навчання новачків у сфері криптовалют	Освітні платформи, YouTube, інтерактивні курси	Простота інтерфейсу, легкість освоєння	Популяризація знань про криптовалюту та використання аналітичних інструментів	Допомога у швидкому освоєнні ринку криптовалют
---------------------------------------	--	--	---	--

Висновки до розділу 4

У цьому розділі було розроблено стартап-проект, спрямований на створення програмного продукту для аналізу ринку криптовалют за допомогою методів штучного інтелекту. Було визначено основні цільові групи споживачів, такі як трейдери, фінансові установи, стартапи у сфері фінансових технологій та новачки, які прагнуть освоїти ринок криптовалют. Розроблено маркетингову стратегію, яка включає просування продукту через соціальні мережі, конференції, фінансові форуми та освітні платформи. Крім того, було визначено ключові конкурентні переваги продукту: висока точність прогнозів, швидкість обробки великих обсягів даних та інтуїтивно зрозумілий інтерфейс.

Розроблено трирівневу модель продукту, що включає функціональні можливості, реальне виконання з адаптацією до потреб користувачів та постпродажну підтримку. Було сформовано стратегію позиціонування, яка підкреслює інноваційність, надійність та технологічну досконалість продукту. Також проаналізовано ризики, пов'язані з конкуренцією та зміною стандартів, і запропоновано шляхи їх мінімізації.

Проект має значний потенціал у вирішенні актуальних проблем трейдингу та фінансової аналітики. Інноваційність продукту, здатність

інтегруватися з API криптобірж та високий попит на інструменти прогнозування забезпечують його конкурентоспроможність на ринку. Фінансування проекту є обґрунтованим, оскільки очікувана економічна вигода для трейдерів, фінансових установ та інших кінцевих користувачів значно перевищує витрати на його розробку.

ВИСНОВКИ

У процесі дослідження були розглянуті підходи до прогнозування динаміки цін криптовалют з урахуванням не лише їхніх історичних часових рядів, але й зовнішніх інформаційних чинників. Зокрема, було приділено увагу інтеграції екзогенних фіч, отриманих із новинних потоків, завдяки чому з'явилась можливість більш комплексно підходити до оцінки ринкових тенденцій. Традиційні статистичні моделі, такі як ARIMA або SARIMA, продемонстрували свою спроможність будувати прогнози за умови стаціонарності та відносної передбачуваності даних. Втім, із огляду на високоволатильну та динамічну природу ринку криптовалют, одних класичних методів недостатньо для адекватного врахування нелінійних та швидкозмінних залежностей. Введення у модель зовнішніх індикаторів, наприклад, ознак, які відображають тональність, важливість або популярність новин, дозволило суттєво розширити інформаційну базу для ухвалення прогнозних рішень.

Використання сучасних нейронних архітектур, зокрема рекурентних мереж типу LSTM та GRU, а також трансформерних моделей для векторизації тексту, підтвердило їх потенціал у контексті складних фінансових задач. Ці підходи виявилися особливо дієвими для завдань, де необхідно враховувати довгострокові залежності й поєднувати декілька типів даних — часові ряди, новини, індикатори ринку та поведінкові сигнали. Результати експериментів засвідчили, що включення до моделей текстових фіч із криптоновин мало позитивний вплив на точність прогнозів у випадку деяких моделей, наприклад, SARIMAX та LSTM. Водночас інші архітектури, зокрема GRU, не продемонстрували суттєвого поліпшення, що сигналізує про потребу в ретельному налаштуванні параметрів, відборі ознак та виборі відповідних гіперпараметрів, а також у подальшому вдосконаленні процесу попередньої обробки даних. Важливо також зазначити, що глибина та складність моделей

не завжди гарантує кращий результат: нерідко простіший підхід із ретельним відбором даних, налаштуванням вікна прогнозування або правильним масштабуванням сигналів може продемонструвати кращу узгодженість із реальними ринковими рухами.

Загалом результати роботи свідчать, що поєднання традиційного аналізу часових рядів із методами машинного навчання та обробки природної мови створює перспективний шлях для побудови більш точних, гнучких та інформативних моделей прогнозування динаміки криптовалют. Такий підхід дозволяє краще врахувати складні взаємозв'язки між ринковими індикаторами, новинним фоном та історичними даними. На практиці це може мати значну цінність для трейдерів, інвесторів, фінансових аналітиків та інших зацікавлених сторін, адже покращена точність оцінки майбутньої вартості активів сприятиме прийняттю більш виважених та аргументованих рішень. Водночас подальше дослідження залишається актуальним, оскільки постійний розвиток як ринку криптовалют, так і технологій обробки інформації, відкриває нові можливості для вдосконалення підходів та підвищення якості прогнозів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3), 1139-1168.
2. Gross-Klussmann, A., & Hautsch, N. (2011). News sentiment in financial markets. *Journal of Economic Behavior & Organization*, 85(2), 144-158.
3. Baker, M., & Wurgler, J. (2006). Investor sentiment and the cross-section of stock returns. *The Journal of Finance*, 61(4), 1645-1680.
4. Heston, S. L., & Sinha, N. R. (2017). News vs. sentiment: Predicting stock returns from news stories. *Financial Analysts Journal*, 73(3), 18-25.
5. Lazer, D. M., Baum, M. A., Grinberg, N., et al. (2018). The science of fake news. *Science*, 359(6380), 1094-1096.
6. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*.
7. Gujarati, D. N., & Porter, D. C. (2009). *Basic Econometrics*.
8. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*.
9. Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). *The Econometrics of Financial Markets*.
10. Wooldridge, J. M. (2010). *Econometric Analysis of Cross Section and Panel Data*.
11. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
12. Vaswani, A., et al. (2017). Attention Is All You Need.
13. Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space.
14. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.
15. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*.
16. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory.

17. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*.
18. Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1-8.
19. Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), 35-65.
20. Madan, I., Saluja, S., & Vankataraman, A. (2015). Automated bitcoin trading via machine learning algorithms. *IEEE Conference on SmartCity/SocialCom/SustainCom*, 2015, 494-501.
21. Řehůřek, R., & Sojka, P. (2010). Gensim—Statistical Semantics in Python. *Statistical Software*, 70(1), 1-9.
22. Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*, 4171–4186.
23. Chen, S., Chen, C., & Li, T. (2020). Empirical Study of Bitcoin Price Prediction Using LSTM Neural Networks. *IEEE International Conference on Big Data*, 2020, 2805–2814.
24. McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of Bitcoin using Machine Learning. *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 339–343.
25. Jiang, Z., Liang, J., & Liu, Y. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.
26. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. *NeurIPS*, 2017, 5998–6008.
27. Box, G.E.P., Jenkins, G.M., & Reinsel, G.C. *Time Series Analysis: Forecasting and Control*. Wiley, 2008.
28. Hyndman, R.J., & Athanasopoulos, G. *Forecasting: Principles and Practice*. OTexts, 2018.

29. Pichl, L., & Kaizoji, T. Volatility analysis of Bitcoin price time series. *Quantitative Finance and Economics*, 2017.
30. Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press, 2016.
31. Hochreiter, S., & Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 1997.
31. Greff, K. et al. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.
32. Cho, K. et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *EMNLP*, 2014.
33. Devlin, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*, 2019.
34. Ding, Z. et al. Forecasting Bitcoin price with a hybrid ARIMA and artificial neural network model. *Journal of International Technology and Information Management*, 2018.
35. Hyndman, R.J., & Koehler, A.B. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), pp. 679-688, 2006.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import pandas as pd

cryptopanic_news =
pd.read_csv(r'C:\Users\serhii\Documents\diploma_crypto_currency\CryptoNewsDat
aset_csvOutput\CryptoNewsDataset_csvOutput\cryptopanic_news.csv')
source =
pd.read_csv(r'C:\Users\serhii\Documents\diploma_crypto_currency\CryptoNewsDat
aset_csvOutput\CryptoNewsDataset_csvOutput\source.csv')
currency =
pd.read_csv(r'C:\Users\serhii\Documents\diploma_crypto_currency\CryptoNewsDat
aset_csvOutput\CryptoNewsDataset_csvOutput\currency.csv')
news__currency =
pd.read_csv(r'C:\Users\serhii\Documents\diploma_crypto_currency\CryptoNewsDat
aset_csvOutput\CryptoNewsDataset_csvOutput\news__currency.csv')
news_currencies_source_joinedResult =
pd.read_csv(r'C:\Users\serhii\Documents\diploma_crypto_currency\CryptoNewsDat
aset_csvOutput\CryptoNewsDataset_csvOutput\news_currencies_source_joinedResul
t.csv')

print('cryptopanic_news')
print(cryptopanic_news.columns)
print('source')
print(source.columns)
print('currency')
print(currency.columns)
print('news__currency')
print(news__currency.columns)
print('news_currencies_source_joinedResult')
print(news_currencies_source_joinedResult.columns)

news_with_source = pd.merge(
    cryptopanic_news, source, left_on='sourceId', right_on='id',
    suffixes=('', '_source')
)

news_with_currency_ids = pd.merge(
    news_with_source, news__currency, left_on='id', right_on='newsId'
)

news_with_currencies = pd.merge(
    news_with_currency_ids, currency, left_on='currencyId', right_on='id',
    suffixes=('', '_currency')
)

news_with_currencies_grouped = news_with_currencies.groupby(
    ['id', 'title', 'sourceId', 'domain', 'newsDatetime', 'url', 'negative',
    'positive',
    'important', 'liked', 'disliked', 'lol', 'toxic', 'saved', 'comments']
).agg({'code': lambda x: list(x)}).reset_index()

```

```

news_with_currencies_grouped.rename(columns={'code': 'currencies'},
inplace=True)

news_with_currencies_grouped

news_with_currencies_grouped['newsDatetime'] =
pd.to_datetime(news_with_currencies_grouped['newsDatetime'], format='%m/%d/%Y
%H:%M')

news_with_currencies_grouped['newsDatetime'].unique()

import pandas as pd
import nltk
from nltk.corpus import stopwords
import re

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

def extract_keywords(title):
    words = re.findall(r'\b\w+\b', title.lower())
    keywords = [word for word in words if word not in stop_words]
    return keywords

from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
label_encoder.fit_transform(news_with_currencies_grouped['title'])

sentence_vectors_df = pd.DataFrame(sentence_vectors,
columns=[f"sentence_vector_{i}" for i in range(384)])

news_with_currencies_grouped = pd.concat([news_with_currencies_grouped,
sentence_vectors_df], axis=1)

news_with_currencies_grouped.to_csv(r'news_final_df.csv')

df =
pd.read_csv(r'C:\Users\andrew\Documents\diploma_crypto_currency\btcusd_1-
min_data.csv')

df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')

print(news_with_currencies_grouped['newsDatetime'].min())
print(news_with_currencies_grouped['newsDatetime'].max())
print(len(news_with_currencies_grouped))
print(df['Timestamp'].min())
print(df['Timestamp'].max())
print(len(df))

