

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

**НН Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено:

Завідувач кафедри

_____ Оксана ТИМОЩУК

«__» _____ 2023 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системний аналіз і управління»

спеціальності 124 «Системний аналіз»

на тему: «Прогнозування часових рядів на прикладі криптовалют»

Виконала:

студентка 4 курсу, групи КА-95

Гриценко Олександра Володимирівна _____

Керівник:

ст. викладач, к.т.н. Савастьянов Володимир Володимирович _____

Консультант з нормоконтролю:

к.ф. -м.н. Статкевич Віталій Михайлович _____

Консультант з економічного розділу:

доц. к.е.н. Рощина Надія Василівна _____

Рецензент:

ст. викладач, к.т.н. Терентьев Олександр Михайлович _____

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

Студентка _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
НН Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Оксана ТИМОЩУК

«__» травня 2023 р.

ЗАВДАННЯ

на дипломну роботу студентці

Гриценко Олександрі Володимирівні

1. Тема роботи «Прогнозування часових рядів на прикладі криптовалют», керівник роботи ст. викладач, к.т.н. Савастьянов Володимир Володимирович, затверджені наказом по університету від 30 травня 2023 р. № 2065-с
2. Термін подання студентом роботи 12.06.2023.
3. Вихідні дані до роботи
 1. Операційна система Linux Ubuntu 20.04
 2. Частота процесора 3.5 ГГц
 3. Мова програмування Python 3.8
 4. Середовище розробки - Jupyter Notebook
 5. Бібліотеки, що використовувалися: NumPy, Pandas, Matplotlib, Keras, TensorFlow, Pycaret, Prophet, Arima

4. Зміст роботи: аналіз області прогнозування часових рядів, прогнозування курсу криптовалют, обробка та аналіз даних про курс криптовалют, розробка системи прогнозування курсу криптовалют, функціонально-вартісний аналіз.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

6. Консультанти розділів роботи^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н.В., доцент		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми ДР	16.04.2023 – 30.04.2023	виконано
2	Ознайомлення зі структурою БДР згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	16.04.2023 – 22.04.2023	виконано
3	Ознайомлення з ДСТУ 3008-95 та стандарти ЄСПД	16.04.2023 – 22.04.2023	виконано
4	Проведення дослідження за темою БДР під керівництвом керівника	23.04.2023 – 08.05.2023	виконано
5	Завершення роботи над першим варіантом частини БДР	08.05.2023 – 12.05.2023	виконано
6	Проведення роботи над експериментальною частиною БДР	08.05.2023 – 16.05.2023	виконано
7	Проведення роботи над програмним продуктом	16.05.2023 – 24.05.2023	виконано
8	Оформлення БДР та аналіз отриманих результатів	16.05.2023 – 24.05.2023	виконано

Студент

Олександра ГРИЦЕНКО

Керівник

Володимир САВАСТЬЯНОВ

РЕФЕРАТ

Дипломна робота: 126 с., 27 рис., 12 табл., 52 джерела, 2 додатка.

ПРОГНОЗУВАННЯ, ЧАСОВІ РЯДИ, LSTM, PROPHET, ARIMA, EXPONENTIAL SMOOTHING.

Тема: Прогнозування часових рядів на прикладі криптовалют

У роботі розглянуті різні моделі прогнозування часових рядів на прикладі курсів криптовалют.

Об'єкт дослідження: застосування методів прогнозування часових рядів для передбачення курсів криптовалют.

Предмет дослідження: методи прогнозування часових рядів та їх застосування.

Мета роботи: розробка програмного забезпечення для прогнозування курсів криптовалют із використанням часових рядів.

Створено програмне забезпечення на мові програмування Python у вигляді декількох моделей для прогнозування цін на криптовалюти. Для розробки моделей був використаний набір даних із цінами на криптовалюти із сайту Kaggle.

ABSTRACT

Diploma thesis: 126 p., 27 fig., 12 tabl., 52 sources, 2 appendices.

FORECASTING, TIME SERIES, LSTM, PROPHET, ARIMA, EXPONENTIAL SMOOTHING.

Theme: Forecasting time series on the example of cryptocurrencies

The paper considers various models of time series forecasting on the example of cryptocurrency rates.

Object of research: application of time series forecasting methods for predicting cryptocurrency rates.

Subject of research: methods of time series forecasting and their application.

Purpose: to develop software for forecasting cryptocurrency rates using time series.

Software in the form of several models for forecasting cryptocurrency prices were implemented in Python programming language. To develop the models, a dataset with cryptocurrency prices from the Kaggle website was used.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ОБЛАСТІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ...	9
1.1 Визначення та особливості часових рядів.....	9
1.1.1 Визначення поняття "часовий ряд"	9
1.1.2 Властивості та особливості часових рядів, які роблять їх привабливими для прогнозування	10
1.1.3 Різниця між часовими рядами та звичайними даними	11
1.2 Застосування прогнозування часових рядів у фінансових ринках	12
12.1 Роль прогнозування в інвестуванні та управлінні ризиками на фінансових ринках.....	12
1.2.2 Особливості прогнозування фінансових ринків.....	13
1.2.3 Потенційні виклики та переваги прогнозування фінансових часових рядів.....	14
1.3 Висновки до розділу	16
РОЗДІЛ 2 ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ	18
2.1 Попередній досвід прогнозування курсу криптовалют	18
2.1.1 Огляд досліджень, що вже проводились у галузі прогнозування курсу криптовалют	18
2.1.2 Аналіз розроблених моделей та методів прогнозування криптовалютних курсів.....	19
2.1.3 Виявлення сильних і слабких сторін попередніх досліджень.....	21
2.2 Методи прогнозування курсу криптовалют.....	22
2.2.3 Гібридні методи, які поєднують фундаментальний та технічний аналіз.....	25

2.2.4	Методи машинного навчання та штучного інтелекту для прогнозування криптовалютних курсів	27
2.3	Проблеми та виклики прогнозування курсу криптовалют.....	28
2.3.1	Висока волатильність та непередбачуваність ринку криптовалют	28
2.3.2	Вплив новин та інформації на курс криптовалют	29
2.3.3	Обмежена історична інформація та недостатність даних для прогнозування.....	31
2.3.4	Проблеми зі збором, чисткою та обробкою даних для прогнозування криптовалютних курсів	32
2.4	Особливості прогнозування криптовалютних курсів	33
2.4.1	Вплив масштабних змін та глобальних подій на курс криптовалют	33
2.4.2	Роль алгоритмічної торгівлі та торгових ботів у прогнозуванні криптовалют.....	34
2.4.3	Аналіз соціальних медіа та громадської думки для прогнозування	35
2.5	Висновки до розділу	36
РОЗДІЛ 3 ОБРОБКА ТА АНАЛІЗ ДАНИХ ПРО КУРС КРИПТОВАЛЮТ		
.....		38
1.1	Огляд обраного набору даних	38
1.2	Аналіз доступних даних.....	42
1.3	Обробка даних для LSTM.....	45
1.4	Обробка даних для ARIMA, Prophet та ExpSmoothing	47
1.4	Висновки до розділу	47
РОЗДІЛ 4 РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ		
.....		49
4.1	Вибір методів прогнозування.....	49

	7
4.1.1 Метод Prophet.....	49
4.1.2 Метод Auto ARIMA.....	53
4.1.3 Метод Exponential Smoothing	54
4.1.4 Метод LSTM	57
4.2 Впровадження методів	59
4.2.1 LSTM	59
4.2.2 ARIMA.....	65
4.2.3 Prophet.....	68
4.2.4 Exponential Smoothing	71
4.4 Аналіз результатів	73
4.5 Висновки до розділу	75
5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ	76
5.1 Постановка задачі	77
5.2 Функції програмного продукту	78
5.3 Вибір параметрів.....	82
5.4 Експертний аналіз.....	84
5.5 Аналіз якості реалізації варіантів функцій	88
5.6 Економічний аналіз розробки.....	91
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	95
.....	96
5.8 Висновки до розділу	97
ВИСНОВКИ.....	98
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	99
ДОДАТОК А ЛІСТИНГ	105

ВСТУП

У сучасному світі, де цифрові технології та фінансові ринки знаходяться в постійній динаміці, прогнозування стає все важливішим інструментом для прийняття обґрунтованих рішень. Особливо актуальним є прогнозування часових рядів, таких як криптовалюти, які є одними з найбільш волатильних активів на фінансових ринках.

Тема цієї дипломної роботи - "Прогнозування часових рядів на прикладі криптовалют". Метою даного дослідження є розробка та порівняння моделей прогнозування для криптовалютних ринків, з метою отримання точних та надійних прогнозів цінової динаміки криптовалют.

У цій роботі буде проведений аналіз різних методів та підходів до прогнозування часових рядів, таких як аналіз часових рядів, статистичні методи, машинне навчання та штучні нейронні мережі. Будуть розглянуті та порівняні різні моделі, такі як ARIMA, LSTM, Exp Smoothing та Prophet, що комбінують кілька підходів для отримання кращих прогнозів.

Для проведення дослідження будуть використані реальні дані криптовалютних ринків, які будуть піддані аналізу та моделюванню. Результати дослідження допоможуть встановити ефективність різних моделей прогнозування та їх придатність для прогнозування цінової динаміки криптовалют.

Ця дипломна робота важлива з точки зору розвитку фінансової аналітики та інструментів прогнозування. Прогнозування часових рядів криптовалют може мати велике значення для трейдерів, інвесторів та фінансових установ, які прагнуть приймати обґрунтовані рішення в умовах швидкозмінного ринку.

Далі в роботі буде представлений огляд літератури, методологія дослідження, опис моделей та експерименти, а також аналіз та обговорення результатів. Заключна частина містить висновки та рекомендації щодо подальших напрямів дослідження в даній області.

В першому розділі буде розглянута область прогнозування часових рядів в цілому.

В другому розділі буде розглянуто методи передбачення криптовалют, їх переваги та недолі та потенційні шляхи покращення.

В третьому розділі буде проведено аналіз та обробка даних для обраних моделей для передбачення.

Після цього в четвертому розділі буде створено моделі для прогнозування та порівняно результати їх роботи.

В п'ятому розділі буде проведений функціонально-вартісний аналіз, що дозволяє знайти найкращі шляхи для реалізації системи, що була досліджена в даній роботі та оцінити ціну її розробки.

РОЗДІЛ 1 АНАЛІЗ ОБЛАСТІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1 Визначення та особливості часових рядів

1.1.1 Визначення поняття "часовий ряд"

Часовий ряд - це набір спостережень, зібраних через регулярні проміжки часу [1], послідовність точок даних, вимірних, як правило, в послідовні моменти часу, що використовуються для розуміння поведінки системи в часі. Аналіз часових рядів передбачає аналіз цієї послідовності даних для вилучення значущої статистики, закономірностей та інших характеристик даних [1].

Часовий ряд може бути одновимірним або багатовимірним. Одновимірні часові ряди включають одну змінну, виміряну в часі, тоді як багатовимірні часові ряди включають дві або більше змінних [2].

Дані часових рядів можна аналізувати за допомогою різних статистичних методів, включаючи ковзні середні, експоненціальне згладжування та автокореляцію. Ці методи можуть допомогти виявити закономірності та тенденції в даних, що полегшує прогнозування майбутніх значень [2]. Стаціонарність є важливим поняттям в аналізі часових рядів, оскільки вона є методом роботи багатьох моделей прогнозування часових рядів. Стаціонарний процес - це процес, в якому статистичні властивості процесу, що створює часовий ряд, є постійними в часі [3]. Стаціонарні процеси легше моделювати та прогнозувати, ніж нестационарні.

Всього існує кілька типів нестационарних процесів: тренд, волатильність, сезонність і шум. Тренд - це послідовне збільшення або зменшення даних у часі. Волатильність - це ступінь варіації даних у часі. Сезонність - закономірність, яка повторюється через регулярні проміжки часу, наприклад, щотижня або щомісяця. Шум - це випадкові коливання в даних, які не можна віднести до якоїсь конкретної закономірності і тенденції [3].

Отже, часовий ряд - це набір спостережень, зібраних через регулярні проміжки часу, а аналіз часових рядів передбачає аналіз цієї послідовності точок даних для вилучення значущої статистики, закономірностей та інших характеристик даних. Стаціонарність є важливим поняттям в аналізі часових рядів, і існують різні типи нестаціонарних процесів, включаючи тренд, волатильність, сезонність і шум.

1.1.2 Властивості та особливості часових рядів, які роблять їх привабливими для прогнозування

Часові ряди мають декілька властивостей та особливостей, що роблять їх привабливими для прогнозування. Вкажемо деякі з найважливіших особливостей рядів.

1. Часова впорядкованість, що означає, що спостереження збираються через рівні проміжки часу [4]. Така впорядкованість дозволяє виявити закономірності в даних, які можуть бути використані для прогнозування майбутніх значень [5].
2. Дані часових рядів часто демонструють тенденції, сезонність та інші закономірності, які можна моделювати та використовувати для прогнозування [6]. Наприклад, часовий ряд може демонструвати щоденні, щотижневі, щомісячні або річні закономірності, які можна зафіксувати та використати для прогнозування майбутніх значень [5].
3. Дані часових рядів можна аналізувати за допомогою різноманітних статистичних методів, таких як ковзні середні, експоненціальне згладжування та автокореляція, які можуть допомогти виявити закономірності та тенденції в даних.
4. Дані часових рядів можна використовувати для короткострокових і довгострокових прогнозів, залежно від довжини часового ряду та горизонту прогнозування.

5. Дані часових рядів можна використовувати для прогнозування широкого спектру змінних, таких як продажі, ціни на акції та погодні умови.
6. Дані часових рядів можна комбінувати з іншими джерелами даних, такими як демографічні дані або економічні показники, щоб підвищити точність прогнозів.
7. Дані часових рядів можна трансформувати та попередньо обробити, щоб видалити шум і викиди, а також зробити дані більш придатними для моделювання [2].

Отже, часові ряди мають ряд властивостей і особливостей, які роблять їх привабливими для прогнозування, включаючи часову впорядкованість, закономірності і тенденції, різноманітні статистичні методи, короткострокове і довгострокове прогнозування, прогнозування широкого спектру змінних, поєднання з іншими джерелами даних і методи попередньої обробки.

1.1.3 Різниця між часовими рядами та звичайними даними

Дані часових рядів відрізняються від звичайних даних декількома ознаками. Ключові відмінності наведені нижче:

- Дані часових рядів збираються через регулярні проміжки часу, тоді як регулярні дані можуть збиратися в будь-який момент часу і можуть не мати часової впорядкованості [7].
- Дані часових рядів часто використовуються для прогнозування, тоді як регулярні дані можуть бути використані для широкого кола аналітичних цілей, таких як описова статистика, перевірка гіпотез та машинне навчання [7].
- Дані часових рядів часто демонструють закономірності та тенденції, які є унікальними для певного часу, такі як сезонність і тренд, тоді як регулярні дані можуть не демонструвати цих закономірностей [8].

- Дані часових рядів часто залежать від попередніх значень у часовому ряді, тоді як звичайні дані можуть не залежати від попередніх значень [8].
- Дані часових рядів чутливі до змін у часі, тоді як звичайні дані можуть не бути чутливими до змін у часі.
- Дані часових рядів часто використовуються в економіці, фінансах та інших галузях, де важлива часова впорядкованість, тоді як регулярні дані можуть використовуватися в широкому спектрі галузей [9].

Отже, дані часових рядів відрізняються від звичайних даних кількома способами, включаючи їхню часову впорядкованість, використання для прогнозування, унікальні закономірності та тенденції, залежність від попередніх значень, чутливість до змін у часі та використання в конкретних галузях.

1.2 Застосування прогнозування часових рядів у фінансових ринках

1.2.1 Роль прогнозування в інвестуванні та управлінні ризиками на фінансових ринках

Прогнозування відіграє важливу роль в інвестуванні та управлінні ризиками на фінансових ринках. Вкажемо деякі способи використання прогнозування.

1. Виявлення тенденцій і закономірностей на фінансових ринках, які можуть бути використані для прийняття інвестиційних рішень [10]. Наприклад, фундаментальний аналіз передбачає аналіз фінансових даних для визначення внутрішньої вартості цінного паперу, яка може бути використана для прийняття інвестиційних рішень. Технічний аналіз, з іншого боку, передбачає використання історичних даних про цінні папери для прогнозування майбутніх цін на основі припущення, що ціни на акції визначаються ринковими силами і що історія має тенденцію повторюватись.

2. Прогнозування використовується для оцінки ризику та невизначеності на фінансових ринках. Прогнозуючи майбутні ціни та дохідність, інвестори та ризик-менеджери можуть визначити потенційні ризики та вжити заходів для їх зменшення [11]. Наприклад, диверсифікація портфеля, розподіл активів та визначення розміру позиції - це стратегії, які можна використовувати для управління ризиками на фінансових ринках.
3. Прогнозування використовується для прийняття короткострокових і довгострокових інвестиційних рішень [10]. Короткострокові прогнози можна використовувати для здійснення операцій на основі ринкових тенденцій і моделей, тоді як довгострокові прогнози можна використовувати для здійснення інвестицій в акції, облігації та інші цінні папери [10].
4. Прогнозування використовується політиками та регуляторами для прийняття відповідних рішень та вжиття корегувальних заходів. Прогнозуючи майбутні економічні тенденції та ринкові умови, політики можуть вжити заходів для сприяння економічному зростанню та стабільності [10].

Отже, прогнозування відіграє важливу роль в інвестуванні та управлінні ризиками на фінансових ринках, визначаючи тенденції та закономірності, оцінюючи ризики та невизначеність, приймаючи короткострокові та довгострокові інвестиційні рішення, а також надаючи інформацію для формування політики та регулювання.