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

```

```

plt.plot(df['Timestamp'], df['Close'], label='Close Price',
color='lightblue', linewidth=2)

plt.title('Close Prices', fontsize=14)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Close Price', fontsize=12)
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()

df = df[(df['Timestamp'] >= '2017-09-23') & (df['Timestamp'] <= '2024-11-
30')]

df = df.resample('D', on='Timestamp').mean().reset_index()

df.index = df.Timestamp

df = df.dropna()

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.plot(df['Timestamp'], df['Close'], label='Close Price',
color='lightblue', linewidth=2)

plt.title('Close Prices', fontsize=14)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Close Price', fontsize=12)
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

plt.tight_layout()
plt.savefig(r'images\BTC_close_price')
plt.show()

# Model learning without cryptonews data

metrics = {}

train_size = int(len(df) * 0.8)
train, test = df.iloc[:train_size], df.iloc[train_size:]

from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

target = df['Close']

exog = df[['Open', 'High', 'Low', 'Volume']]

```

```

train_size = int(len(df) * 0.8)
train_target, test_target = target.iloc[:train_size],
target.iloc[train_size:]
train_exog, test_exog = exog.iloc[:train_size], exog.iloc[train_size:]

model = SARIMAX(train_target, exog=train_exog, order=(2, 1, 2))
fitted_model = model.fit()

forecast = fitted_model.forecast(steps=len(test_target), exog=test_exog)

mae = mean_absolute_error(test_target, forecast)
mse = mean_squared_error(test_target, forecast)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((test_target - forecast) / test_target)) * 100
r2 = r2_score(test_target, forecast)

metrics['SARIMAX_trade_data_only'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")
print(f"R-squared (R²): {r2:.2f}")

import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.plot(test_target.index, test_target, label='Actual', color='blue',
linestyle='-', linewidth=2)
plt.plot(test_target.index, forecast, label='Forecast', color='red',
linestyle='--', linewidth=2)
plt.legend()
plt.title('SARIMAX Forecast vs Actual(екзогенні фічі, без новин)')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.savefig(r'images\SARIMAX Forecast vs Actual(екзогенні фічі, без
новин).jpeg')
plt.show()

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

scaler_target = MinMaxScaler(feature_range=(0, 1))
train_target_scaled =
scaler_target.fit_transform(train_target.values.reshape(-1, 1))
test_target_scaled = scaler_target.transform(test_target.values.reshape(-1,
1))

```

```

scaler_exog = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = scaler_exog.fit_transform(train_exog)
test_exog_scaled = scaler_exog.transform(test_exog)

def create_lstm_data(target, exog, timesteps):
    X, y = [], []
    for i in range(timesteps, len(target)):
        X.append(np.hstack([exog[i-timesteps:i], target[i-timesteps:i]]))
        y.append(target[i])
    return np.array(X), np.array(y)

timesteps = 10

X_train, y_train = create_lstm_data(train_target_scaled, train_exog_scaled,
timesteps)
X_test, y_test = create_lstm_data(test_target_scaled, test_exog_scaled,
timesteps)

model = Sequential([
    LSTM(50, activation='relu', input_shape=(X_train.shape[1],
X_train.shape[2])),
    Dense(1)
])

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

history = model.fit(X_train, y_train, epochs=20, batch_size=32,
validation_data=(X_test, y_test), verbose=1)

predictions_scaled = model.predict(X_test)
predictions = scaler_target.inverse_transform(predictions_scaled)

y_test_actual = scaler_target.inverse_transform(y_test)

mae = mean_absolute_error(y_test_actual, predictions)
mse = mean_squared_error(y_test_actual, predictions)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_test_actual - predictions) / y_test_actual)) * 100
r2 = r2_score(y_test_actual, predictions)

metrics['LSTM_trade_data_only'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")
print(f"R-squared (R2): {r2}")

import matplotlib.pyplot as plt

```

```

plt.figure(figsize=(12, 6))
plt.plot(y_test_actual, label='Actual', color='blue', linestyle='-',
linewidth=2)
plt.plot(predictions, label='Predicted', color='red', linestyle='--',
linewidth=2)
plt.legend()
plt.title('LSTM Prediction vs Actual(екзогенні фічі, без новин)')
plt.xlabel('Time Steps')
plt.ylabel('Target Variable')
plt.savefig(r'images\LSTM Forecast vs Actual(екзогенні фічі, без
новин).jpeg')
plt.show()

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt

scaler_target = MinMaxScaler(feature_range=(0, 1))
train_target_scaled =
scaler_target.fit_transform(train_target.values.reshape(-1, 1))
test_target_scaled = scaler_target.transform(test_target.values.reshape(-1,
1))

scaler_exog = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = scaler_exog.fit_transform(train_exog)
test_exog_scaled = scaler_exog.transform(test_exog)

def create_gru_data(target, exog, timesteps):
    X, y = [], []
    for i in range(timesteps, len(target)):
        X.append(np.hstack([exog[i-timesteps:i], target[i-timesteps:i]]))
        y.append(target[i])
    return np.array(X), np.array(y)

timesteps = 10

X_train, y_train = create_gru_data(train_target_scaled, train_exog_scaled,
timesteps)
X_test, y_test = create_gru_data(test_target_scaled, test_exog_scaled,
timesteps)

model_gru = Sequential([
    GRU(50, activation='relu', input_shape=(X_train.shape[1],
X_train.shape[2])),
    Dense(1)
])