1.2.2 Особливості прогнозування фінансових ринків

Прогнозування фінансових ринків має деякі унікальні особливості порівняно з іншими видами прогнозування. Деякі з ключових особливостей прогнозування фінансових ринків містять:

- Передбачення майбутньої вартості фінансових активів, таких як акції, облигації та валюти, серед іншого [12].
- Прогнозування фінансових ринків базується на широкому спектрі джерел, включаючи історичні ціни, економічні показники, політичні події та новини, серед іншого [12].
- Прогнозування фінансових ринків вимагає розуміння фінансових ринків та їх основних рушійних сил, таких як попит і пропозиція, процентні ставки та ринкові настрої, серед іншого [12].
- Прогнозування фінансових ринків пов'язане з високим ступенем невизначеності та ризику, оскільки на фінансові ринки впливає широкий спектр факторів, які важко передбачити [12].
- Прогнозування фінансових ринків використовується інвесторами, трейдерами та ризик-менеджерами для прийняття інвестиційних рішень, управління ризиками та хеджування від волатильності ринку [13].
- Прогнозування фінансових ринків ґрунтується на різноманітних методах, таких як фундаментальний аналіз, технічний аналіз та кількісний аналіз [13].
- Прогнозування фінансових ринків вимагає доступу до своєчасних і точних даних, а також передових аналітичних інструментів і програмного забезпечення для прийняття обґрунтованих рішень [13].

Таким чином, прогнозування фінансових ринків має ряд унікальних особливостей порівняно з іншими видами прогнозування, серед яких фокус на фінансових активах, широкий спектр джерел даних, необхідність знання ринку, високий ступінь невизначеності та ризику, використання інвесторами та ризик-менеджерами, використання декількох методів прогнозування, а також потреба в сучасних аналітичних інструментах та програмному забезпеченні.

1.2.3 Потенційні виклики та переваги прогнозування фінансових часових рядів

Прогнозування фінансових часових рядів має як потенційні виклики, так і переваги.

Потенційні виклики:

- Високий ступінь невизначеності та ризику на фінансових ринках може зробити прогнозування складним і непередбачуваним [14].
- Дані фінансових часових рядів можуть бути складними та мінливими, що ускладнює їх точне моделювання [15].
- На дані фінансових часових рядів може впливати широкий спектр факторів, таких як економічні показники, політичні події та новини, серед інших, які важко передбачити [15].
- Прогнозування фінансових часових рядів вимагає доступу до своєчасних і точних даних, а також передових аналітичних інструментів і програмного забезпечення для прийняття обґрунтованих рішень [15].
- На прогнозування фінансових часових рядів можуть впливати проблеми з якістю даних, такі як переривчастість даних, розрідженість, відсутність часових міток та прогалини, що може призвести до отримання "сміттєвих" результатів [16].

Потенційні переваги:

- Прогнозування фінансових часових рядів може допомогти інвесторам та ризик-менеджерам приймати обґрунтовані інвестиційні рішення, управляти ризиками та хеджувати від волатильності ринку [15].
- Прогнозування фінансових часових рядів може допомогти політикам та регуляторам приймати відповідні рішення та вживати коригувальних заходів для сприяння економічному зростанню та стабільності [15].
- Прогнозування фінансових часових рядів може допомогти підприємствам та організаціям приймати стратегічні рішення щодо

ціноутворення, управління запасами та розподілу ресурсів, серед іншого [14].

- Прогнозування фінансових часових рядів може допомогти підприємствам та організаціям покращити свою діяльність та ефективність шляхом виявлення тенденцій та закономірностей у фінансових даних [14].
- Прогнозування фінансових часових рядів може забезпечити конкурентну перевагу, дозволяючи підприємствам та організаціям передбачати ринкові тенденції та швидко реагувати на зміни ринкових умов [15].

Отже, прогнозування фінансових часових рядів має як потенційні виклики, так і переваги. Хоча прогнозування фінансових часових рядів може бути складним і непередбачуваним через високий ступінь невизначеності та ризику на фінансових ринках, воно також може допомогти інвесторам, ризик-менеджерам, політикам і підприємствам приймати обґрунтовані рішення, управляти ризиками та покращувати свою діяльність і результати.

1.3 Висновки до розділу

Отже, в ході роботи над першим розділом було визначено, що часові ряди мають властивості і особливості, які роблять їх привабливими для прогнозування. Ці властивості включають часову впорядкованість, закономірності і тенденції, різноманітні статистичні методи, короткострокове і довгострокове прогнозування, прогнозування широкого спектру змінних, поєднання з іншими джерелами даних і методи попередньої обробки.

Також було визначено, що часові ряди відрізняються від звичайних даних кількома способами. Вони мають часову впорядкованість, унікальні закономірності та тенденції, залежність від попередніх значень, чутливість до змін у часі та використання в конкретних галузях.

Прогнозування часових рядів відіграє важливу роль в інвестуванні та управлінні ризиками на фінансових ринках. Воно допомагає визначати тенденції та закономірності, оцінювати ризики та невизначеність, приймати короткострокові та довгострокові інвестиційні рішення, а також надає інформацію для формування політики та регулювання.

Прогнозування фінансових часових рядів має свої унікальні особливості порівняно з іншими видами прогнозування. Це включає фокус на фінансових активах, широкий спектр джерел даних, необхідність знання ринку, високий ступінь невизначеності та ризику, використання інвесторами та ризик-менеджерами, використання декількох методів прогнозування, а також потребу в сучасних аналітичних інструментах та програмному забезпеченні.

Прогнозування фінансових часових рядів може бути складним і непередбачуваним через високий ступінь невизначеності та ризику на фінансових ринках. Однак, воно має переваги, оскільки допомагає інвесторам, ризик-менеджерам, політикам і підприємствам приймати обґрунтовані рішення, управляти ризиками та покращувати свою діяльність і результати.

Узагальнюючи, можна сказати, що аналіз області прогнозування часових рядів показує, що вони мають унікальні властивості і особливості, що роблять їх цікавими для прогнозування. Прогнозування часових рядів, зокрема фінансових, відіграє важливу роль у прийнятті рішень в інвестиціях, управлінні ризиками та регулюванні. Хоча прогнозування може бути складним, воно може принести значну користь, якщо використовується правильно і з урахуванням усіх унікальних особливостей цих рядів.

РОЗДІЛ 2 ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

2.1 Попередній досвід прогнозування курсу криптовалют

2.1.1 Огляд досліджень, що вже проводились у галузі прогнозування курсу криптовалют

Прогнозування криптовалют завжди цікавило інвесторів та аналітиків. На точність прогнозування цін на криптовалюту впливають різні фактори, включаючи знання криптології та токеноміки, обсяги торгів, ринкову капіталізацію, попит і пропозицію, використання в якості платіжних tokenів, похідних продуктів, графіки, історичні тенденції та метрики [17].

Аналітики використовують алгоритми для прогнозування цін, а вищезазначені фактори є вхідними даними для цих алгоритмів та моделей прогнозування. Алгоритми можуть базуватися на машинному навчанні, штучному інтелекті або бути ручними. Короткостроковий аналіз цін в основному базується на технічних графіках та аналітичних методах [17].

Довгострокові прогнози цін на біткоїн з точністю до 55% поширені при використанні моделей машинного навчання. Моделі машинного навчання працюють краще, ніж статистичні методи в довгостроковій перспективі. Іноді точність прогнозування може досягати 70 і 80% [17].

В останні роки зростає інтерес до використання штучного інтелекту (ШІ) і машинного навчання для прогнозування криптовалют. ШІ і машинне навчання можуть бути використані для інвестування в криптовалюту, щоб підвищити точність торгових прогнозів, подолавши деяку невизначеність інвестування в криптовалюту [18].

Існує багато веб-сайтів та інструментів, які можна використовувати для оцінки майбутньої ціни криптовалюти, включаючи Finder.com, Fxstreet, Walletinvestor, Longforecast та Bitgur [17].

Прогнозування цін і торгівля активами є високоризикованою справою, тому важливо бути в курсі ринкових умов і використовувати прогнози цін поряд з іншими фінансовими порадами, оцінкою ризиків, інформацією та факторами при інвестуванні [17].

Вартість біткойна визначається щосекунди, день за днем, ринком, що ніколи не спить. Оскільки вартість криптовалюти визначається відкритим ринком, це створює унікальні проблеми, пов'язані з волатильністю, з якими більшість валют не стикаються. Не заглиблюючись у терни технічного аналізу, більшість трейдерів і ринкових аналітиків зазвичай сподіваються розпізнати загальний настрій ринку [19].

Що стосується прогнозів цін на біткойн, то аналітики передбачають, що ціна зросте приблизно до 500 000 доларів за монету в 2025 році і до 1 мільйона доларів за монету в 2030 році, хоча це лише прогнози, а ринок криптовалют є дуже волатильним [17].

2.1.2 Аналіз розроблених моделей та методів прогнозування криптовалютних курсів

Було проведено кілька досліджень щодо передбачуваності криптовалют, таких як Bitcoin, Ethereum, Litecoin та Monero, та прибутковості торгових стратегій, розроблених на основі методів машинного навчання, таких як лінійні моделі, випадкові ліси та машини опорних векторів [20]. Моделі перевіряються в період, що характеризується безпрецедентними потрясіннями, і тестуються в період ведмежих ринків, що дозволяє оцінити, чи є прогнози правильними, навіть якщо напрямок ринку змінюється між періодами перевірки і тестування.

Моделі у згаданих дослідженнях перевіряються та тестуються на історичних даних для оцінки їхньої ефективності. Точність прогнозування досить сильно відрізняється для різних моделей і криптовалют, і не існує чіткої закономірності, яка б дозволила зробити висновок про те, яка модель є кращою або яка криптовалюта є

найбільш передбачуваною в періодах валідації і тестування. Однак, в цілому, точність прогнозування окремих моделей здається низькою, якщо порівнювати з іншими подібними дослідженнями [20].

Основною метою згаданих досліджень не є широке тестування альтернативних стратегій прогнозування і торгівлі, тому немає ніякої гарантії, що використовуються найкращі з доступних методів. Натомість мета є більш скромною, оскільки дослідники просто намагались з'ясувати, чи може машинне навчання в цілому призвести до прибуткових стратегій на ринку криптовалют, і чи зберігається ця прибутковість, коли ринкові умови змінюються і враховуються більш реалістичні ринкові особливості. Використання більш частотних даних, ширшого набору вхідних даних, включаючи більш точні атрибути, такі як індикатори технічного аналізу, а також розгляд ф'ючерсів на біткоїн, де легко створювати короткі позиції, а транзакційні витрати є нижчими, можливо, змогли б призвести до кращого результату [20].

Результати досліджень вказують на те, що найкращими торговими стратегіями є Ensemble 5, застосована до ефіріуму та лайткоїну, яка досягла річного коефіцієнту Шарпа 80,17% та 91,35%, а також річного прибутку, після пропорційних торгових витрат 0,5%, 9,62% та 5,73%, відповідно. Ці позитивні результати підтверджують твердження, що машинне навчання надає надійні методи для вивчення передбачуваності криптовалют і для розробки прибуткових торгових стратегій на цих ринках, навіть за несприятливих ринкових умов [20].

ARIMA вважається найкращою моделлю для прогнозування ціни біткойна на крипторинку з показником RMSE 322,4 та MAE 227,3 [21].

Отже, в наведеній статті, дослідниками було проведено кілька досліджень щодо передбачуваності криптовалют і прибутковості торгових стратегій, розроблених на основі методів машинного навчання. У цих дослідженнях використовуються атрибути торгової та мережевої активності за період з 15 серпня 2015 року по 03 березня 2019 року, а тестова вибірка починається з 13 квітня 2018 року. Точність прогнозування окремих моделей здається низькою в порівнянні з іншими подібними дослідженнями, але найкращою торговою стратегією є Ensemble

5, що була застосована до ethereum і litecoin, яка досягла річного коефіцієнта Шарпа 80,17% і 91,35% і річного прибутку після пропорційних торгових витрат в 0,5%, 9,62% і 5,73% відповідно. ARIMA вважається найкращою моделлю для прогнозування ціни біткойна на крипторинку. Однак мета згаданих досліджень була скромнішою, ніж отримання високих прибутків від трейдингових ботів, тому необхідно провести більше досліджень для тестування альтернативних стратегій прогнозування і торгівлі з використанням більш частотних даних, ширшого набору вхідних даних і розгляду ф'ючерсів на біткоїн.

2.1.3 Виявлення сильних і слабких сторін попередніх досліджень

В цілому, можна виділити деякі сильні та слабкі сторони попередніх досліджень у сфері прогнозування криптовалют.

Сильні сторони:

- використання передових моделей машинного навчання та штучного інтелекту для прогнозування цін на криптовалюту;
- використання історичних даних для перевірки і тестування моделей;
- визначення прибуткових торгових стратегій за допомогою методів машинного навчання;
- оцінка ефективності різних моделей і методів прогнозування курсів криптовалют;
- внесок у суспільне розуміння передбачуваності криптовалют.

Слабкі сторони:

- низька точність прогнозування окремих моделей у порівнянні з іншими подібними дослідженнями;
- обмеження розміру вибірки та відбору;
- обмежена можливість узагальнення результатів на інші популяції або контексти;

- відсутність розгляду більш частотних даних і ширшого набору вхідних даних, включаючи більш точні атрибути, такі як показники технічного аналізу;
- висока волатильність ринку криптовалют, що робить прогнозування цін і спекуляції активами високоризикованою справою.

Важливо враховувати ці сильні і слабкі сторони при оцінці попередніх досліджень і визнати, що необхідні додаткові дослідження для тестування альтернативних стратегій прогнозування і торгівлі з використанням більш досконалих методів і даних.

2.2 Методи прогнозування курсу криптовалют

2.2.1 Фундаментальний аналіз та його застосування у прогнозуванні криптовалют

Фундаментальний аналіз - це метод, який використовується інвесторами на фінансових ринках для оцінки та вивчення змінних, що впливають на вартість. Він спрямований на виявлення реальної вартості токена або криптовалюти, щоб інвестувати на основі різниці між нею та її ринковою вартістю [22]. Цей метод базується на тих самих принципах, що й на інших фінансових ринках, які містять більш традиційні активи, такі як акції. Однак тут менше уваги приділяється фінансовій звітності, балансам або історичним фінансовим показникам компанії. Інвестори, які прагнуть навчитися прогнозувати тенденції крипторинку, можуть розглянути наступні параметри [23]:

- Розмір спільноти навколо валюти, що є важливим показником для досягнення більш широкого прийняття криптовалют
- Метрики блокчейну, які розкривають найважливіші аспекти технології та процесів, що лежать в основі криптовалюти, такі як хешрейт, загальна

сукупна обчислювальна потужність, що використовується для видобутку та обробки транзакцій у блокчейні

- Рівень прийняття криптовалюти користувачами і висвітлення в ЗМІ
- Криптовалютні новини, які можуть надати корисну інформацію про сприйняття криптовалюти громадськістю, а також про плани компаній і партнерства, які можуть сприяти зростанню або падінню криптовалюти.

Також метод фундаментального аналізу потенційно може допомогти у прийнятті довгострокових інвестиційних рішень, допомагаючи трейдерам визначити вартість криптовалюти на основі широкого спектру доступної інформації [23]. Він особливо корисний для тих, хто прагне передбачити, чи може монета зрости або впасти, оскільки цей метод робить висновки на основі майбутніх прогнозів, а не минулих подій, представлених у вигляді цінних графіків. Оцінюючи внутрішню вартість, інвестори можуть визначити, чи є ціна переоціненою або недооціненою. Це можна зробити, проаналізувавши якісні фактори, такі як стан економіки, кон'юнктура ринку, менеджмент криптокомпанії або ринкова капіталізація [24].

Метод фундаментального аналізу має певні сильні та слабкі сторони при застосуванні до прогнозування криптовалют.

Сильні сторони:

- цей метод може надати ширшу картину вартості, враховуючи якомога більше факторів;
- він може допомогти інвесторам визначити довгострокові інвестиційні можливості, оцінивши внутрішню вартість монети;
- він може допомогти інвесторам визначити, чи є криптовалюта переоціненою або недооціненою.

Слабкі сторони:

- фундаментальний аналіз це глибокий, іноді тривалий процес, який включає в себе розумну частку інтуїції та думок інвестора;
- в ході використання методу може бути складно оцінити деякі показники, такі як дії користувачів, що відіграють вирішальну роль у вартості валюти;

- метод може не підходити для короткострокових торгових можливостей, оскільки він робить висновки на основі майбутніх прогнозів, а не минулих подій, представлених у вигляді цінових графіків.

Отже, метод фундаментального аналізу - це метод, який використовується інвесторами на фінансових ринках для оцінки та вивчення змінних, що впливають на вартість. Він корисний для прогнозування довгострокових тенденцій криптовалют і може допомогти інвесторам визначити, чи є монета переоціненою або недооціненою. Однак цей метод може не підходити для короткострокових торгових можливостей і може займати велику кількість часу.

2.2.2 Технічний аналіз та його використання у прогнозуванні крипто валютних курсів

Технічний аналіз - це метод, який використовується для прогнозування курсів криптовалют шляхом аналізу історичних цін та виявлення закономірностей і тенденцій у ринкових даних, таких як обсяг торгів і рух цін [25]. Технічний аналіз ґрунтується на припущенні, що історичні цінові дані можуть бути використані для прогнозування майбутнього руху цін, і зазвичай використовується трейдерами для прийняття короткострокових торгових рішень на ринку криптовалют.

Деякі з ключових понять, що використовуються в технічному аналізі для прогнозування, включають:

- Графіки: Технічний аналіз використовує графіки для візуалізації історичних цінових даних і виявлення закономірностей і тенденцій в даних.
- Індикатори: Технічні аналітики використовують індикатори, такі як ковзаючі середні, для виявлення тенденцій і прогнозування майбутнього руху цін.

- Рівні підтримки і опору: Технічні аналітики використовують рівні підтримки і опору, щоб визначити області, де ціна криптовалюти, ймовірно, зустрине опір або підтримку.

Однією з сильних сторін технічного аналізу є те, що його можна використовувати для визначення короткострокових торгових можливостей на ринку. Технічний аналіз також може бути використаний для виявлення тенденцій і закономірностей, які можуть бути не відразу помітні в інших формах аналізу [25].