model_gru.compile(optimizer='adam', loss='mean_squared_error')

```

```

history = model_gru.fit(X_train, y_train, epochs=20, batch_size=32,
validation_data=(X_test, y_test), verbose=1)

predictions_scaled = model_gru.predict(X_test)
predictions = scaler_target.inverse_transform(predictions_scaled)

y_test_actual = scaler_target.inverse_transform(y_test)

mae = mean_absolute_error(y_test_actual, predictions)
mse = mean_squared_error(y_test_actual, predictions)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_test_actual - predictions) / y_test_actual)) * 100
r2 = r2_score(y_test_actual, predictions)

metrics['GRU_trade_data_only'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape}%")
print(f"R-squared (R2): {r2}")

plt.figure(figsize=(12, 6))
plt.plot(y_test_actual, label='Actual', color='blue', linestyle='-',
linewidth=2)
plt.plot(predictions, label='Predicted', color='red', linestyle='--',
linewidth=2)
plt.legend()
plt.title('GRU Prediction vs Actual(екзогенні фічі, без новин)')
plt.xlabel('Time Steps')
plt.ylabel('Target Variable')
plt.savefig(r'images\GRU Forecast vs Actual(екзогенні фічі, без новин).jpeg')
plt.show()

# Models with cryptonews features without label encoding

news_with_currencies_grouped[['negative', 'positive', 'important', 'liked', 'disl
iked', 'lol', 'toxic', 'saved', 'comments']] =
news_with_currencies_grouped[['negative', 'positive', 'important', 'liked', 'disl
iked', 'lol', 'toxic', 'saved', 'comments']].astype('float32')

news_with_currencies_grouped = news_with_currencies_grouped.dropna(how='all')
news_resample = news_with_currencies_grouped[['newsDatetime',
'negative', 'positive', 'important', 'liked', 'disliked', 'lol', 'toxic', 'saved', 'c
omments']].resample('D', on='newsDatetime').mean().reset_index()

news_resample = news_resample.dropna()

```

```

df.drop(columns=['Timestamp'], inplace=True)

df2 = pd.merge(news_resample, df, left_on='newsDatetime',
               right_on='Timestamp', how='inner')

train_size = int(len(df2) * 0.8)
train, test = df2.iloc[:train_size], df2.iloc[train_size:]

from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

target = df2['Close']

exog = df2[['Open', 'High', 'Low', 'Volume',
            'negative', 'positive', 'important', 'liked', 'disliked', 'lol', 'toxic', 'saved', 'c
omments']]

train_size = int(len(df2) * 0.8)
train_target, test_target = target.iloc[:train_size],
target.iloc[train_size:]
train_exog, test_exog = exog.iloc[:train_size], exog.iloc[train_size:]

model = SARIMAX(train_target, exog=train_exog, order=(2, 1, 2))
fitted_model = model.fit()

forecast = fitted_model.forecast(steps=len(test_target), exog=test_exog)

mae = mean_absolute_error(test_target, forecast)
mse = mean_squared_error(test_target, forecast)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((test_target - forecast) / test_target)) * 100
r2 = r2_score(test_target, forecast)

metrics['SARIMAX_trade_data_and_cryptonews'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")
print(f"R-squared (R2): {r2:.2f}")

import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.plot(test_target.index, test_target, label='Actual', color='blue',
         linestyle='-', linewidth=2)
plt.plot(test_target.index, forecast, label='Forecast', color='red',
         linestyle='--', linewidth=2)
plt.legend()

```

```
plt.title('SARIMAX Forecast vs Actual(екзогенні фічі, з новинами)')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.savefig(r'images\SARIMAX Forecast vs Actual(екзогенні фічі, з новинами).jpeg')
plt.show()
```

```
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
```

```
scaler_target = MinMaxScaler(feature_range=(0, 1))
train_target_scaled =
scaler_target.fit_transform(train_target.values.reshape(-1, 1))
test_target_scaled = scaler_target.transform(test_target.values.reshape(-1, 1))
```

```
scaler_exog = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = scaler_exog.fit_transform(train_exog)
test_exog_scaled = scaler_exog.transform(test_exog)
```

```
def create_lstm_data(target, exog, timesteps):
    X, y = [], []
    for i in range(timesteps, len(target)):
        X.append(np.hstack([exog[i-timesteps:i], target[i-timesteps:i]]))
        y.append(target[i])
    return np.array(X), np.array(y)
```

```
timesteps = 10
```

```
X_train, y_train = create_lstm_data(train_target_scaled, train_exog_scaled,
timesteps)
X_test, y_test = create_lstm_data(test_target_scaled, test_exog_scaled,
timesteps)
```

```
model = Sequential([
    LSTM(50, activation='relu', input_shape=(X_train.shape[1],
X_train.shape[2])),
    Dense(1)
])
```

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

```
history = model.fit(X_train, y_train, epochs=20, batch_size=32,
validation_data=(X_test, y_test), verbose=1)
```

```
predictions_scaled = model.predict(X_test)
predictions = scaler_target.inverse_transform(predictions_scaled)
```

```
y_test_actual = scaler_target.inverse_transform(y_test)
```

```
mae = mean_absolute_error(y_test_actual, predictions)
mse = mean_squared_error(y_test_actual, predictions)
```

```

rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_test_actual - predictions) / y_test_actual)) * 100
r2 = r2_score(y_test_actual, predictions)

metrics['LSTM_trade_data_and_cryptonews'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")
print(f"R-squared (R2): {r2}")

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
plt.plot(y_test_actual, label='Actual', color='blue', linestyle='-',
         linewidth=2)
plt.plot(predictions, label='Predicted', color='red', linestyle='--',
         linewidth=2)
plt.legend()
plt.title('LSTM Prediction vs Actual(екзогенні фічі, з новинами)')
plt.xlabel('Time Steps')
plt.ylabel('Target Variable')
plt.savefig(r'images\LSTM Forecast vs Actual(екзогенні фічі, з новинами).jpeg')
plt.show()

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt

scaler_target = MinMaxScaler(feature_range=(0, 1))
train_target_scaled = scaler_target.fit_transform(train_target.values.reshape(-1, 1))
test_target_scaled = scaler_target.transform(test_target.values.reshape(-1, 1))

scaler_exog = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = scaler_exog.fit_transform(train_exog)
test_exog_scaled = scaler_exog.transform(test_exog)

def create_gru_data(target, exog, timesteps):
    X, y = [], []
    for i in range(timesteps, len(target)):

```

```

        X.append(np.hstack([exog[i-timesteps:i], target[i-timesteps:i]]))
        y.append(target[i])
    return np.array(X), np.array(y)

timesteps = 10

X_train, y_train = create_gru_data(train_target_scaled, train_exog_scaled,
timesteps)
X_test, y_test = create_gru_data(test_target_scaled, test_exog_scaled,
timesteps)

model_gru = Sequential([
    GRU(50, activation='relu', input_shape=(X_train.shape[1],
X_train.shape[2])),
    Dense(1)
])

model_gru.compile(optimizer='adam', loss='mean_squared_error')

history = model_gru.fit(X_train, y_train, epochs=20, batch_size=32,
validation_data=(X_test, y_test), verbose=1)

predictions_scaled = model_gru.predict(X_test)
predictions = scaler_target.inverse_transform(predictions_scaled)

y_test_actual = scaler_target.inverse_transform(y_test)

mae = mean_absolute_error(y_test_actual, predictions)
mse = mean_squared_error(y_test_actual, predictions)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_test_actual - predictions) / y_test_actual)) * 100
r2 = r2_score(y_test_actual, predictions)

metrics['GRU_trade_data_and_cryptonews'] = {
    'mae': mae,
    'mse': mse,
    'rmse': rmse,
    'mape': mape,
    'r2': r2
}

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape}%")
print(f"R-squared (R2): {r2}")

plt.figure(figsize=(12, 6))
plt.plot(y_test_actual, label='Actual', color='blue', linestyle='-',
linewidth=2)
plt.plot(predictions, label='Predicted', color='red', linestyle='--',
linewidth=2)
plt.legend()
plt.title('GRU Prediction vs Actual(екзогенні фічі, з новинами)')

```

```

plt.xlabel('Time Steps')
plt.ylabel('Target Variable')
plt.savefig(r'images\GRU Forecast vs Actual(екзогенні фічі, з
новинами).jpeg')
plt.show()

# Models using only Close

from statsmodels.tsa.statespace.sarimax import SARIMAX

target = df['Close']

train_size = int(len(df) * 0.8)
train_target, test_target = target[:train_size], target[train_size:]

history = [x for x in train_target]

arima_forecast = []
for t in range(len(test_target)):
    model = SARIMAX(history, order=(2,1,2))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    arima_forecast.append(yhat)
    obs = test_target[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))

mae_arima = mean_absolute_error(test_target, arima_forecast)
mse_arima = mean_squared_error(test_target, arima_forecast)
rmse_arima = np.sqrt(mse_arima)
mape_arima = np.mean(np.abs((test_target - arima_forecast) / test_target)) *
100
r2_arima = r2_score(test_target, arima_forecast)

plt.figure(figsize=(10, 6))
plt.plot(test_target.index, test_target, color='blue', label='True')
plt.plot(test_target.index, arima_forecast, color='red', label='SARIMAX
Forecast')
plt.title('SARIMAX Forecast vs True(лише Close, без новин)')
plt.legend()
plt.savefig(r'images\SARIMAX Forecast vs Actual(лише Close, без новин).jpeg')
plt.show()

metrics['SARIMAX_only_Close'] = {
    'mae': mae_arima,
    'mse': mse_arima,
    'rmse': rmse_arima,
    'mape': mape_arima,
    'r2': r2_arima
}

test_target.shape

```

```
X_test.shape
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score
```

```
scaler = MinMaxScaler(feature_range=(0, 1))
train_scaled = scaler.fit_transform(train_target.values.reshape(-1, 1))
test_scaled = scaler.transform(test_target.values.reshape(-1, 1))
```

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

```
time_step = 60
X_train, y_train = create_dataset(train_scaled, time_step)
X_test, y_test = create_dataset(test_scaled, time_step)
```

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

```
lstm_model = Sequential()
lstm_model.add(LSTM(units=50, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
lstm_model.add(LSTM(units=50, return_sequences=False))
lstm_model.add(Dense(units=1))
```

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

```
lstm_model.fit(X_train, y_train, epochs=20, batch_size=32)
```

```
lstm_forecast = lstm_model.predict(X_test)
```

```
lstm_forecast_rescaled = scaler.inverse_transform(lstm_forecast)
```

```
mae_lstm = mean_absolute_error(test_target[time_step:],
lstm_forecast_rescaled)
mse_lstm = mean_squared_error(test_target[time_step:],
lstm_forecast_rescaled)
rmse_lstm = np.sqrt(mse_lstm)
mape_lstm = mean_absolute_percentage_error(test_target[time_step:],
lstm_forecast_rescaled)
r2_lstm = r2_score(test_target[time_step:], lstm_forecast_rescaled)
```