Однак технічний аналіз також має деякі слабкі сторони при застосуванні до прогнозування криптовалют:

- Технічний аналіз базується на історичних даних і може не враховувати нові розробки або події, які можуть вплинути на ринок.
- Технічний аналіз може бути суб'єктивним, і різні аналітики можуть по-різному інтерпретувати одні й ті ж дані.
- Технічний аналіз може не підходити для довгострокових інвесторів, оскільки він в першу чергу зосереджений на короткострокових цінових рухах.

Отже, технічний аналіз - це метод, який використовується для прогнозування курсів криптовалют шляхом аналізу історичних цін і виявлення закономірностей і тенденцій в ринкових даних. Він може бути корисним інструментом для виявлення короткострокових торгових можливостей на ринку, але він також має деякі недоліки і може не підходити для довгострокових інвесторів.

2.2.3 Гібридні методи, які поєднують фундаментальний та технічний аналіз

Гібридні методи, які поєднують фундаментальний і технічний аналіз, стають все більш популярними на ринку криптовалют. Поєднуючи технічний і фундаментальний аналіз, трейдери можуть отримати більш повну картину криптовалютного ринку і приймати більш обґрунтовані торгові рішення [26].

Ось кілька прикладів того, як можна поєднувати фундаментальний і технічний аналіз:

- Використання технічного аналізу для виявлення короткострокових торгових можливостей і використання фундаментального аналізу для оцінки довгострокових тенденцій.
- Використання фундаментального аналізу для виявлення недооцінених монет і використання технічного аналізу для визначення точок входу і виходу з торгів.
- Використання технічного аналізу для виявлення тенденцій і закономірностей і використання фундаментального аналізу для підтвердження цих тенденцій і закономірностей на основі більш широких ринкових умов.

Однією з переваг використання гібридних методів є те, що вони можуть допомогти трейдерам визначити як короткострокові, так і довгострокові торгові можливості. Поєднання обох методів також може допомогти трейдерам зменшити вплив ринкового шуму і зробити більш точні прогнози щодо майбутнього руху цін [26].

Однак гібридні методи також мають деякі слабкі сторони, коли вони застосовуються для прогнозування криптовалют:

- Поєднання обох методів може бути трудомістким процесом, що вимагає від трейдерів аналізу значної кількості даних.
- Гібридні методи можуть бути складними, і трейдерам може знадобитися глибоке розуміння як технічного, так і фундаментального аналізу.
- Поєднання обох методів не завжди призводить до більш точних прогнозів, оскільки ринкові умови можуть змінюватися швидко і непередбачувано.

Підсумовуючи, можна сказати, що гібридні методи, які поєднують фундаментальний і технічний аналіз, стають все більш популярними на ринку. Поєднуючи обидва методи, трейдери можуть отримати більш повну картину криптовалютного ринку і приймати більш обґрунтовані торгові рішення. Однак

поєднання обох методів може бути складним і трудомістким, і трейдерам може знадобитися глибоке розуміння як технічного, так і фундаментального аналізу.

2.2.4 Методи машинного навчання та штучного інтелекту для прогнозування криптовалютних курсів

Останніми роками методи машинного навчання та штучного інтелекту (ШІ) набувають все більшої популярності для прогнозування курсів криптовалют. Ці методи використовують алгоритми для аналізу великих обсягів даних і виявлення закономірностей, які можуть бути неочевидними для аналітиків-людей [27]. Деякі приклади методів машинного навчання та штучного інтелекту для прогнозування курсів включають такі методи:

- Нейронні мережі: Це алгоритми, які можна навчити виявляти закономірності в даних і робити прогнози щодо майбутнього руху цін [27].
- Випадкові ліси: Алгоритми, які використовують дерева рішень для аналізу даних з ринку і прогнозування майбутнього руху цін [28].
- Машини опорних векторів: Алгоритми, які використовують статистичні методи для аналізу даних ринків та прогнозування майбутнього руху цін [28].

Дослідження показали, що машинне навчання і торгівля за допомогою штучного інтелекту можуть бути ефективними для прогнозування цін на криптовалюту і отримання аномальних прибутків [27]. Наприклад, дослідження показало, що торгові стратегії, засновані на методах машинного навчання, таких як лінійні моделі, випадкові ліси та машини опорних векторів, можуть бути прибутковими навіть за несприятливих ринкових умов [28].

Однією з переваг використання методів машинного навчання та штучного інтелекту для прогнозування курсів є те, що вони можуть швидко аналізувати великі

обсяги даних і виявляти закономірності, які можуть бути неочевидними для аналітиків-людей. Ці методи також можуть адаптуватися до мінливих ринкових умов і можуть використовуватися як для короткострокового, так і для довгострокового прогнозування [27].

Однак використання методів машинного навчання та штучного інтелекту для прогнозування курсів криптовалют також має певні недоліки:

- Ці методи можуть бути складними і вимагають глибокого розуміння як машинного навчання, так і даних ринку.
- Ці методи можуть підходити не всім інвесторам, оскільки вони можуть бути дорогими в реалізації і вимагати значних обчислювальних ресурсів.
- На точність цих методів можуть впливати зміни на ринку або несподівані події.

Отже, методи машинного навчання та штучного інтелекту стають все більш популярними для прогнозування курсів криптовалют. Ці методи можуть швидко аналізувати великі обсяги даних і виявляти закономірності, які можуть бути неочевидними для аналітиків-людей. Однак використання цих методів може бути складним і дорогим, а на їхню точність можуть впливати зміни на криптовалютному ринку або несподівані події.

2.3 Проблеми та виклики прогнозування курсу криптовалют

2.3.1 Висока волатильність та непередбачуваність ринку криптовалют

Ринок криптовалют відомий своєю високою волатильністю та непередбачуваністю [29]. Волатильність вимірює ступінь зміни ціни активу з плином часу і використовується фінансовими експертами для оцінки вартості активу за певний період часу порівняно з середнім показником або індексом [30]. У випадку з

криптовалютами, їхні ціни можуть різко змінюватися протягом декількох секунд під впливом різних факторів, в тому числі із таких причин:

- Новини та анонси: Новини, пов'язані з конкретною монетою або ринком в цілому можуть викликати різкі коливання цін.
- Ринкові настрої: Загальний настрій ринку може спонукати інвесторів купувати або продавати криптовалюту, що призводить до раптових змін у цінах.
- Регуляторні зміни: Урядові постанови або зміни в політиці, пов'язані з криптовалютами, можуть спричинити різкі коливання цін.
- Хакерські атаки і порушення безпеки: Порушення безпеки або хакерські атаки можуть спричинити втрату довіри інвесторів до монети, що призводить до раптового падіння ціни.

Висока волатильність ринку може ускладнити точне прогнозування майбутнього руху цін. Ця непередбачуваність може бути як викликом, так і можливістю для інвесторів. З одного боку, раптове падіння цін може призвести до значних втрат. З іншого боку, раптове зростання цін може призвести до значних прибутків [29].

Підсумовуючи, ринок криптовалют відомий своєю високою волатильністю та непередбачуваністю. Ціни на монети можуть різко змінюватися протягом декількох секунд через різноманітні фактори, включаючи новини та оголошення, ринкові настрої, регуляторні зміни та порушення безпеки. Висока волатильність ринку може ускладнити точне прогнозування майбутніх цінових змін, але вона також може надати інвесторам можливість отримати значні прибутки.

2.3.2 Вплив новин та інформації на курс криптовалют

Новини та інформація відіграють значну роль на ринку криптовалют і можуть мати суттєвий вплив на їх курси [31]. Новини та інформація можуть надходити з різних джерел, включаючи такі, як:

- Урядові нормативні акти: Зміни в державній політиці або нові правила, пов'язані з криптовалютами, можуть мати значний вплив на їх курси.
- Ринкові настрої: Загальний настрій ринку може спонукати інвесторів купувати або продавати монети, що призводить до раптових змін цін.
- Зломи і порушення безпеки: Порушення безпеки або хакерські атаки можуть змусити інвесторів втратити довіру до монети, що призведе до раптового падіння ціни.
- Новини про використання: Новини про те, що великі фінансові установи починають приймати певні криптовалюти, можуть дати імпульс до зростання цін.
- Скорочення видачі майнерам: Цикл халвінгу біткоіна, який зменшує кількість нових BTC, що випускаються в обіг кожні чотири роки, також може впливати на курси.

Вплив новин та інформації на курси криптовалют може бути як позитивним, так і негативним. Позитивні новини можуть призвести до збільшення попиту і зростання цін, а негативні - до зменшення попиту і зниження цін. Наприклад, оголошення про нову криптовалютну біржу, яка підтримує певну монету, може збільшити попит і призвести до зростання цін, в той час як новина про порушення безпеки може викликати у інвесторів втрату довіри і призвести до зниження цін [31].

В цілому, новини та інформація відіграють важливу роль на ринку криптовалют і можуть мати значний вплив на курси монет. Новини та інформація можуть надходити з різних джерел, включаючи урядові постанови, ринкові настрої, хакерські атаки та порушення безпеки, новини про усиновлення та цикл халвінгу. Вплив новин та інформації на курси криптовалют може бути як позитивним, так і

негативним, що призводить до збільшення або зменшення попиту і підвищення або зниження цін.

2.3.3 Обмежена історична інформація та недостатність даних для прогнозування

Однією з проблем прогнозування цін на криптовалюти є обмеженість історичної інформації та недостатня кількість даних, доступних для аналізу [32]. Криптовалюти є відносно новим класом активів, і порівняно з традиційними активами, такими як акції та облігації, існує менше історичних даних, доступних для аналізу. Це ускладнює розробку точних моделей прогнозування, які можуть враховувати тенденції та закономірності в даних.

Крім того, криптовалютний ринок є дуже волатильним і схильним до раптових цінових коливань через різноманітні фактори, включаючи новини та оголошення, ринкові настрої, регуляторні зміни та порушення безпеки. Ці раптові зміни цін можуть ще більше ускладнити розробку точних моделей прогнозування, які можуть передбачити майбутні зміни цін.

Незважаючи на ці виклики, все ще існують способи прогнозування цін на монети, використовуючи історичні дані та інші методи. Один з підходів полягає у використанні машинного навчання та алгоритмів штучного інтелекту для аналізу великих обсягів даних і виявлення закономірностей, які можуть бути неочевидними для аналітиків-людей [33]. Інший підхід полягає у використанні технічного аналізу, який передбачає аналіз минулих ринкових даних для виявлення тенденцій та прогнозування майбутнього руху цін [34].

Таким чином, обмеженість історичної інформації та недостатня кількість даних, доступних для аналізу, є значними проблемами, коли йдеться про прогнозування цін. Однак все ще існують способи розробки точних моделей прогнозування, в тому числі з використанням алгоритмів машинного навчання і штучного інтелекту, а також технічного аналізу. Хоча ці методи не є надійними, вони

можуть допомогти інвесторам приймати більш обґрунтовані рішення щодо купівлі, продажу та зберігання криптовалют.

2.3.4 Проблеми зі збором, чисткою та обробкою даних для прогнозування криптовалютних курсів

Збір, очищення та обробка даних для прогнозування курсів криптовалют може бути складним завданням [35]. Нижче наведено деякі з проблем, які можуть виникнути при зборі, очищенні та обробці даних:

- Обмежена доступність даних. Криптовалюти є відносно новим класом активів, і для аналізу може бути обмежена кількість історичних даних. Це може ускладнити розробку точних моделей прогнозування, які можуть враховувати тенденції та закономірності в даних.
- Якість даних. Якість наявних даних також може бути значною проблемою. Наприклад, даних про ICO може бути мало, і може бути важко зібрати певні дані в цифровому вираженні, наприклад, виміряти якість команди розробників, які працювали над ICO.
- Неузгодженість даних. Дані, зібрані з різних джерел, можуть бути неузгодженими, що ускладнює їх очищення та підготовку до аналізу. Наприклад, на веб-сайтах можуть бути різні дати початку і закінчення ICO або країни, в якій воно було запущено.
- Пропущені дані. Дані можуть бути пропущеними для деяких змінних, і звичайний підхід до обробки таких даних часто полягає у використанні середнього значення або медіани. Однак, через розрідженість такий підхід може бути неефективним.

Незважаючи на ці проблеми, все ж таки існують способи збирати, очищати та обробляти дані для прогнозування курсів криптовалют. Один з підходів полягає у використанні машинного навчання та алгоритмів штучного інтелекту для аналізу

великих обсягів даних і виявлення закономірностей, які можуть бути неочевидними для аналітиків-людей [35]. Інший підхід полягає у використанні технічного аналізу, який передбачає аналіз минулих ринкових даних для виявлення тенденцій і прогнозування майбутнього руху цін [36].

Отже, збір, очищення та обробка даних для прогнозування курсів монет може бути складним завданням через обмежену доступність даних, суперечливість даних, відсутність даних та якість наявних даних. Однак, все ще існують способи розробки точних моделей прогнозування, в тому числі за допомогою машинного навчання, алгоритмів штучного інтелекту і технічного аналізу. Хоча ці методи не є надійними, вони можуть допомогти інвесторам приймати більш обґрунтовані рішення щодо купівлі, продажу та зберігання криптовалют.

2.4 Особливості прогнозування криптовалютних курсів

2.4.1 Вплив масштабних змін та глобальних подій на курс криптовалют

Масштабні зміни та глобальні події можуть мати значний вплив на ринки [37]. Існує багато різних подій, що можуть спричинити скачки в курсах криптовалют і ускладнити прогнозування для моделей.

Зміни в державній політиці або нові правила, пов'язані з криптовалютами, можуть мати значний вплив. Наприклад, у 2017 році Китай заборонив первинне розміщення монет (ICO), що призвело до різкого падіння цін [38].

Глобальні економічні умови також можуть впливати на ринки. Наприклад, під час пандемії COVID-19 ціни на монети спочатку впали через ринкову невизначеність, але згодом відновилися, оскільки інвестори шукали альтернативні інвестиції [39].

Геополітичні події, такі як політична нестабільність або війна, також можуть мати вплив на курс. Наприклад, під час громадянської війни в Сирії деякі сирійці

звернулися до електронних валют як до способу захистити своє багатство від інфляції та нестабільності [38].

Технологічний прогрес у криптовалютному просторі також може чинити вплив на ринок. Наприклад, розробка нових технологій блокчейну або вдосконалення існуючих момент може призвести до збільшення попиту і зростання цін [38].

Важливо відзначити, що вплив масштабних змін і глобальних подій на курси може бути як позитивним, так і негативним. Позитивні новини можуть призвести до збільшення попиту і зростання цін, а негативні - до зменшення попиту і зниження цін.

Отже, масштабні зміни та глобальні події можуть суттєво впливати на курси. Державне регулювання, економічні умови, геополітичні події та технологічний прогрес - ось деякі приклади факторів, які можуть впливати на курси криптовалют. Як і у випадку з будь-якою інвестицією, важливо бути в курсі цих факторів та їхнього потенційного впливу на ціни.

2.4.2 Роль алгоритмічної торгівлі та торгових ботів у прогнозуванні криптовалют

Алгоритмічна торгівля та торгові боти можуть відігравати значну роль у прогнозуванні криптовалют в майбутньому [40]. Торгові боти - це комп'ютерні програми, які створюють і подають заявки на купівлю та продаж на біржі на основі правил заздалегідь визначеної торгової стратегії. Алгоритмічна торгівля, з іншого боку, передбачає використання комп'ютерних алгоритмів для аналізу великих обсягів даних і прийняття торгових рішень на основі цього аналізу [41].

Однією з головних переваг алгоритмічної торгівлі та торгових ботів є те, що вони можуть усунути емоції з торгівлі. Криптовалютні трейдери можуть використовувати автоматизацію на основі даних для збільшення прибутку, мінімізуючи ризик і обмежуючи втрати [40]. Торгові боти можуть працювати 24/7, а

це означає, що вони можуть швидко реагувати на зміни на ринку і здійснювати угоди на основі даних і тенденцій, а не емоційного імпульсу [42].

Алгоритмічна торгівля та торгові боти також можуть допомогти інвесторам виявити закономірності та тенденції на ринку, які можуть бути неочевидними для людей-аналітиків [41]. Аналізуючи великі обсяги даних, ці інструменти можуть виявляти потенційні торгові можливості та приймати обґрунтовані торгові рішення на основі цього аналізу.

Важливо зазначити, що алгоритмічна торгівля та торгові боти не є безвідмовними і не гарантують отримання прибутку. Ефективність цих інструментів значною мірою залежить від ряду факторів, включаючи платформу і ботів, які обираємо, а також наш рівень знань і досвіду [42]. Важливо провести ретельне дослідження і тестування перед використанням будь-якого торгового бота, щоб переконатися в його надійності та ефективності.

Підсумовуючи, алгоритмічна торгівля і торгові боти можуть відігравати значну роль у прогнозуванні криптовалют. Ці інструменти можуть позбавити торгівлю емоцій, виявити закономірності та тенденції на ринку і швидко реагувати на ринкові зміни. Однак перед використанням будь-якого торгового бота важливо провести ретельне дослідження і тестування, щоб переконатися в його надійності та ефективності.

2.4.3 Аналіз соціальних медіа та громадської думки для прогнозування

Аналіз соціальних мереж та громадської думки може бути корисним інструментом для прогнозування курсів криптовалют [43]. Платформи соціальних мереж, такі як Twitter, Reddit і Telegram, можуть надати цінну інформацію про ставлення громадськості до теми криптографії та валют і допомогти визначити ринкові тенденції [2]. Аналізуючи дані соціальних мереж, інвестори можуть краще

зрозуміти, як люди ставляться до монет, що може допомогти їм у прийнятті торгових рішень.

Одним з підходів до аналізу даних соціальних мереж є аналіз настроїв. Він передбачає використання методів обробки природної мови для виявлення позитивних, негативних і нейтральних настроїв у публікаціях в соціальних мережах, пов'язаних з криптовалютами [43]. Аналіз настроїв може допомогти інвесторам виявити тенденції та закономірності в громадській думці, які можуть бути неочевидними при традиційному аналізі ринку.