```
plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, lstm_forecast_rescaled, color='red',
label='LSTM Forecast')
plt.title('LSTM Forecast vs True(лише Close, без новин)')
plt.legend()
```

```

plt.savefig(r'images\LSTM Forecast vs Actual(лише Close, без новин).jpeg')
plt.show()

metrics['LSTM_only_Close'] = {
    'mae': mae_lstm,
    'mse': mse_lstm,
    'rmse': rmse_lstm,
    'mape': mape_lstm,
    'r2': r2_lstm
}

gru_model = Sequential()
gru_model.add(GRU(units=50, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
gru_model.add(GRU(units=50, return_sequences=False))
gru_model.add(Dense(units=1))

gru_model.compile(optimizer='adam', loss='mean_squared_error')

gru_model.fit(X_train, y_train, epochs=20, batch_size=32)

gru_forecast = gru_model.predict(X_test)

gru_forecast_rescaled = scaler.inverse_transform(gru_forecast)

mae_gru = mean_absolute_error(test_target[time_step:], gru_forecast_rescaled)
mse_gru = mean_squared_error(test_target[time_step:], gru_forecast_rescaled)
rmse_gru = np.sqrt(mse_gru)
mape_gru = mean_absolute_percentage_error(test_target[time_step:],
gru_forecast_rescaled)
r2_gru = r2_score(test_target[time_step:], gru_forecast_rescaled)

plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, gru_forecast_rescaled,
color='purple', label='GRU Forecast')
plt.title('GRU Forecast vs True(лише Close, без новин)')
plt.legend()
plt.savefig(r'images\GRU Forecast vs Actual(лише Close, без новин).jpeg')
plt.show()

metrics['GRU_only_Close'] = {
    'mae': mae_gru,
    'mse': mse_gru,
    'rmse': rmse_gru,
    'mape': mape_gru,
    'r2': r2_gru
}

print(metrics)

```

```

import pmdarima as pm

model = pm.auto_arma(train_target, seasonal=False, stepwise=True,
trace=True)
model.fit(train_target)

forecast = model.predict(n_periods=len(test_target))

# Вычисление метрик
mae_arma = mean_absolute_error(test_target, forecast)
mse_arma = mean_squared_error(test_target, forecast)
rmse_arma = np.sqrt(mse_arma)
mape_arma = mean_absolute_percentage_error(test_target, forecast)
r2_arma = r2_score(test_target, forecast)

print(f"MAE: {mae_arma}")
print(f"MSE: {mse_arma}")
print(f"RMSE: {rmse_arma}")
print(f"MAPE: {mape_arma}")
print(f"R2: {r2_arma}")

# Models using Close with news data

news_with_currencies_grouped[['negative', 'positive', 'important', 'liked', 'disl
iked', 'lol', 'toxic', 'saved', 'comments']] =
news_with_currencies_grouped[['negative', 'positive', 'important', 'liked', 'disl
iked', 'lol', 'toxic', 'saved', 'comments']].astype('float32')
news_with_currencies_grouped = news_with_currencies_grouped.dropna(how='all')
news_resample = news_with_currencies_grouped[['newsDatetime',
'negative', 'positive', 'important', 'liked', 'disliked', 'lol', 'toxic', 'saved', 'c
omments']].resample('D', on='newsDatetime').mean().reset_index()
news_resample = news_resample.dropna()
df2 = pd.merge(news_resample, df, left_on='newsDatetime',
right_on='Timestamp', how='inner')
train_size = int(len(df2) * 0.8)
train, test = df2.iloc[:train_size], df2.iloc[train_size:]

test_exog[:t]

from statsmodels.tsa.statespace.sarimax import SARIMAX
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score

exog_columns = ['negative', 'positive', 'important', 'liked', 'disliked',
'lol', 'toxic', 'saved', 'comments']
exog = df2[exog_columns]

target = df2['Close']

train_size = int(len(df2) * 0.8)
train_target, test_target = target[:train_size], target[train_size:]

```

```

train_exog, test_exog = exog[:train_size], exog[train_size:]

history = [x for x in train_target]

arima_forecast = []
for t in range(len(test_target)):
    model = SARIMAX(history, exog=exog[:train_size+t], order=(2, 1, 2))
    model_fit = model.fit()
    output = model_fit.forecast(exog=test_exog[t:t+1])
    yhat = output.iloc[0]
    arima_forecast.append(yhat)
    obs = test_target.iloc[t]
    history.append(obs)
    print(f'predicted={yhat:.6f}, expected={obs:.6f}')

mae_arima = mean_absolute_error(test_target, arima_forecast)
mse_arima = mean_squared_error(test_target, arima_forecast)
rmse_arima = np.sqrt(mse_arima)
mape_arima = mean_absolute_percentage_error(test_target, arima_forecast)
r2_arima = r2_score(test_target, arima_forecast)

print(f'MAE: {mae_arima}')
print(f'MSE: {mse_arima}')
print(f'RMSE: {rmse_arima}')
print(f'MAPE: {mape_arima}')
print(f'R-squared: {r2_arima}')

plt.figure(figsize=(10, 6))
plt.plot(test_target.index, test_target, color='blue', label='True')
plt.plot(test_target.index, arima_forecast, color='red', label='SARIMAX
Forecast')
plt.title('SARIMAX Forecast vs True(лише Close, з новинами)')
plt.legend()
plt.savefig(r'images\SARIMAX Forecast vs Actual(лише Close, з
новинами).jpeg')
plt.show()