Іншим підходом до аналізу даних соціальних мереж є аналіз соціальних мереж. Він передбачає вивчення взаємовідносин між користувачами на платформах соціальних мереж і того, як вони взаємодіють один з одним [43]. Аналіз соціальних мереж може допомогти інвесторам визначити впливових осіб у криптовалютному просторі та зрозуміти, як інформація проходить через соціальні мережі.

Важливо зазначити, що аналіз соціальних мереж не є надійним і не гарантує точних прогнозів. На настрої в соціальних мережах може впливати цілий ряд факторів, включаючи фейкові новини, маніпуляції та ринковий шум. Інвестори повинні використовувати аналіз соціальних мереж як один з багатьох інструментів при прийнятті торгових рішень і не повинні покладатися виключно на дані соціальних мереж при прийнятті інвестиційних рішень.

Отже, аналіз соціальних мереж і громадської думки може бути корисним інструментом для прогнозування курсів криптовалют. Аналіз настроїв і аналіз соціальних мереж може допомогти інвесторам виявити тенденції і закономірності в громадській думці і зрозуміти, як інформація проходить через соціальні мережі. Однак аналіз соціальних мереж не є безвідмовним і повинен використовуватися як один з багатьох інструментів при прийнятті торгових рішень.

2.5 Висновки до розділу

Прогнозування курсу криптовалюти є складним завданням через високу волатильність ринку та непередбачуваність факторів, що впливають на ціни. Аналітики передбачають зростання ціни біткойна до 500 000 доларів за монету в 2025 році і до 1 мільйона доларів за монету в 2030 році. Проте ці прогнози повинні розглядатися з обережністю, оскільки ринок криптовалют є дуже волатильним.

У даному розділі було проведено кілька досліджень щодо прогнозування криптовалют та прибутковості торгових стратегій на основі методів машинного навчання. Дослідники використовували різні атрибути торгової та мережевої активності, але точність прогнозування окремих моделей варіюється. Зазначається, що Ensemble 5 була найкращою торговою стратегією для ethereum і litecoin, досягнувши високого річного коефіцієнта Шарпа та прибутку. ARIMA вважається найкращою моделлю для прогнозування ціни біткоїна.

Фундаментальний аналіз може бути корисним для прогнозування довгострокових тенденцій криптовалют, оцінки вартості та визначення переоціненості або недооціненості активу. Технічний аналіз, зі свого боку, може бути корисним для виявлення короткострокових торгових можливостей на ринку. Гібридні підходи, які поєднують обидва методи, стають все більш популярними серед трейдерів, оскільки вони дозволяють отримати більш повну картину ринку та приймати обґрунтовані рішення.

Методи машинного навчання та штучного інтелекту набувають все більшої популярності для прогнозування курсів криптовалют. Вони можуть швидко аналізувати великі обсяги даних та виявляти закономірності, які можуть бути неочевидними для аналітиків. Проте, використання цих методів може бути складним і вимагати додаткових зусиль і ресурсів. При цьому, їхня точність може бути під впливом змін на ринку або несподіваних подій.

Ринок криптовалют відомий своєю високою волатильністю та непередбачуваністю. Ціни на монети можуть різко змінюватися протягом декількох

секунд через різноманітні фактори, такі як новини, регуляторні зміни та безпекові порушення. Висока волатильність ринку криптовалют ускладнює точне прогнозування майбутніх цінових змін, але також надає можливість для отримання значних прибутків.

Усі ці висновки вказують на складність прогнозування курсу криптовалют та необхідність поєднання різних методів та підходів для отримання більш обґрунтованих торгових рішень. Крім того, важливо бути обізнаним із ризиками, пов'язаними з волатильністю ринку криптовалют та непередбачуваністю факторів, які впливають на ціни.

РОЗДІЛ 3 ОБРОБКА ТА АНАЛІЗ ДАНИХ ПРО КУРС КРИПТОВАЛЮТ

У розділі 3 нашого дослідження ми зосередимось на обробці та аналізі даних про курс криптовалют. Криптовалюти стали все більш популярними, і їх курси можуть бути дуже змінними. Це створює великий інтерес для інвесторів та трейдерів, які бажають прогнозувати майбутні зміни вартості монет для прийняття розумних інвестиційних рішень.

У цьому розділі ми вивчимо різні методи обробки та аналізу даних про курс монет. Ми розпочнемо з завантаження даних про курс із зовнішніх джерел та їх передбачення з використанням моделі LSTM, ARIMA, Prophet, Exp Smoothing. Кожна із моделей вимагає унікального підходу для обробки даних, на яких буде навчатись, тому в ході роботи доведеться створити декілька різних варіантів.

3.1 Огляд обраного набору даних

Датасет, що був обраний, містить історичну інформацію про ціни криптовалют. Кожна криптовалюта представлена окремим CSV-файлом, і дані про ціни доступні щоденно з 28 квітня 2013 року. Даний датасет містить історичну інформацію про деякі з найбільш впливових криптовалют за розміром ринкової капіталізації.

Структура кожного файлу включає наступні колонки (рис. 3.1):

1. Date - дата спостереження
2. Open - ціна відкриття на заданий день
3. High - найвища ціна на заданий день
4. Low - найнижча ціна на заданий день
5. Close ціна закриття на заданий день
6. Volume - обсяг транзакцій на заданий день

7. Marketcap - ринкова капіталізація у доларах США

SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap	
0	1	Chainlink	LINK	2017-09-21 23:59:59	0.207892	0.155292	0.189132	0.169680	2.126270e+06	5.938800e+07
1	2	Chainlink	LINK	2017-09-22 23:59:59	0.181675	0.139869	0.169826	0.150503	1.423220e+06	5.267605e+07
2	3	Chainlink	LINK	2017-09-23 23:59:59	0.157440	0.126297	0.149562	0.131129	1.186670e+06	4.589515e+07
3	4	Chainlink	LINK	2017-09-24 23:59:59	0.190250	0.130275	0.131110	0.170278	1.025740e+06	5.959730e+07
4	5	Chainlink	LINK	2017-09-25 23:59:59	0.182637	0.158580	0.170636	0.182198	1.021370e+06	6.376930e+07
...
1380	1381	Chainlink	LINK	2021-07-02 23:59:59	18.520256	17.305501	18.220669	18.283561	6.865419e+08	7.990091e+09
1381	1382	Chainlink	LINK	2021-07-03 23:59:59	18.886003	17.898075	18.285458	18.538818	5.090763e+08	8.101641e+09
1382	1383	Chainlink	LINK	2021-07-04 23:59:59	19.894503	18.028957	18.548805	19.262838	6.802681e+08	8.418044e+09
1383	1384	Chainlink	LINK	2021-07-05 23:59:59	19.246891	17.814752	19.246891	18.371589	6.986191e+08	8.028560e+09
1384	1385	Chainlink	LINK	2021-07-06 23:59:59	20.240997	18.351877	18.351877	20.080466	1.156986e+09	8.775356e+09

1385 rows × 10 columns

Рисунок 3.1 - Зображення однієї із таблиць з даними

Ці дані дозволяють досліджувати зміни історичних цін та ринкових капіталізацій різних криптовалют з плином часу. Датасет також надає можливість прогнозування майбутніх цін криптовалют, аналізування волатильності, вивчення взаємозв'язків між змінами цін та виявлення сезонних тенденцій.

Дані в цьому датасеті були зібрані з coinmarketcap, а використання цих даних є вільним та безкоштовним.

Цей датасет включає історичні дані про такі криптовалюти:

1. Aave (AAVE)
2. Binance Coin (BNB)
3. Bitcoin (BTC)
4. Cardano (ADA)
5. Chainlink (LINK)
6. Cosmos (ATOM)
7. Crypto.com Coin (CRO)
8. Dogecoin (DOGE)
9. EOS (EOS)

10. Ethereum (ETH)
11. IOTA (MIOTA)
12. Litecoin (LTC)
13. Monero (XMR)
14. NEM (XEM)
15. Polkadot (DOT)
16. Solana (SOL)
17. Stellar (XLM)
18. Tether (USDT)
19. TRON (TRX)
20. USD Coin (USDC)
21. Uniswap (UNI)
22. Wrapped Bitcoin (WBTC)
23. XRP (XRP)

Дані про криптовалюти мають різну довжину, оскільки представлені дані відображають щоденну історію цін на монети. Кожен рядок у файлі відповідає конкретному дню, починаючи з певної дати і закінчуючи останнім доступним днем. Ці дані дозволяють досліджувати зміни цін та капіталізацію різних криптовалют протягом певного періоду часу і виконувати аналіз цих змін. (Рис. 3.2-3)

SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap	
0	1	Solana	SOL	2020-04-11 23:59:59	1.049073	0.765020	0.951054	0.776819	4.386244e+07	0.000000e+00
1	2	Solana	SOL	2020-04-12 23:59:59	0.956670	0.762426	0.785448	0.882507	3.873690e+07	0.000000e+00
2	3	Solana	SOL	2020-04-13 23:59:59	0.891603	0.773976	0.890760	0.777832	1.821129e+07	0.000000e+00
3	4	Solana	SOL	2020-04-14 23:59:59	0.796472	0.628169	0.777832	0.661925	1.674761e+07	0.000000e+00
4	5	Solana	SOL	2020-04-15 23:59:59	0.704964	0.621531	0.669289	0.646651	1.307528e+07	0.000000e+00
...
447	448	Solana	SOL	2021-07-02 23:59:59	34.031786	31.479924	33.306310	34.020482	4.402988e+08	9.275257e+09
448	449	Solana	SOL	2021-07-03 23:59:59	35.404770	33.298475	34.015575	34.478816	3.270200e+08	9.400216e+09
449	450	Solana	SOL	2021-07-04 23:59:59	35.502372	33.555737	34.495117	34.310601	3.034205e+08	9.354354e+09
450	451	Solana	SOL	2021-07-05 23:59:59	34.461824	32.482692	34.282550	32.984588	3.138393e+08	8.992833e+09
451	452	Solana	SOL	2021-07-06 23:59:59	34.978319	32.930307	32.930307	34.269140	3.653360e+08	9.343050e+09

452 rows × 10 columns

Рисунок 3.2 - дані про монету, що включають 452 дні

SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap	
0	1	Bitcoin	BTC	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.000000e+00	1.603769e+09
1	2	Bitcoin	BTC	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.000000e+00	1.542813e+09
2	3	Bitcoin	BTC	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.000000e+00	1.298955e+09
3	4	Bitcoin	BTC	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.000000e+00	1.168517e+09
4	5	Bitcoin	BTC	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.000000e+00	1.085995e+09
...
2986	2987	Bitcoin	BTC	2021-07-02 23:59:59	33939.588699	32770.680780	33549.600177	33897.048590	3.872897e+10	6.354508e+11
2987	2988	Bitcoin	BTC	2021-07-03 23:59:59	34909.259899	33402.696536	33854.421362	34668.548402	2.438396e+10	6.499397e+11
2988	2989	Bitcoin	BTC	2021-07-04 23:59:59	35937.567147	34396.477458	34665.564866	35287.779766	2.492431e+10	6.615748e+11
2989	2990	Bitcoin	BTC	2021-07-05 23:59:59	35284.344430	33213.661034	35284.344430	33746.002456	2.672155e+10	6.326962e+11
2990	2991	Bitcoin	BTC	2021-07-06 23:59:59	35038.536363	33599.916169	33723.509655	34235.193451	2.650126e+10	6.418992e+11

2991 rows x 10 columns

Рисунок 3.3 - Дані про монету, що включають 2991 день

Різна довжина даних про криптовалюти може вплинути на процес передбачення. Оскільки кожна монета має свою власну унікальну історію цін, а враховувати довжину даних є доволі важливим аспектом при створенні моделей передбачення.

Наявність більш тривалих даних дозволяє моделі отримати більше інформації про зміни цін у минулому і може допомогти виявити певні закономірності або тренди, що впливають на ціни криптовалют. З іншого боку, короткі дані можуть обмежити розмаїтість закономірностей, які модель може виявити, і можуть призвести до менш точних передбачень.

Крім того, різна довжина даних може вплинути на вибір алгоритму передбачення. Деякі алгоритми можуть бути більш ефективними на короткотермінових даних, тоді як інші можуть показати кращі результати на довготермінових даних. Важливо адаптувати модель передбачення до конкретної довжини даних, щоб забезпечити якісні результати.

Загалом, різна довжина даних про криптовалюти впливає на розуміння та передбачення цін. Це вимагає обережного вибору моделей, алгоритмів і методів обробки даних для досягнення точних та надійних передбачень.

3.2 Аналіз доступних даних

Перед тим, як перейти до розробки моделей передбачення цін криптовалют, необхідно ретельно проаналізувати доступні дані. Аналіз даних дозволяє отримати інсайти щодо їх структури, характеристик і особливостей, що є важливими для правильного побудови моделей та отримання точних передбачень.

У цьому пункті ми проведемо аналіз доступних даних про криптовалюту. Дані включають історичні ціни, відкриті, максимальні та мінімальні ціни, обсяги торгів, капіталізацію ринку та інші показники.

Для початку слід вивести графіки всіх валют і подивитись на спільні ознаки, які можуть бути виявлені у даних. (Рис 3.4-3.8)

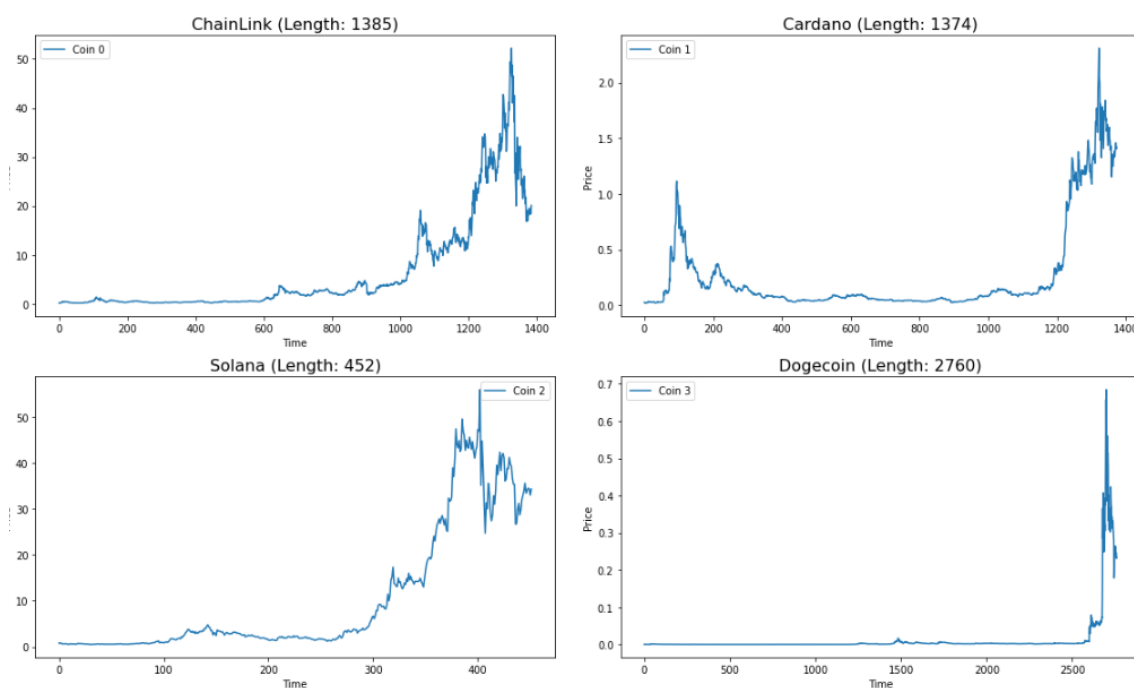


Рисунок 3.4 - Ціни на криптовалюти

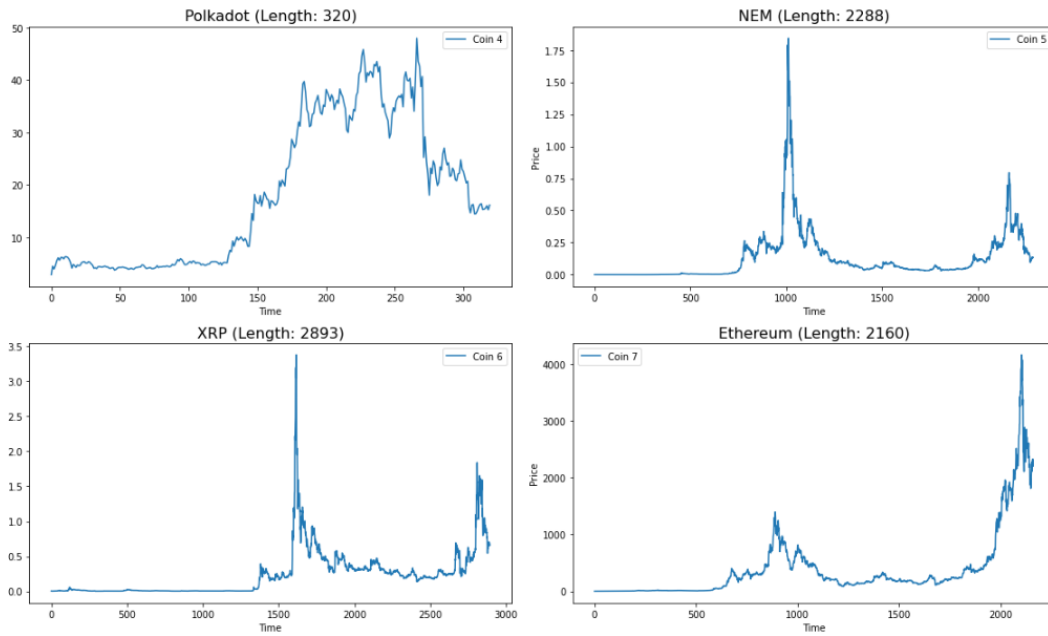


Рисунок 3.5 - Ціни на криптовалюти

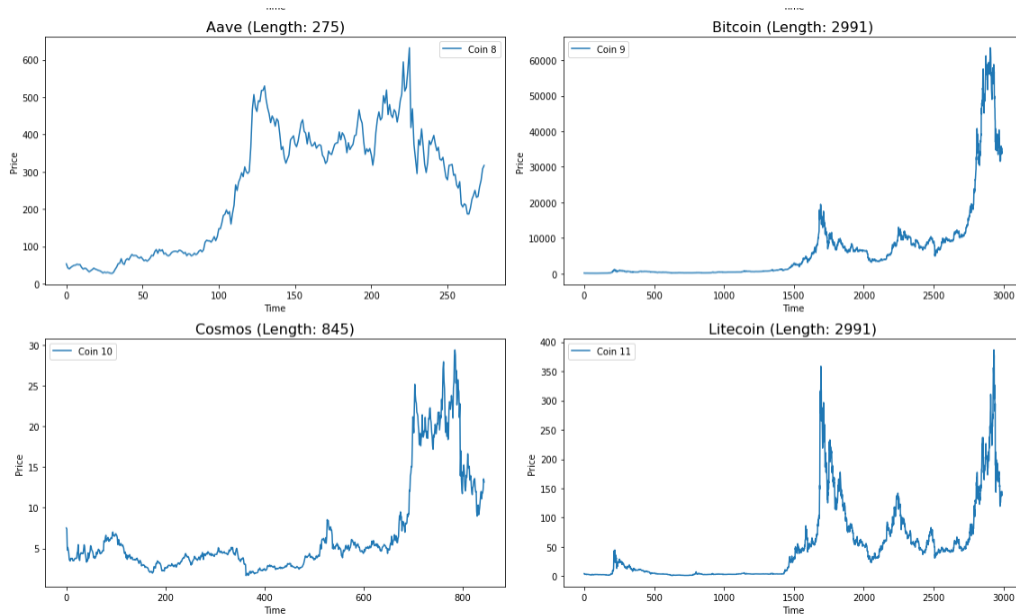


Рисунок 3.6 - Ціни на криптовалюти

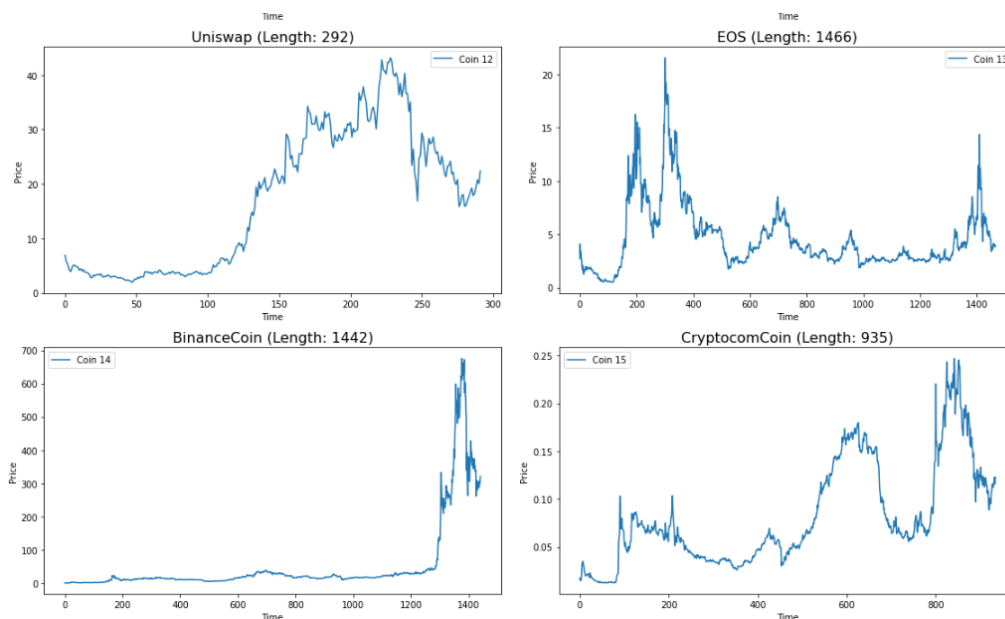


Рисунок 3.7 - Ціни на криптовалюти

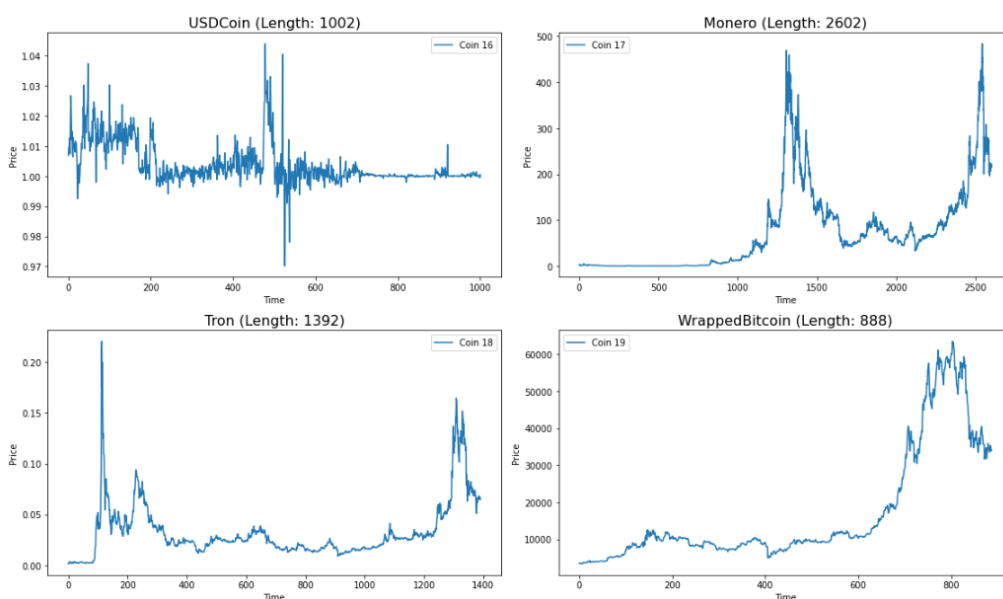


Рисунок 3.8 - Ціни на криптовалюти

Як видно із графіків вище, монети із довгою історією були доволі дешевими в порівнянні із теперішніми цінами, також такі моделі пережили декілька сезонних коливань, що може бути корисними даними для моделей.

Також на графіках присутні стейбл-коїни, що є криптовалютними репрезентаціями валют в реальному світі. Прикладом такої валюти є USD coin. Графік

цієї монети залежить від курсу долара, а отже, до цієї монети звичайні правила навряд будуть працювати.

3.3 Обробка даних для LSTM

Для LSTM нам буде необхідно створити проміжки даних про ціни криптовалют по 60 днів, в яких 60 день буде цільовим для моделі.

Для цього, в першу чергу, ми проведемо масштабування даних про криптовалюту. Для цієї задачі буде використано метод Min Max Scaling. Метод Min Max Scaling (мінімаксне масштабування) є одним із методів масштабування даних, який використовується для нормалізації значень в певному діапазоні. Цей метод дозволяє перетворити значення на новий діапазон, зазвичай від 0 до 1. Процес Min Max Scaling полягає у відніманні мінімального значення від кожного значення та поділі отриманої різниці на розмах (різницю між максимальним і мінімальним значеннями). Це дозволяє "стиснути" діапазон значень до вказаного інтервалу. Основна перевага методу MinMaxScaling полягає в тому, що він зберігає відношення між значеннями, а також дозволяє легше порівнювати та інтерпретувати дані. Формула мінімаксного масштабування наведена нижче:

$$x' = a + \frac{(x - \min(x)) \times (b - a)}{\max(x) - \min(x)},$$

де x - значення вхідної змінної,

x' - нормалізоване значення вхідної змінної,

$\min(x)$ - мінімальне значення вхідної змінної,

$\max(x)$ - максимальне значення вхідної змінної,

a, b - новий діапазон, у який відбувається масштабування.

Далі буде проведено розділення даних на проміжки. Для цього буде створено набір даних для кожної монети, що буде порізаний на проміжки по 60 днів із кроком в 1 день. Це дозволить моделі брати до уваги попередні ціни на графіку та модель матиме дані на вхід із однаковою розмірністю.

Отже, загальна ідея полягає в тому, що дані про криптовалюту масштабуються для нормалізації їх значень, а потім створюються вхідні-вихідні послідовності для тренування, валідації та тестування моделі із однаковими розмірами. Це дозволить моделі працювати на будь-яких даних та робити швидкі передбачення на будь-якому проміжку даних.

3.4 Обробка даних для ARIMA, Prophet та ExpSmoothing

Для даних моделей дані слід обробити іншим способом. Ці моделі відрізняються від моделі LSTM можливістю сприймати дані на всіх із будь-якою розмірністю. Ці моделі аналогічно першій моделі будуть проходити тестування на 60 останніх днях ціни криптовалюти.

Тренування всіх моделей буде відбуватись для кожної монети індивідуально, виходячи тільки із її графіка. Тому дані на обробку не будуть сумуватись та складатись в послідовності.

На вхід моделі будуть отримувати дані за весь час перед конкретним днем, тому нам доведеться створити 60 наборів даних із конкретною моделлю та навчити кожну з моделей на цьому наборі, щоб отримати найбільш точні передбачення на наступний день.

Всі моделі мають змогу робити передбачення на декілька днів уперед, але для більшої точності слід використовувати передбачення на один крок уперед.

Для точності оцінки роботи та порівняння всіх моделей всі дані будуть проходити той самий процес MinMaxScaling'a.

Отже, загальна ціль полягає в розбитті даних на 60 частин із кроком в плюс один день для кожної частини, щоб утворити набір для тренування та порівняння роботи моделей.

3.4 Висновки до розділу

В розділі нашого дослідження ми зосередились на обробці та аналізі даних про курс криптовалют. Криптовалюти стають все більш популярними, і їх курси можуть бути дуже змінними, що створює великий інтерес для інвесторів та трейдерів, які бажають прогнозувати майбутні зміни вартості криптовалют для прийняття розумних інвестиційних рішень.

У цьому розділі ми досліджували різні методи обробки та аналізу даних про курс криптовалют. Ми будемо використовувати моделі LSTM, ARIMA, Prophet та Exponential Smoothing для передбачення цін на криптовалюту. Кожна з цих моделей має свої особливості та вимоги до обробки даних.

Для моделі LSTM ми створили проміжки даних про ціни криптовалют по 60 днів, де 60-й день є цільовим значенням для моделі. Для цього ми застосували метод Min Max Scaling для нормалізації значень криптовалют. Масштабування даних дозволило зведення значень до встановленого діапазону та збереження відносних відношень між ними. Після цього дані були розділені на проміжки довжиною 60 днів для тренування та порівняння моделей.

Для ARIMA, Prophet та Exponential Smoothing дані не потребували масштабування, але воно було використано для зручності порівняння похибки та точності. Моделі будуть навчені на утворених проміжках від всієї довжини числового ряду - 60 днів, до всієї довжини числового ряду мінус один день, на цінах криптовалют.

В цілому, розділ надав нам можливість дослідити різні методи обробки та аналізу даних про курс криптовалют. Ми з'ясували, що кожна модель має свої

переваги та обмеження, і вибір моделі залежить від конкретного контексту та потреб дослідження. Також ми виявили, що, хоча більшість криптовалют і мають проміжки зі значними підвищеннями цін на них, існують різні типи монет, такі як USD coin, що не можна назвати типічними для інших випадків. Для таких даних слід використовувати моделі натреновані на схожих графіках, або натреновані спеціально саме на графіку поточної монети.

РОЗДІЛ 4 РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

4.1 Вибір методів прогнозування

4.1.1 Метод Prophet

Prophet - це бібліотека прогнозування часових рядів, розроблена Facebook, яка призначена для автоматичного прогнозування одновимірних даних часових рядів. Вона працює на основі адитивної моделі, де нелінійні тренди підганяються під річну, тижневу та денну сезонність, а також під вплив святкових днів. Бібліотека є стійкою до пропущених даних та зсувів у тренді, і зазвичай добре справляється з викидами [44].

PyCaret - це бібліотека машинного навчання з низьким рівнем коду, яка надає простий у використанні інтерфейс для проведення наскрізних експериментів з машинного навчання. PyCaret інтегрується з декількома популярними бібліотеками машинного навчання з відкритим вихідним кодом, включаючи Prophet, щоб забезпечити спрощений робочий процес для завдань машинного навчання [45].

Prophet розроблено так, щоб бути простим і повністю автоматичним, тобто можемо навести його на часовий ряд і отримати прогноз. Бібліотека призначена для внутрішнього використання в компанії, наприклад, для прогнозування продажів, виробничих потужностей тощо [46].

Етапи проведення прогнозного аналізу за допомогою бібліотеки Prophet є наступними:

- Попередня обробка даних: Візьміть лише два стовпчики - стовпчик дати та стовпчик цілі - і проігноруйте інші. Перетворіть стовпець дати у формат Date Time, а потім перейменуйте обидва стовпці на "ds" для дати і "y" для цілі. Можна використовувати масштабування функцій,

наприклад, нормалізацію або стандартизацію, для швидкого виконання коду і кращого прогнозування.

- Підгонка/навчання всієї моделі під бібліотеку Prophet.
- Створення нових даних за допомогою Prophet, а потім прогнозування результатів на основі цих нових даних.

Приклад коду, що демонструє підготовку даних для методу Prophet наведено на рисунку 4.1:

```
import pandas as pd
from fbprophet import Prophet

# Завантаження даних з таблиці у об'єкт DataFrame
df = pd.read_csv('шлях_до_файлу.csv')

# Перевірка та зміна формату дати
df['Дата'] = pd.to_datetime(df['Дата'])

# Вибір необхідних стовпців
df = df[['Дата', 'Значення']]

# Перейменування стовпців на "ds" (для дати) та "y" (для значення)
df = df.rename(columns={'Дата': 'ds', 'Значення': 'y'})

# Опціонально: обрізання аномалій або видалення пропущених значень
df = df.loc[df['y'].notnull()]

# Ініціалізація моделі Prophet та її навчання на даних
model = Prophet()
model.fit(df)

# Генерація майбутніх дат для прогнозу
future_dates = model.make_future_dataframe(periods=365)

# Виклик методу прогнозу для отримання результатів
forecast = model.predict(future_dates)
```

Рисунок 4.1 - Приклад коду для підготовки даних до методу Prophet

Prophet реалізує так звану адитивну модель прогнозування часових рядів, яка підтримує тренди, сезонність та свята. Бібліотека розроблена так, щоб бути простою і повністю автоматизованою, тобто вона призначена для внутрішнього використання в компанії, наприклад, для прогнозування продажів, виробничих потужностей тощо.

Prophet стійкий до пропущених даних і змін у тренді, і, як правило, добре справляється з викидами [47].

Декомпозиція в моделі відбувається за такою формулою:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t,$$

де $g(t)$ - функція тренду, яка моделює неперіодичні зміни у значенні часового ряду,

$s(t)$ - періодичні зміни (наприклад, тижнева та річна сезонність),

а $h(t)$ - ефекти святкових днів, які відбуваються за потенційно нерегулярним графіком протягом одного або більше днів. Похибка ϵ_t представляє будь-які специфічні зміни, які не враховуються моделлю.

Модель тренду обраховується за такою формулою:

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma),$$

де, k - темп зростання, δ це поправки на темп, m - заданий параметр o , а j задано на рівні для того, щоб зробити функцію неперервною. І γ_i дорівнює $s_i \delta_i$, щоб зробити функцію неперервною.

Модель покладається на ряди Фур'є, щоб забезпечити гнучку модель періодичних ефектів. Нехай регулярний період, який ми очікуємо від часового ряду (наприклад, $P=365:25$ для річних даних або $P=7$ для тижневих даних, коли ми масштабуємо нашу змінну в часі в днях). Ми можемо апроксимувати довільні гладкі сезонні ефекти за допомогою

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right),$$

Для врахування сезонності потрібно оцінити $2N$ параметрів $\beta = [a_1, b_1, \dots, a_N, b_N]^T$. Це робиться шляхом побудови матриці векторів сезонності для

кожного значення t у наших історичних та майбутніх даних, наприклад, з річною сезонністю та $N=10$,

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), \dots, \sin\left(\frac{2\pi(10)t}{365.25}\right) \right],$$

Тоді сезонна складова становить:

$$s(t) = X(t)\beta$$

Включення списку свят до моделі спрощується припущенням, що всі свята є незалежними. Для кожного свята нехай D_i - множина минулих та майбутніх дат для цього свята. Ми додаємо індикаторну функцію, яка показує, чи припадає час t на свято у i , і присвоюємо кожному святу параметр, який є відповідною зміною в прогнозі. Це робиться так само, як і сезонність, шляхом створення матриці регресорів

$$Z(t) = [1(t \in D_1), \dots, 1(t \in D_L)],$$

з чого випливає:

$$h(t) = Z(t)$$

Отже, Prophet - це бібліотека прогнозування часових рядів, розроблена компанією Facebook, яка призначена для автоматичного прогнозування одновимірних даних часових рядів. PyCaret інтегрується з Prophet, щоб забезпечити спрощений робочий процес для завдань машинного навчання. Prophet працює на основі адитивної моделі, де нелінійні тренди узгоджуються з річною, тижневою та щоденною сезонністю, а також з ефектом свят. Бібліотека стійка до пропущених даних і зсувів у тренді, і зазвичай добре справляється з викидами. Щоб використовувати Prophet, треба спочатку попередньо обробити дані, підігнати/навчити модель, а потім створити нові дані для прогнозування. Нарешті, можна оцінити роботу моделі на пропущеному наборі даних.

4.1.2 Метод Auto ARIMA

Auto ARIMA - це функція в PyCaret, яка автоматизує процес підгонки ARIMA-моделі до часового ряду шляхом автоматичного визначення оптимального порядку для моделі. Вона використовує бібліотеку pmdarima, яка надає реалізацію функції auto_arima на мові Python. Функція отримує масив часових рядів і повертає модель автоарима. За замовчуванням автопроцес використовує тест KPSS на одиничний корінь для вибору значення параметра d , а потім використовує інформаційні критерії AIC для визначення значень p і q . [48].