metrics['SARIMAX_only_Close_and_cryptonews'] = {
    'mae': mae_gru,
    'mse': mse_gru,
    'rmse': rmse_gru,
    'mape': mape_gru,
    'r2': r2_gru
}

train_exog_scaled.shape

from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score
import matplotlib.pyplot as plt

```

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

scaler = MinMaxScaler(feature_range=(0, 1))

train_scaled = scaler.fit_transform(train_target.values.reshape(-1, 1))
test_scaled = scaler.transform(test_target.values.reshape(-1, 1))

exog_scaler = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = exog_scaler.fit_transform(train_exog)
test_exog_scaled = exog_scaler.transform(test_exog)

def create_dataset(data, exog_data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        temp_data = data[i:(i + time_step), 0].reshape(-1, 1) # Преобразуем
в столбец
        temp_exog = exog_data[i:(i + time_step)].mean(axis=0).reshape(-1, 1)
# Это уже 2D, так что можно объединить
        X.append(np.vstack((temp_data, temp_exog))) # Объединяем временные
данные с экзогенными
        y.append(data[i + time_step, 0]) # Целевое значение
    return np.array(X), np.array(y)

time_step = 100

X_train, y_train = create_dataset(train_scaled, train_exog_scaled, time_step)
X_test, y_test = create_dataset(test_scaled, test_exog_scaled, time_step)

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

lstm_model = Sequential()
lstm_model.add(LSTM(units=200, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
lstm_model.add(LSTM(units=100, return_sequences=False))
lstm_model.add(Dense(units=1))

lstm_model.compile(optimizer='adam', loss='mean_squared_error')

lstm_model.fit(X_train, y_train, epochs=20, batch_size=32)

lstm_forecast = lstm_model.predict(X_test)

lstm_forecast_rescaled = scaler.inverse_transform(lstm_forecast)

mae_lstm = mean_absolute_error(test_target[time_step:],
lstm_forecast_rescaled)
mse_lstm = mean_squared_error(test_target[time_step:],
lstm_forecast_rescaled)
rmse_lstm = np.sqrt(mse_lstm)
mape_lstm = mean_absolute_percentage_error(test_target[time_step:],
lstm_forecast_rescaled)
r2_lstm = r2_score(test_target[time_step:], lstm_forecast_rescaled)

```

```
plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, lstm_forecast_rescaled, color='red',
label='LSTM Forecast')
plt.title('LSTM Forecast vs True(лише Close, з новинами)')
plt.legend()
plt.savefig(r'images\LSTM Forecast vs Actual(лише Close, з новинами).jpeg')
plt.show()
```

```
metrics['LSTM_only_Close_and_cryptonews'] = {
    'mae': mae_lstm,
    'mse': mse_lstm,
    'rmse': rmse_lstm,
    'mape': mape_lstm,
    'r2': r2_lstm
}
```

```
plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, lstm_forecast_rescaled, color='red',
label='LSTM Forecast')
plt.title('LSTM Forecast vs True(лише Close, з новинами)')
plt.legend()
plt.savefig(r'images\LSTM Forecast vs Actual(лише Close, з новинами).jpeg')
plt.show()
```

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense
```

```
scaler = MinMaxScaler(feature_range=(0, 1))
```

```
train_scaled = scaler.fit_transform(train_target.values.reshape(-1, 1))
test_scaled = scaler.transform(test_target.values.reshape(-1, 1))
```

```
exog_scaler = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = exog_scaler.fit_transform(train_exog)
test_exog_scaled = exog_scaler.transform(test_exog)
```

```
def create_dataset(data, exog_data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        temp_data = data[i:(i + time_step), 0].reshape(-1, 1)
        temp_exog = exog_data[i:(i + time_step)].mean(axis=0).reshape(-1, 1)
        X.append(np.vstack((temp_data, temp_exog)))
```

```

        y.append(data[i + time_step, 0])
    return np.array(X), np.array(y)

time_step = 100

X_train, y_train = create_dataset(train_scaled, train_exog_scaled, time_step)
X_test, y_test = create_dataset(test_scaled, test_exog_scaled, time_step)

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

gru_model = Sequential()
gru_model.add(GRU(units=200, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
gru_model.add(GRU(units=100, return_sequences=False))
gru_model.add(Dense(units=1))

gru_model.compile(optimizer='adam', loss='mean_squared_error')

gru_model.fit(X_train, y_train, epochs=20, batch_size=32)

gru_forecast = gru_model.predict(X_test)

gru_forecast_rescaled = scaler.inverse_transform(gru_forecast)

mae_gru = mean_absolute_error(test_target[time_step:], gru_forecast_rescaled)
mse_gru = mean_squared_error(test_target[time_step:], gru_forecast_rescaled)
rmse_gru = np.sqrt(mse_gru)
mape_gru = mean_absolute_percentage_error(test_target[time_step:],
gru_forecast_rescaled)
r2_gru = r2_score(test_target[time_step:], gru_forecast_rescaled)

plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, gru_forecast_rescaled, color='red',
label='GRU Forecast')
plt.title('GRU Forecast vs True(лише Close, з новинами)')
plt.legend()
plt.savefig(r'images\GRU Forecast vs Actual(лише Close, з новинами).jpeg')
plt.show()

metrics['GRU_only_Close_and_cryptonews'] = {
    'mae': mae_gru,
    'mse': mse_gru,
    'rmse': rmse_gru,
    'mape': mape_gru,
    'r2': r2_gru
}