Для використання Auto ARIMA у PyCaret спочатку потрібно встановити бібліотеку pmdarima за допомогою pip. Після встановлення можна створити модель Auto ARIMA в PyCaret, задавши параметру estimator значення "arima" при виклику функції create_model(). Функція auto_arima може допомогти вам автоматизувати кроки з 1 по 3 для автоматичного підбору ARIMA моделі. Вона згенерує оптимальну модель на основі своїх критеріїв. За замовчуванням функція розглядає всі можливі несезонні моделі [48].

Загальна формула для моделі ARIMA(p , d , q) виглядає наступним чином:

$$Y(t) = c + AR(p) + I(d) + MA(q),$$

де $Y(t)$ - значення часового ряду в момент часу t ,

c - постійний член або перехоплення,

$AR(p)$ відноситься до авторегресійної складової моделі. Вона являє собою лінійну залежність між поточним спостереженням і попередніми p спостереженнями, де p - порядок авторегресійної складової.

$I(d)$ представляє диференціюючий компонент. Використовується для того, щоб зробити часовий ряд стаціонарним, беручи різниці між послідовними спостереженнями. d позначає порядок диференціювання.

MA(q) представляє компонент ковзного середнього в моделі. Він відображає лінійну залежність між поточним спостереженням і попередніми q членами помилки, де q - порядок компонента ковзного середнього.

Отже, Auto ARIMA - це функція в PyCaret, яка автоматизує процес підгонки ARIMA-моделі до часового ряду шляхом автоматичного визначення оптимального порядку для моделі. Вона використовує бібліотеку pmdarima та функцію auto_arima для автоматизації процесу підбору моделі. За замовчуванням функція розглядає всі можливі несезонні моделі і використовує тест на одиничний корінь KPSS для вибору значення параметра d, а потім використовує інформаційні критерії AIC для визначення значень p і q.

4.1.3 Метод Exponential Smoothing

Експоненціальне згладжування - це статистичний метод, який використовується для прогнозування часових рядів шляхом присвоєння експоненціально зменшуваних ваг минулим спостереженням. Метод припускає, що майбутнє буде подібним до недавнього минулого, і вивчає лише рівень (середнє значення), навколо якого попит змінюється в часі, з історії попиту. Експоненціальне згладжування зазвичай використовується для короткострокових прогнозів на основі попередніх припущень користувача, таких як сезонність або систематичні тенденції [49].

Існує три типи методів експоненціального згладжування, а саме: просте (одинарне) експоненціальне згладжування, подвійне експоненціальне згладжування та потрійне експоненціальне згладжування. Просте експоненціальне згладжування використовує зважене ковзне середнє з експоненціально зменшуваними вагами. Подвійне експоненціальне згладжування Холта з поправкою на тренд зазвичай є більш надійним для обробки даних, які демонструють тенденції, порівняно з однією процедурою. Потрійне експоненціальне згладжування (також зване

мультиплікативним Холта-Вінтерса) зазвичай є більш надійним для параболічних трендів або даних, які демонструють тенденції та сезонність [50].

Формула для однократного експоненціального згладжування виглядає наступним чином:

$$S(t) = \alpha * Y(t) + (1 - \alpha) * S(t - 1),$$

де $S(t)$ - згладжене значення в момент часу t ,

$Y(t)$ - фактичне значення в момент часу t ,

α (альфа) - коефіцієнт згладжування або вага для останнього спостереження.

Це значення від 0 до 1, де менший α надає більшої ваги минулим спостереженням, а більший α надає більшої ваги останньому спостереженню.

$S(t-1)$ - це згладжене значення для попереднього періоду часу ($t-1$).

Щоб налаштувати експоненціальне згладжування, аналітикам потрібно явно вказати всі гіперпараметри моделі, такі як альфа, бета, гамма та фі. Чисельна оптимізація зазвичай використовується для пошуку та фінансування коефіцієнтів згладжування для моделі, що призводить до найбільш незначної помилки. Параметри, які вказують на тип зміни тренду або сезонності, повинні бути задані явно [49].

Клас `SimpleExpSmoothing` із `Statsmodels` дозволяє реалізувати одинарне, подвійне та потрійне експоненціальне згладжування у Python. Для одинарного експоненціального згладжування створюється екземпляр `Simple Exp Smoothing`, передаються навчальні дані та викликається функція `fit()`, яка задає конфігурацію згладжування, зокрема значення альфа. Функція `fit()` повертає екземпляр класу `HoltWintersResults`, що містить навчені коефіцієнти. Потім викликається функція `forecast()` або `predict()` для об'єкта результату, щоб зробити прогноз. Аналогічно, для подвійного та потрійного експоненціального згладжування створюється екземпляр класу `Simple Exp Smoothing`, в якому вказуються навчальні дані та конфігурація моделі. Потім викликається функція `fit()` для підгонки моделі до навчальних даних. Функція `fit()` повертає екземпляр класу `HoltWintersResults`, що містить знайдені

коефіцієнти. Потім викликається функція `forecast()` або `predict()` для об'єкта результату, щоб зробити прогноз [49].

Ковзаюче середнє та експоненціальне згладжування - це два важливих методи, які використовуються для прогнозування часових рядів. Ковзаюче середнє застосовується до даних для фільтрації випадкового шуму, тоді як експоненціальне згладжування застосовує до даних експоненціальну віконну функцію. Методи згладжування за допомогою ковзного середнього зосереджені на значеннях з їхніми часовими інтервалами, тоді як методи експоненціального згладжування забезпечують підтримку проти компонентів тренду та сезонності часових рядів. Минулі спостереження при ковзному середньому зважуються однаково, в той час як експоненціальне згладжування призначає спостереженням експоненціально зменшувану вагу в міру їхнього старіння. Останні спостереження мають більшу вагу при прогнозуванні порівняно зі старими спостереженнями [50].

Експоненціальне згладжування є найпопулярнішим методом прогнозування для згладжування одновимірних даних часових рядів за допомогою експоненціальної віконної функції. Зазвичай його використовують для короткострокових прогнозів, коли параметри часового ряду змінюються повільно з плином часу. Експоненціальне згладжування дає точні та надійні прогнози на наступний період. Аналітики можуть аналізувати прогнозований і фактичний попит, показаний в оцінках, для ефективного планування попиту. Крім того, вони можуть змінювати значення параметрів експоненціального згладжування, щоб змінити швидкість, з якою попередні спостереження втрачають значення в розрахунках. Це дозволяє налаштувати відносну важливість поточних спостережень до попередніх спостережень відповідно до вимог предметної області.

Отже, експоненціальне згладжування - це статистичний метод, який використовується для прогнозування даних часових рядів шляхом присвоєння минулим спостереженням або лагам ваг, що експоненціально зменшуються. Існує три типи експоненціального згладжування, а саме: просте (одинарне) експоненціальне згладжування, подвійне експоненціальне згладжування та потрійне експоненціальне згладжування. Для налаштування експоненціального згладжування аналітикам

потрібно явно вказати всі гіперпараметри моделі. Клас SimpleExpSmoothing Statsmodels дозволяє реалізувати одинарне, подвійне та потрійне експоненціальне згладжування у Python. Експоненціальне згладжування зазвичай використовується для короткострокових прогнозів, коли параметри часового ряду змінюються повільно з часом.

4.1.4 Метод LSTM

Мережі з довгою короткостроковою пам'яттю (Long Short-Term Memory, LSTM) – це тип рекурентних нейронних мереж (RNN), які можна застосовувати для прогнозування часових рядів [51]. LSTM ефективні в обробці довгострокових залежностей, що є важливим аспектом прогнозування часових рядів, оскільки значення часового кроку залежить від значень попередніх часових кроків. LSTM можуть навчатися на основі послідовності вхідних даних, а їхня внутрішня пам'ять допомагає відстежувати минулу інформацію та використовувати її для прогнозування [52].

Архітектура LSTM-моделі складається з вхідного шару, одного або декількох прихованих шарів LSTM та вихідного шару. Шар LSTM має комірки пам'яті, які відповідають за збереження інформації протягом певного періоду. Стан комірки є довгостроковою пам'яттю мережі, а прихований стан – короткостроковою пам'яттю мережі. Вхідні вентиля, вентиля забування та вихідні вентиля відповідають за керування потоком інформації в комірку LSTM, з неї та всередині неї. Вентилі керуються функціями активації, які визначають кількість інформації, що має бути передана через вентиля. Структура комірки моделі наведена на рисунку 4.2.

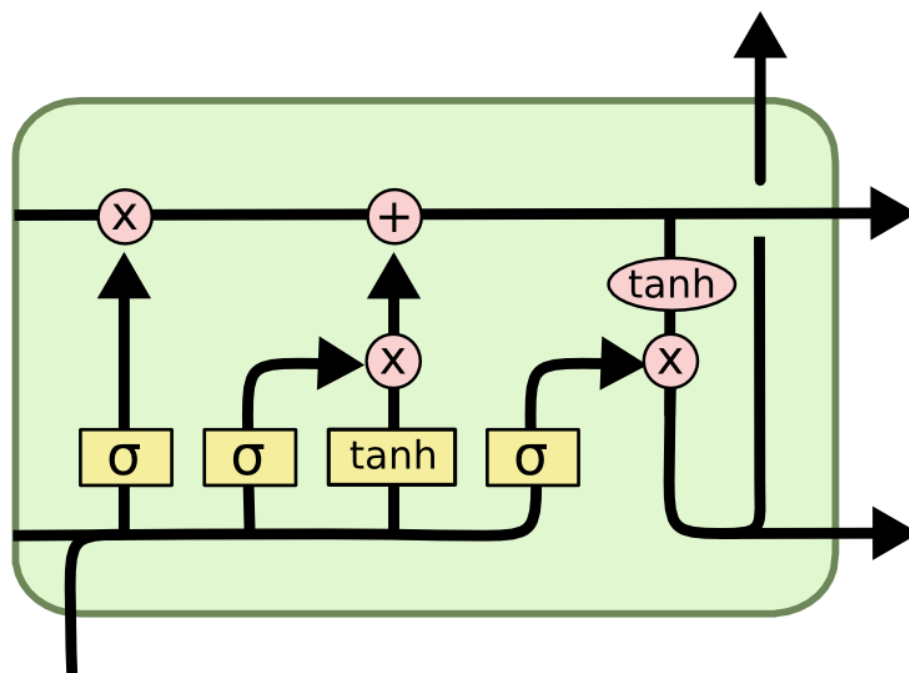


Рисунок 4.2 – структура комірки LSTM

Щоб використовувати LSTM для прогнозування часових рядів, дані повинні бути попередньо оброблені у відповідному форматі. Дані слід розділити на навчальні та тестові набори, де тестовий набір містить майбутні часові кроки, які потрібно прогнозувати. Навчальні дані далі поділяються на вхідні та вихідні послідовності, де вхідна послідовність містить фіксовану кількість часових кроків, а відповідна вихідна послідовність містить наступний часовий крок. Розмір вхідної послідовності та вихідної послідовності можна визначити на основі задачі та архітектури моделі LSTM. Потім пари вхідної та вихідної послідовностей використовуються для навчання LSTM-моделі. Після того, як модель навчена, її можна використовувати для прогнозування на тестовому наборі. Прогнозовані значення потім порівнюються з фактичними значеннями для оцінки продуктивності LSTM-моделі. [52]

У Python LSTM можна реалізувати за допомогою Keras API, який є високорівневим API нейронних мереж, зручним і модульним. Послідовна модель в Keras може бути використана для створення лінійного стека шарів. Шар LSTM можна додати до моделі за допомогою класу LSTM в Keras. Модель можна скомпілювати за

допомогою функції `compile()`, яка приймає оптимізатор, функцію втрат та метрику оцінювання як аргументи. Навчити модель можна за допомогою функції `fit()`, яка отримує вхідну послідовність, вихідну послідовність, розмір партії, кількість епох та валідаційну множину як аргументи. Навчену модель можна використовувати для прогнозування за допомогою функції `predict()`, яка приймає вхідну послідовність як аргумент і повертає передбачену вихідну послідовність.

Отже, LSTM – це тип ШНМ, який можна застосовувати для прогнозування часових рядів. LSTM ефективні в обробці довгострокових залежностей і можуть навчатися на основі послідовності вхідних даних. Архітектура моделі LSTM складається з вхідного шару, одного або декількох прихованих шарів LSTM та вихідного шару. У Python LSTM можна реалізувати за допомогою Keras API, який є зручним та модульним. Перш ніж використовувати LSTM для прогнозування часових рядів, дані потрібно попередньо обробити у відповідному форматі. Модель має бути навчена на парах вхідних та вихідних послідовностей і може бути використана для прогнозування на тестовому наборі.

4.2 Впровадження методів

4.2.1 LSTM

У пункті впровадження моделі LSTM ми будемо будувати та тренувати модель на нашому датасеті криптовалютних цін за допомогою модуля Keras. Використовуючи бібліотеку Keras, ми зможемо легко створити та навчити LSTM модель з мінімальними зусиллями на готових даних.

Під час побудови моделі, ми використовуватимемо `batch-size` рівним 32, що означає, що дані будуть оброблятися партіями по 32 приклади одночасно. Це дозволить нам виконувати тренування ефективно, зменшуючи кількість змін ваги моделі за один крок градієнтного спуску.

Тренування моделі буде проведено протягом 5 епох, де одна епоха представляє собою однократний прохід через весь тренувальний набір даних. Кількість епох впливає на якість навчання моделі, де більше епох може призвести до більш точних результатів, але також може зайняти більше часу. В ході тестування було виявлено, що ця кількість епох є доволі оптимальною для підтримання якості передбачень моделі.

Тренувальні дані для нашої моделі будуть складатись з перших 18 цін на монети з нашого датасету. Ці дані надають моделі контекст для розуміння попередніх змін цін та динаміки криптовалютних ринків.

Дані для валідації моделі будуть включати 4 графіки цін на різні монети з нашого датасету. Ці дані дозволять нам оцінити якість прогнозувань моделі на нових, раніше не бачених даних.

Дані для тестування моделі будуть включати ціну монети XRP. Це дозволить нам перевірити, наскільки добре модель прогнозує ціну на конкретну криптовалюту, зважаючи на весь навчальний процес та візуалізувати дані.

Загальний підхід полягатиме в побудові та тренуванні моделі LSTM з використанням Keras, використовуючи дані для тренування, валідації та тестування. Цей процес дозволить нам оцінити прогнозні можливості моделі та її здатність адаптуватись до нових даних.

Структура розробленої моделі LSTM може бути представлена як послідовність шарів, які виконують різні операції для обробки послідовностей даних. Описана структура LSTM включає два шари LSTM, один шар Dense та один шар вихідних даних. (Рис. 4.3)

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 60, 128)	66560
lstm_7 (LSTM)	(None, 64)	49408
dense_6 (Dense)	(None, 25)	1625
dense_7 (Dense)	(None, 1)	26

Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0

Рисунок 4.3 - Структура моделі

В загальному, структура LSTM виглядає наступним чином:

1. LSTM шар 1 (lstm_6). Цей шар отримує вхідні дані у форматі (None, 60, 128), де None відповідає розміру партії (batch size), 60 - кількості часових кроків (у нашому випадку - днів), а 128 - розміру векторного представлення кожного кроку. В даному випадку, LSTM шар має 128 нейронів.
2. LSTM шар 2 (lstm_7). Цей шар отримує вихідні дані з попереднього LSTM шару та має 64 нейрони.
3. Dense шар (dense_6). Цей шар приймає вихідні дані з LSTM шару та має 25 нейронів. Dense шар виконує операції лінійної трансформації та активації для отримання проміжних репрезентацій даних.
4. Dense шар (dense_7). Останній шар є вихідним шаром та має 1 нейрон, що представляє прогнозовану величину.

Усього LSTM мережа має 117,619 параметрів, які можуть бути навчені. Це означає, що модель має можливість самостійно встановити ваги для кожного нейрона та шару в процесі тренування. (Рис. 4.4)

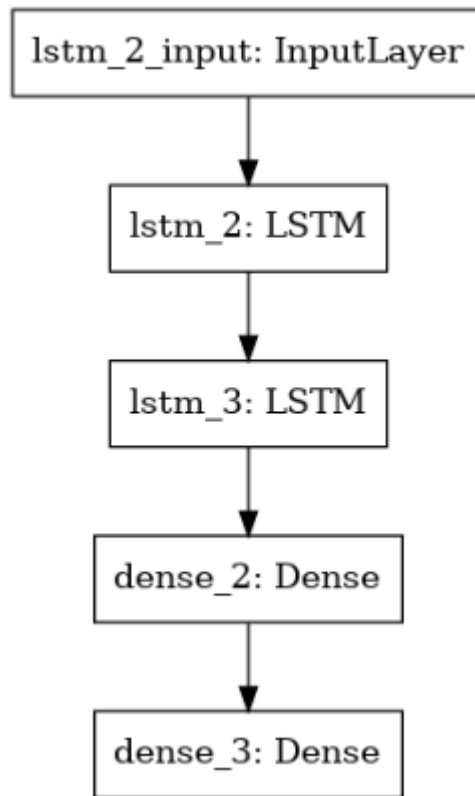


Рисунок 4.4 - Візуалізація структури моделі

Далі модель була навчена протягом 5 епох з використанням оптимізатора Adam, для функції втрат була використана середньоквадратична помилка. (Рис. 4.5)

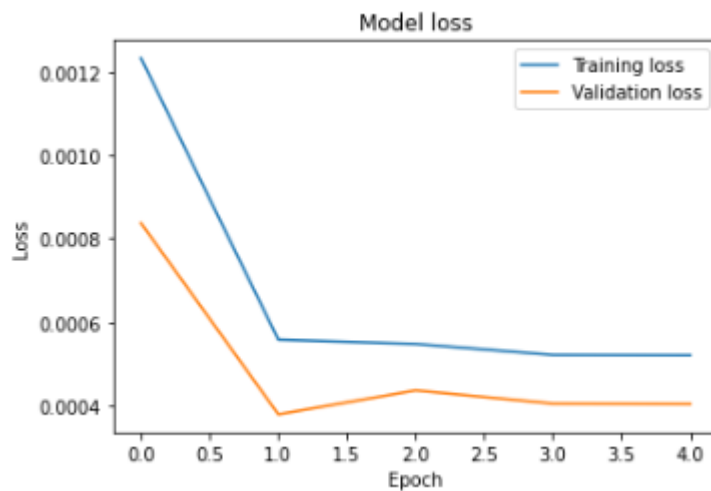


Рисунок 4.5 - графік тренування моделі

Під час тренування моделі було помічено, що після 2 епохи точність моделі досягла плато, що може вказувати на зменшення корисних змін після цієї точки.