metrics

# Models Close only with label encoded features

```

```

news_with_currencies_grouped[['negative','positive','important','liked','disl
iked','lol','toxic','saved','comments'] + [f"sentence_vector_{i}" for i in
range(384)]] =
news_with_currencies_grouped[['negative','positive','important','liked','disl
iked','lol','toxic','saved','comments'] + [f"sentence_vector_{i}" for i in
range(384)]].astype('float32')
news_with_currencies_grouped = news_with_currencies_grouped.dropna(how='all')
news_resample = news_with_currencies_grouped[[f"sentence_vector_{i}" for i in
range(384)]+'newsDatetime',
'negative','positive','important','liked','disliked','lol','toxic','saved','c
omments']].resample('D', on='newsDatetime').mean().reset_index()
news_resample = news_resample.dropna()
df2 = pd.merge(news_resample, df, left_on='newsDatetime',
right_on='Timestamp', how='inner')
train_size = int(len(df2) * 0.8)
train, test = df2.iloc[:train_size], df2.iloc[train_size:]

```

```

from statsmodels.tsa.statespace.sarimax import SARIMAX
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score

```

```

exog_columns =
['negative','positive','important','liked','disliked','lol','toxic','saved','
comments'] + [f"sentence_vector_{i}" for i in range(384)]
exog = df2[exog_columns]

```

```
target = df2['Close']
```

```

train_size = int(len(df2) * 0.8)
train_target, test_target = target[:train_size], target[train_size:]
train_exog, test_exog = exog[:train_size], exog[train_size:]

```

```
history = [x for x in train_target]
```

```

arma_forecast = []
for t in range(len(test_target)):
    model = SARIMAX(history, exog=exog[:train_size+t], order=(2, 1, 2))
    model_fit = model.fit()
    output = model_fit.forecast(exog=test_exog[t:t+1])
    yhat = output.iloc[0]
    arma_forecast.append(yhat)
    obs = test_target.iloc[t]
    history.append(obs)
    print(f'predicted={yhat:.6f}, expected={obs:.6f}')

```

```

mae_arma = mean_absolute_error(test_target, arma_forecast)
mse_arma = mean_squared_error(test_target, arma_forecast)
rmse_arma = np.sqrt(mse_arma)
mape_arma = mean_absolute_percentage_error(test_target, arma_forecast)
r2_arma = r2_score(test_target, arma_forecast)

```

```

print(f'MAE: {mae_arima}')
print(f'MSE: {mse_arima}')
print(f'RMSE: {rmse_arima}')
print(f'MAPE: {mape_arima}')
print(f'R-squared: {r2_arima}')

plt.figure(figsize=(10, 6))
plt.plot(test_target.index, test_target, color='blue', label='True')
plt.plot(test_target.index, arima_forecast, color='red', label='SARIMAX
Forecast')
plt.title('SARIMAX Forecast vs True')
plt.legend()
plt.show()

from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error, r2_score
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

scaler = MinMaxScaler(feature_range=(0, 1))

train_scaled = scaler.fit_transform(train_target.values.reshape(-1, 1))
test_scaled = scaler.transform(test_target.values.reshape(-1, 1))

exog_scaler = MinMaxScaler(feature_range=(0, 1))
train_exog_scaled = exog_scaler.fit_transform(train_exog)
test_exog_scaled = exog_scaler.transform(test_exog)

def create_dataset(data, exog_data, time_step=1):
    X, y = [], []
    for i in range(len(data) - time_step):
        temp_data = data[i:(i + time_step), 0].reshape(-1, 1) # Преобразуем
в столбец
        temp_exog = exog_data[i:(i + time_step)].mean(axis=0).reshape(-1, 1)
# Это уже 2D, так что можно объединить
        X.append(np.vstack((temp_data, temp_exog))) # Объединяем временные
данные с экзогенными
        y.append(data[i + time_step, 0]) # Целевое значение
    return np.array(X), np.array(y)

time_step = 100

X_train, y_train = create_dataset(train_scaled, train_exog_scaled, time_step)
X_test, y_test = create_dataset(test_scaled, test_exog_scaled, time_step)

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

lstm_model = Sequential()
lstm_model.add(LSTM(units=200, return_sequences=True,
input_shape=(X_train.shape[1], 1)))

```

```

lstm_model.add(LSTM(units=100, return_sequences=False))
lstm_model.add(Dense(units=1))

lstm_model.compile(optimizer='adam', loss='mean_squared_error')

lstm_model.fit(X_train, y_train, epochs=20, batch_size=32)

lstm_forecast = lstm_model.predict(X_test)

lstm_forecast_rescaled = scaler.inverse_transform(lstm_forecast)

mae_lstm = mean_absolute_error(test_target[time_step:],
lstm_forecast_rescaled)
mse_lstm = mean_squared_error(test_target[time_step:],
lstm_forecast_rescaled)
rmse_lstm = np.sqrt(mse_lstm)
mape_lstm = mean_absolute_percentage_error(test_target[time_step:],
lstm_forecast_rescaled)
r2_lstm = r2_score(test_target[time_step:], lstm_forecast_rescaled)

plt.figure(figsize=(10, 6))
plt.plot(test_target[time_step:].index, test_target[time_step:],
color='blue', label='True')
plt.plot(test_target[time_step:].index, lstm_forecast_rescaled, color='red',
label='LSTM Forecast')
plt.title('LSTM Forecast vs True')
plt.legend()
plt.show()

metrics['LSTM_only_Close_all_news_'+label_data'] = {
    'mae': mae_lstm,
    'mse': mse_lstm,
    'rmse': rmse_lstm,
    'mape': mape_lstm,
    'r2': r2_lstm
}

metrics

```