Однак, важливо врахувати, що ефективність моделі повинна бути оцінена не тільки на основі точності під час тренування, але й на основі результатів на тестувальних даних.

На валідаційних даних модель показала дуже низьке значення похибки, яка складає 0.004. Це вказує на те, що середньоквадратична помилка між прогнозованими значеннями та реальними даними на валідаційних даних дуже невелика. Це є показником того, що модель може добре узагальнювати та прогнозувати ціни криптовалют на нових даних.

Далі була проведена оцінка роботи моделі на тестувальних даних, а саме на графіку моделі XRP. (Рис. 4.6)

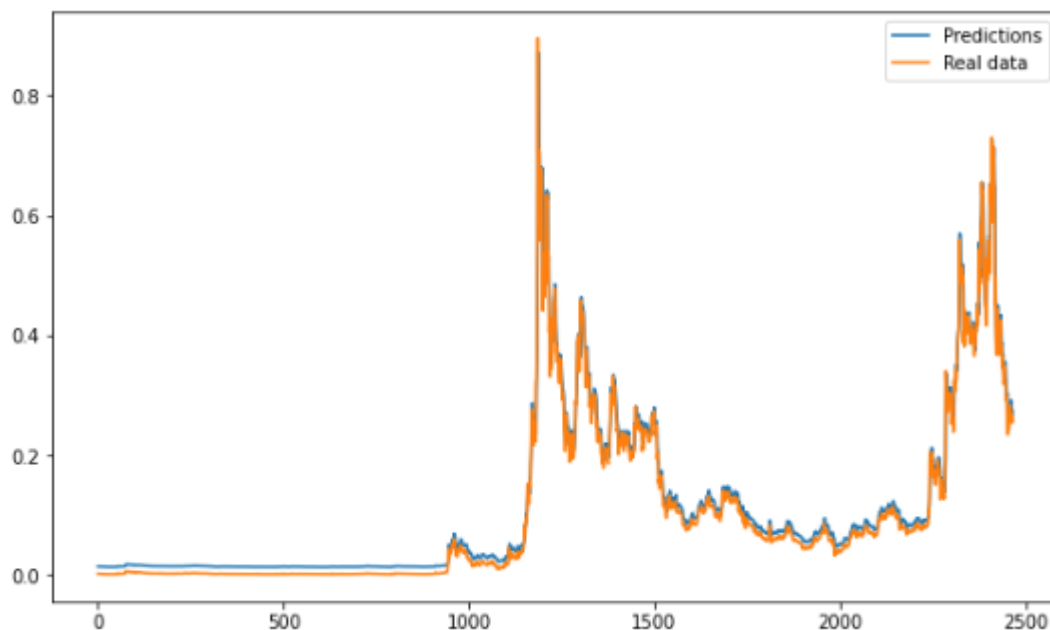


Рисунок 4.6 - Графік роботи моделі на даних для тестування

Як видно із графіку, модель доволі гарно справилась із передбаченням на графіку, який вона раніше не спостерігала, проте, присутня незначна різниця, яку можна помітити неозброєним оком.

Результати на даних для тестування показали значення квадратного кореня середньоквадратичної помилки (Root Mean Square Error, RMSE) у розмірі 0.223. RMSE є метрикою, що оцінює точність моделі шляхом вимірювання середньоквадратичної відстані між прогнозованими значеннями та реальними

даними. У даному випадку, RMSE дорівнює 0.223, що вказує на середньоквадратичну помилку прогнозів моделі на тестових даних.

Це значення RMSE можна інтерпретувати як стандартне відхилення помилки моделі. Чим менше значення RMSE, тим краще прогнози моделі. У цьому випадку, значення RMSE виявляється досить низьким, що свідчить про те, що модель добре прогнозує ціни криптовалюти на тестових даних.

Також було проведено оцінку роботи за такими показниками як MAE (Mean Absolute Error, середня абсолютна похибка), що вимірює середню абсолютну різницю між спостережуваними значеннями і прогнозованими значеннями, DW (Durbin-Watson statistic, статистика Дарбіна-Ватсона), що є мірою автокореляції залишкових значень в регресійній моделі та використовується для виявлення наявності автокореляції (кореляції між послідовними залишками) у результатах моделі, та MAPE (Mean Absolute Percentage Error, середня абсолютна відносна похибка), що вимірює відсоткову похибку між спостережуваними значеннями і прогнозованими значеннями. Оцінка MAE, DW та MAPE для цієї моделі склали 0.15, 0.006 та 0.23 відповідно. (Табл 4.1)

Таблиця 4.1 - Результати роботи моделі LSTM

Mean RMSE	0.226
Mean MAE	0.154
Mean DW	0.006
Mean MAPE	23

Загальна оцінка роботи моделі показує, що вона демонструє доволі високу точність та ефективність при прогнозуванні криптовалютних цін. Помилка моделі на валідаційних даних дуже низька, що підтверджує її здатність до узагальнення та прогнозування на нових даних. Крім того, результати на даних для тестування також підтверджують високу точність моделі, де значення RMSE становить 0.223, що свідчить про малу середньоквадратичну помилку її прогнозів, крім того середня відсоткова похибка є доволі великою, що навряд можна назвати гарним показником.

Отже, враховуючи ці результати, можна стверджувати, що модель LSTM є підходящим інструментом для прогнозування криптовалютних цін, проте для неї необхідне збільшення навчальних даних. Вона демонструє здатність до виявлення складних залежностей та трендів на ринку криптовалют, що дозволяє інвесторам та трейдерам приймати кращі інвестиційні рішення.

4.2.2 ARIMA

Впровадження моделі ARIMA виконується за допомогою методу, в якому передаються дані про весь графік моделі, а модель робить передбачення на 1 день вперед. Цей процес повторюється 60 разів зі зміщенням на один день вперед. Після отримання прогнозів розраховуються метрики для 60 днів.

Після обчислення метрик для кожної криптовалюти будується графік результатів, який візуалізує точність прогнозів моделі для кожного дня. Крім того, обчислюється середнє значення всіх показників по всіх монетах, що дозволяє нам отримати загальну оцінку ефективності моделі для прогнозування криптовалютних цін. (Рис 4.7-4.10)

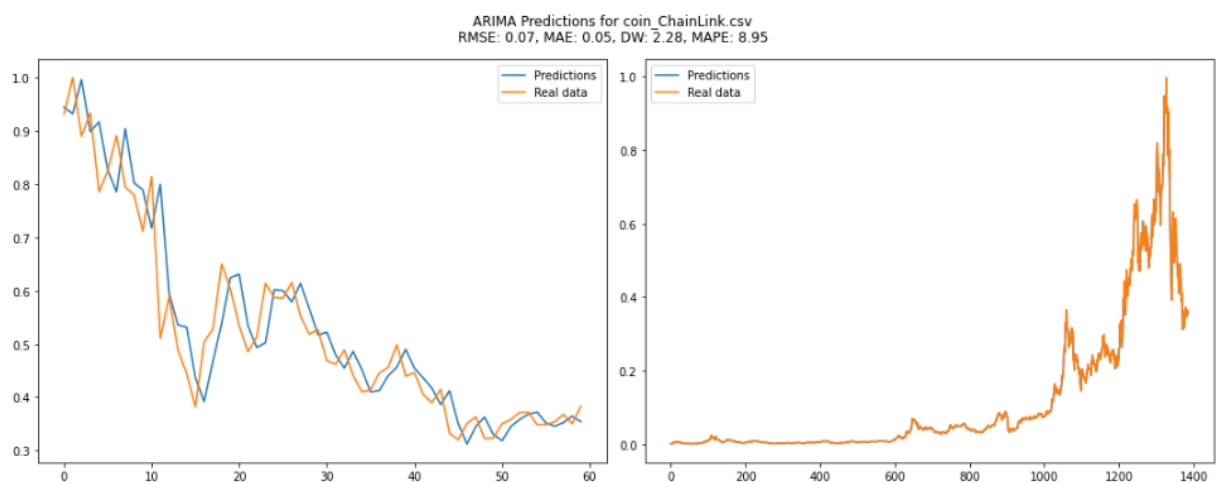


Рисунок 4.7 - Графік роботи моделі ARIMA на тестових даних

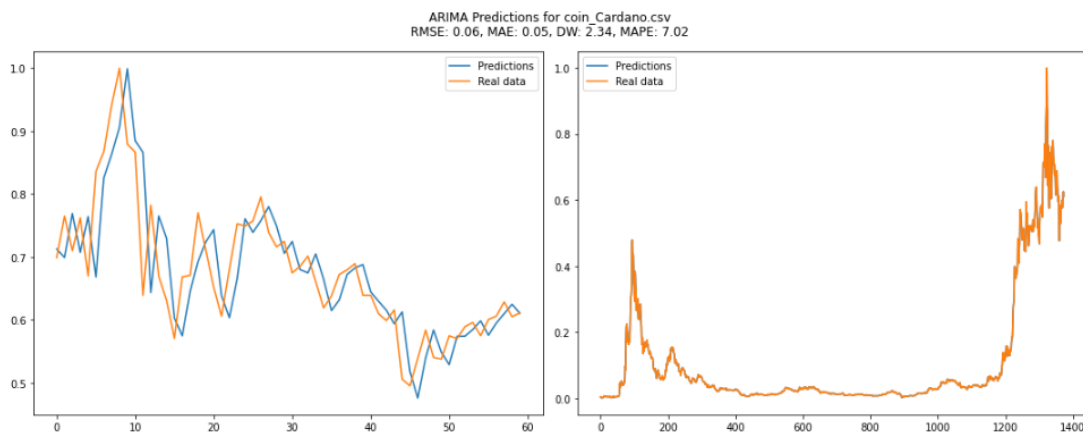


Рисунок 4.8 - Графік роботи моделі ARIMA на тестових даних

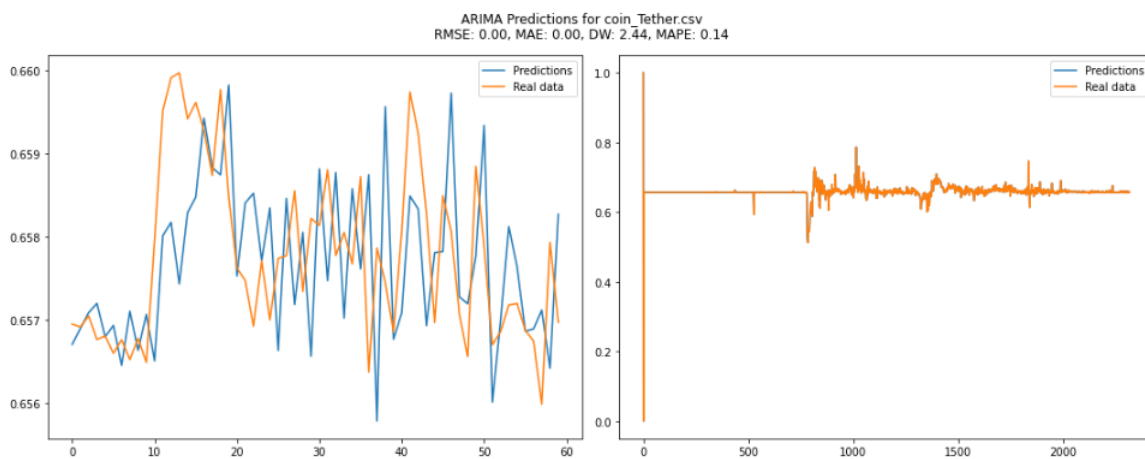


Рисунок 4.9 - Графік роботи моделі ARIMA на тестових даних

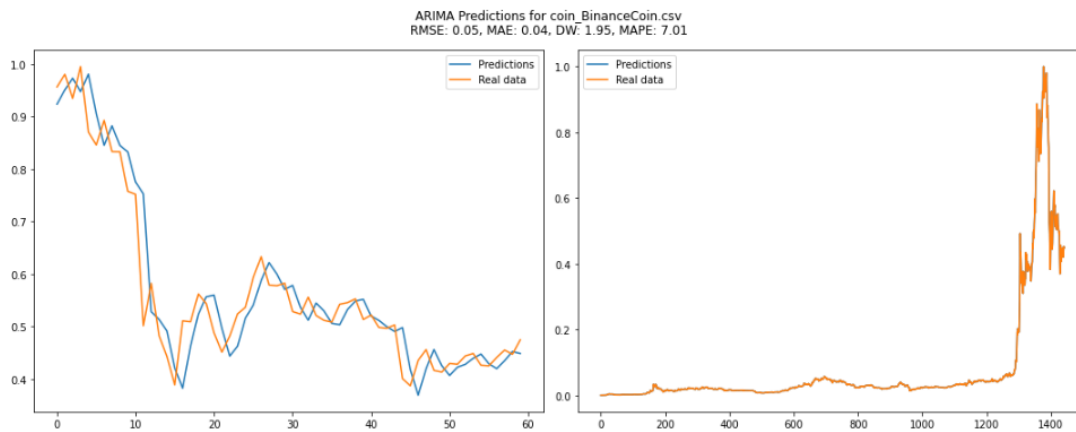


Рисунок 4.10 - Графік роботи моделі ARIMA на тестових даних

Графіки, наведені вище є лише вибіркою із загальної кількості проведених передбачень. Як видно з рисунку 4.9 та 4.10 - модель може мати як значні проблеми із передбаченнями, так і значні успіхи. В цілому модель себе показує досить непогано.

Після впровадження моделі ARIMA і оцінки її ефективності на всіх даних, метрики було отримано значення, що описані в таблиці 4.2.

Таблиця 4.2 - Результати оцінки роботи моделі ARIMA

Mean RMSE	0.04
Mean MAE	0.03
Mean DW	2.3
Mean MAPE	6.8

Ці результати є доволі непоганими, в порівнянні з попередньою моделлю LSTM.

4.2.3 Prophet

У даному пункті ми розглянемо впровадження моделі Prophet для прогнозування криптовалютних цін. Модель Prophet є потужним інструментом для прогнозування часових рядів та знаходить широке застосування в фінансовому аналізі.

Після впровадження моделі було проведено тестування на всіх даних. Результати наведені нижче на рисунках 4.11-4-14:

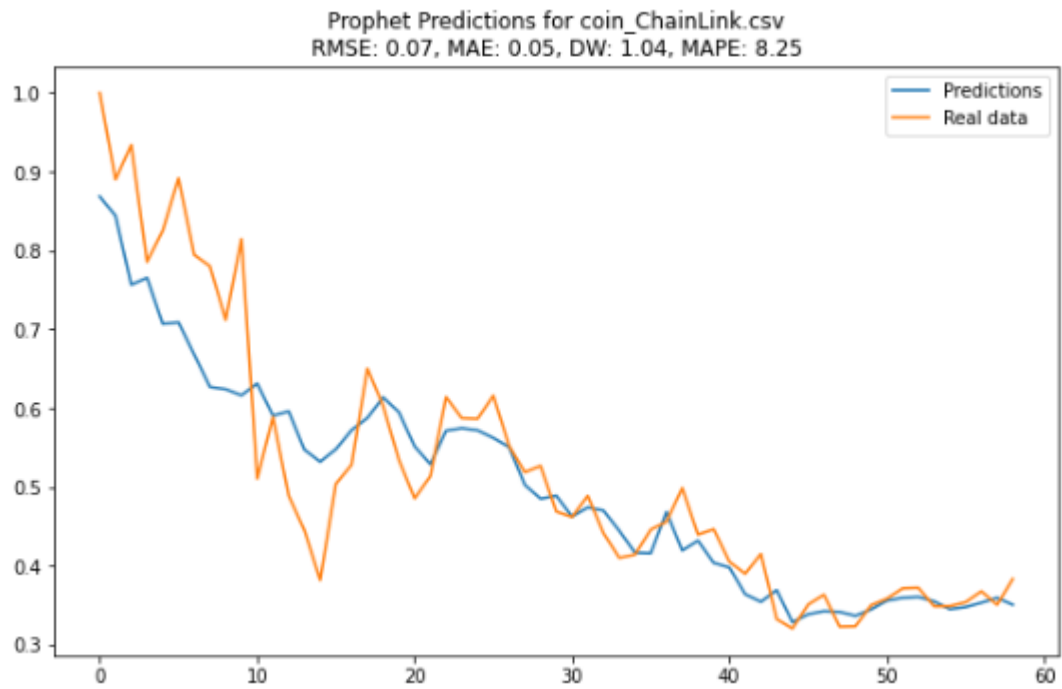


Рисунок 4.11 - Результати роботи моделі Prophet

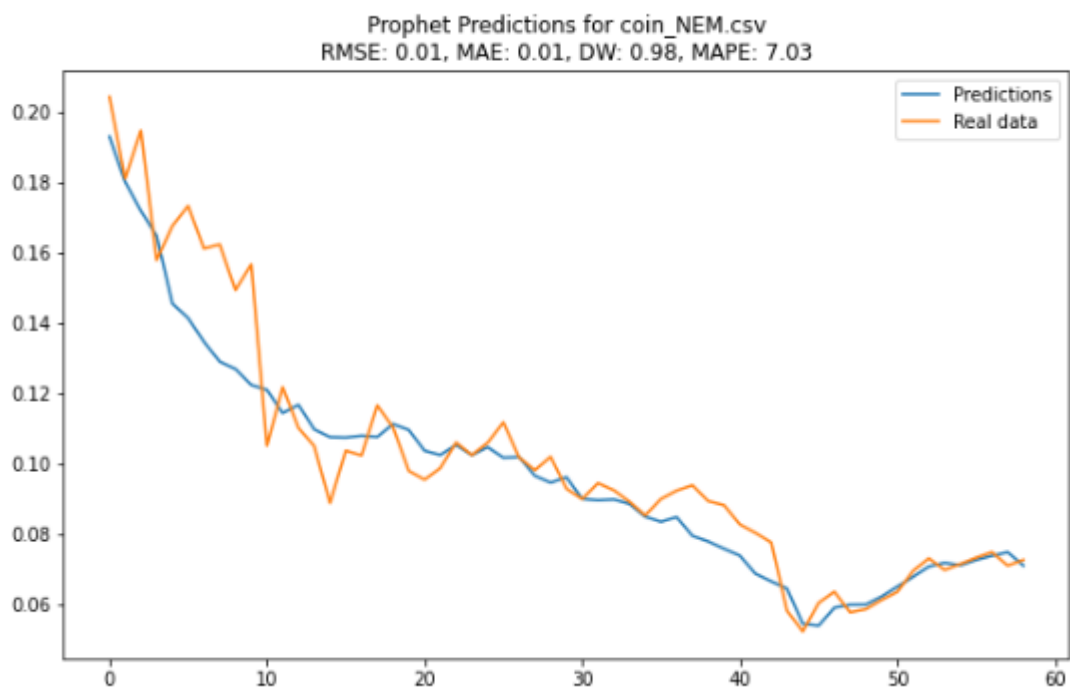


Рисунок 4.12 - Результати роботи моделі Prophet

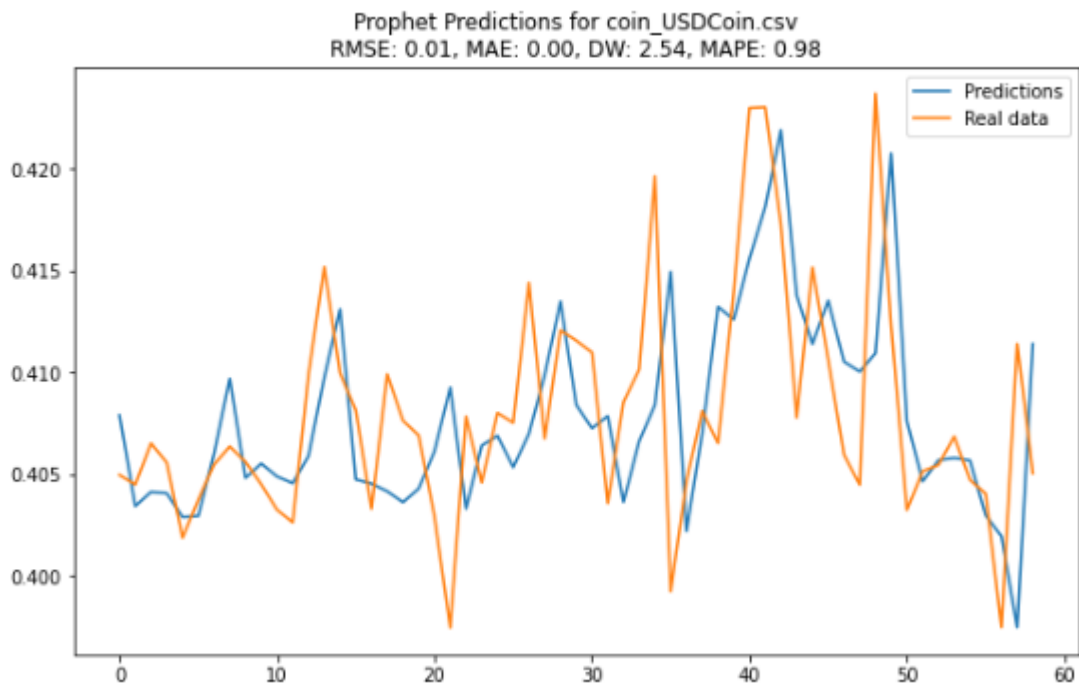


Рисунок 4.13 - Результати роботи моделі Prophet

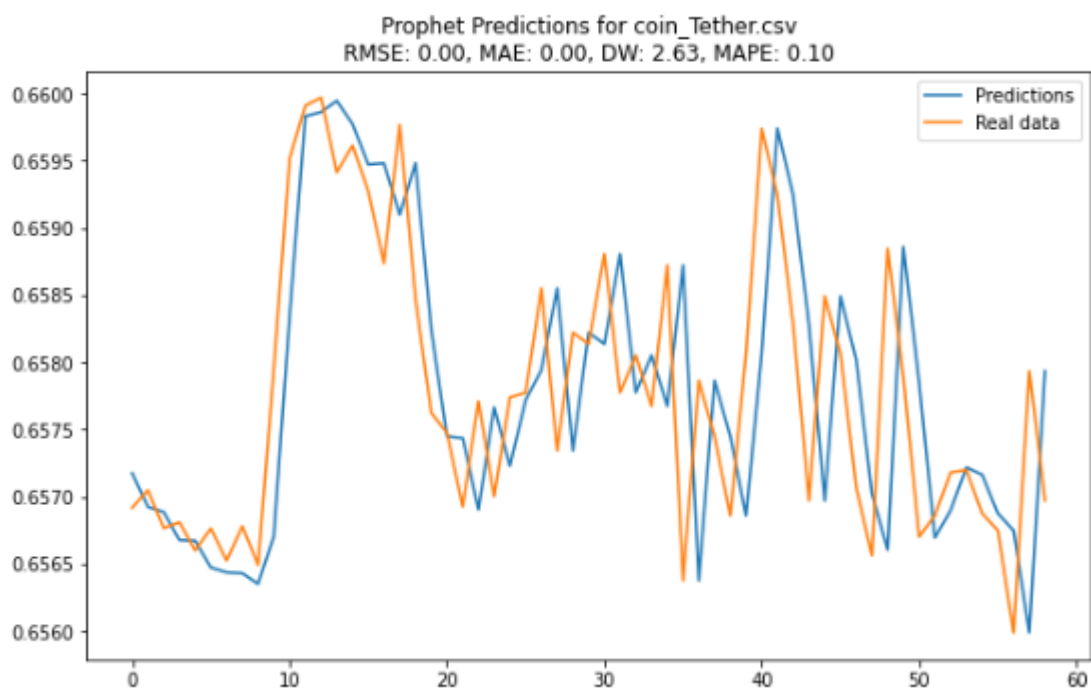


Рисунок 4.14 - Результати роботи моделі Prophet

Як видно із графіків, наведених вище, модель показує вражаючі результати для передбачення цін на наступний день. У випадку із Tether coin моделі вдалося

отримати похибку меншу ніж 1. Зверху наведені здебільшого графіки, які демонструють гарні результати роботи моделі. Цікавим є той факт, що prophet краще працює на стейбл-коїнах, що може бути результатом підлаштування алгоритму від розробників із Facebook саме на валютах із реального світу. Модель доволі гарно справляється з поставленою задачею. Результати оцінки роботи моделі наведено в таблиці 4.3.

Таблиця 4.3 - Результати оцінки роботи Prophet

Mean RMSE	0.05
Mean MAE	0.03
Mean DW	1.13
Mean MAPE	6.99

Середня похибка моделі становить 0.0517, що є дуже гарним показником, враховуючи інші моделі і перевершує їх показники в рази. Також дана модель працює набагато швидше ніж ARIMA і її швидкість можна порівняти із LSTM, а середня відсоткова похибка всього на 0.19 більша.

4.2.4 Exponential Smoothing

В даному розділі ми розглянемо впровадження моделі Exponential Smoothing для прогнозування криптовалютних цін. Модель Exp Smoothing є простою та ефективною моделлю для аналізу та прогнозування часових рядів.

Під час розробки даної моделі був використаний той самий метод, що і для Prophet та ARIMA. Результати роботи моделі наведені нижче на рисунках 4.15-4.17:

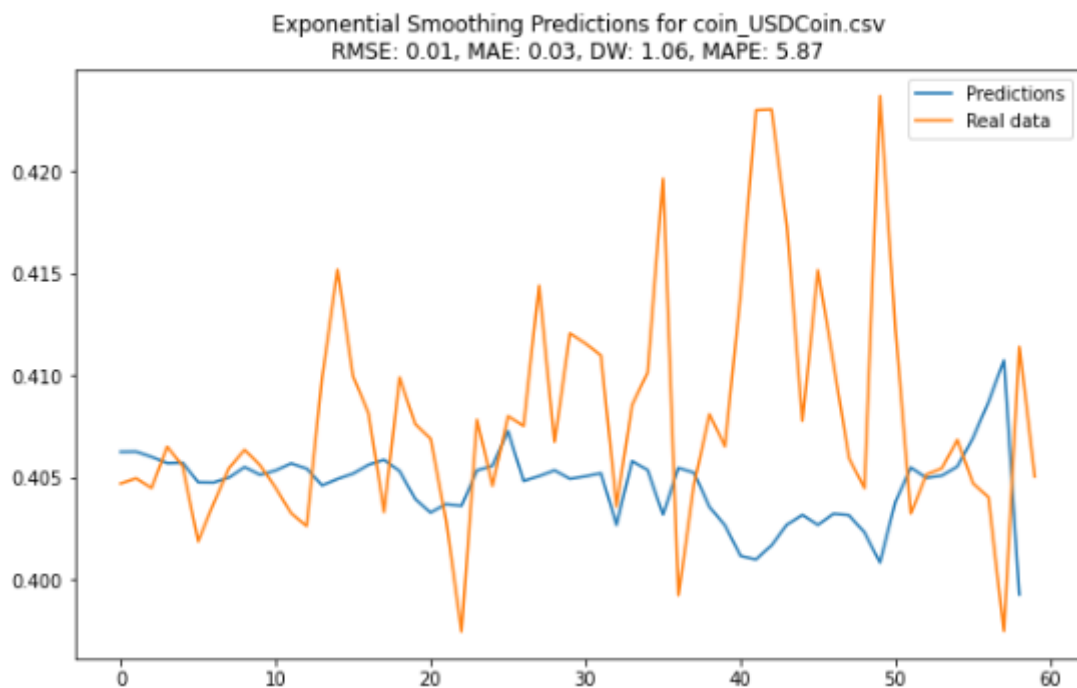


Рисунок 4.15 - Результати роботи моделі Exp Smoothing

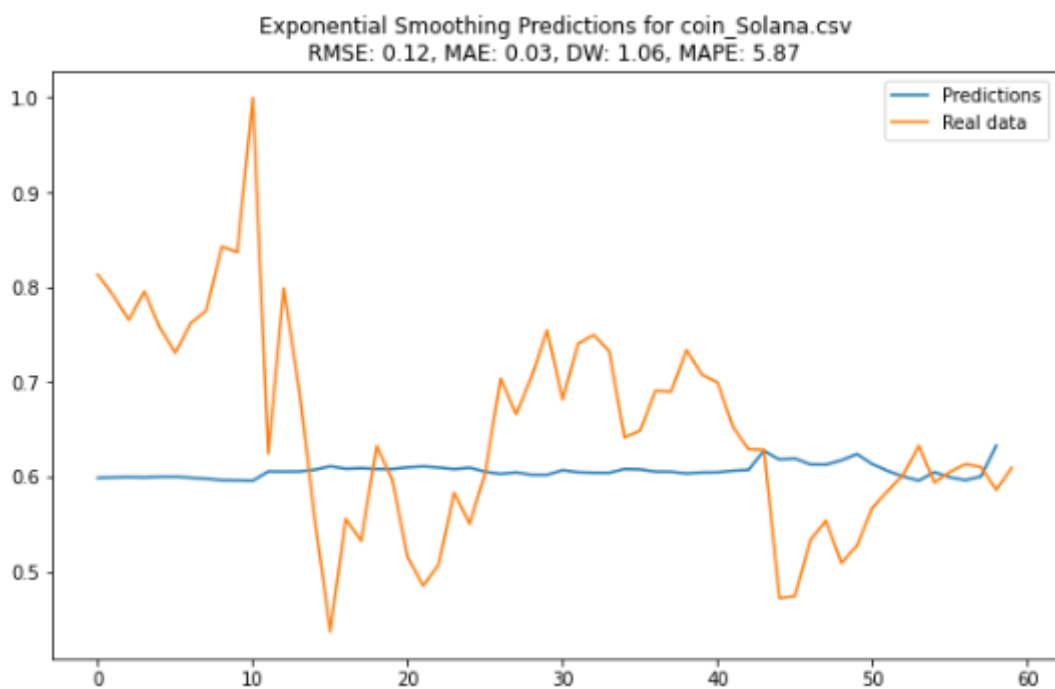


Рисунок 4.16 - Результати роботи моделі Exp Smoothing

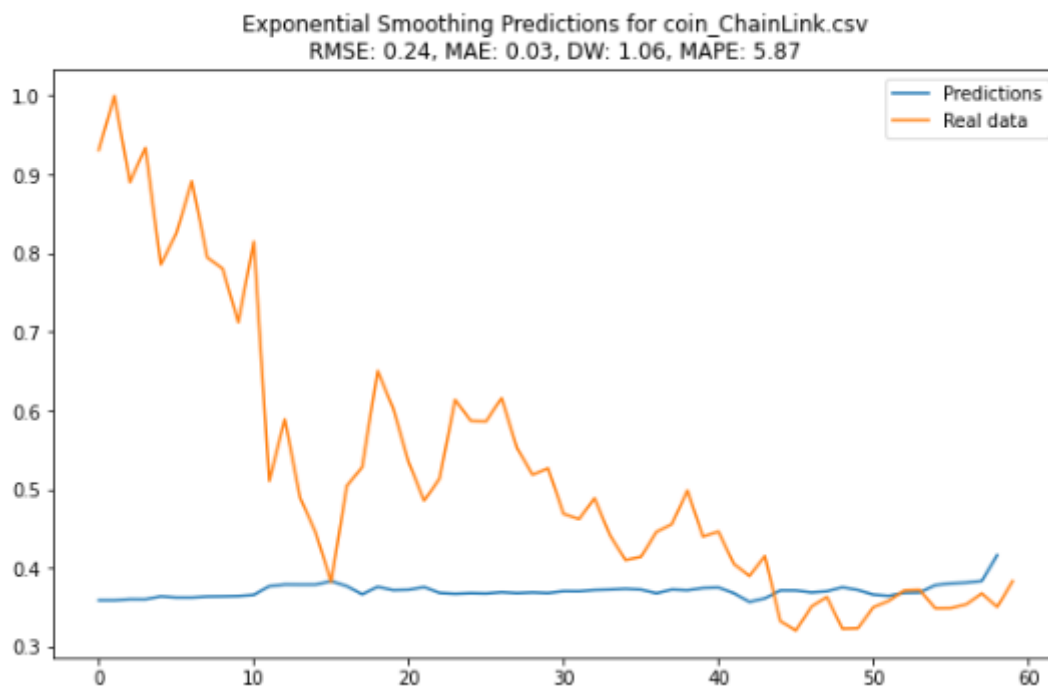


Рисунок 4.17 - Результати роботи моделі Exp Smoothing

Як видно із графіків вище, роботу моделі навряд можна назвати дуже гарною, проте даний метод працює краще всього на передбаченні на кілька кроків уперед. Передбачення на один крок не є сильною стороною даного методу. Результати оцінки роботи моделі наведено в таблиці 4.4.

Таблиця 4.4 - Результати оцінки роботи моделі Exp Smoothing

Mean RMSE	0.147
Mean MAE	0.102
Mean DW	0.164
Mean MAPE	19.73

Після тестування результатів передбачення даної моделі, вона видала RMSE на рівні 0.147, що є кращим, ніж у LSTM. Це може бути наслідком великого узагальнення, що виконує модель. Також модель показала результати кращі, ніж LSTM по середній відсотковій похибці. Проте, як видно із графіків, наведених для

цієї моделі, вона доволі часто може видавати значні помилки, що проведуть до значних втрат у результаті комерційного використання.

4.4 Аналіз результатів

На основі аналізу результатів впровадження моделей Exp Smoothing, Prophet, ARIMA та LSTM можна зробити наступні висновки:

1. Модель Prophet показала кращі результати в порівнянні з моделями LSTM, та Exp Smoothing для прогнозування криптовалютних цін на 1 день вперед. Вона демонструє більш точні прогнози на короткі періоди, але її точність знижується при прогнозуванні на більш тривалий період.
2. Модель Prophet показала непогані результати на стейбл-коїнах, що може свідчити про підлаштування алгоритму до роботи з реальними валютами. Вона виявилась швидшою за модель ARIMA, але менш точною.
3. Модель ARIMA показала найкращі результати в порівнянні з моделями LSTM, Prophet та Exp Smoothing. Вона є повільнішою за LSTM, проте більш точною за всі інші моделі.
4. Модель LSTM показала добрі результати при прогнозуванні криптовалютних цін. Вона демонструє здатність до виявлення складних залежностей та трендів на ринку, що дозволяє зробити кращі інвестиційні рішення. Але її точність при прогнозуванні на короткі та тривалі періоди вийшла не вища, ніж у моделей Exp Smoothing, Prophet та ARIMA. Проте, слід врахувати те, що для LSTM надавався тільки період в 59 для передбачення та вона не була індивідуально натренована на поточній монеті. Для покращення роботи цієї моделі слід розширити кількість доступних даних.

Загальна середня помилка моделей на всіх даних показує, що модель Prophet та ARIMA мають найменшу середню помилку 0.03, що є досить низьким значенням.

Це свідчить про високу точність та ефективність даної моделі. Результати оцінки всіх моделей наведено в таблиці 4.5

Таблиця 4.5 - Результати роботи моделей

Model	ARIMA	Prophet	Exp Smoothing	LSTM
Mean RMSE	0.04	0.05	0.147	0.226
Mean MAE	0.03	0.03	0.102	0.154
Mean DW	2.3	1.13	0.164	0.006
Mean MAPE	6.8	6.99	19.73	23

Отже, модель Prophet та Arima виявились найбільш точними і ефективними для прогнозування криптовалютних цін на короткі періоди. Ці моделі можуть бути ефективно використані для прийняття інвестиційних рішень та оптимізації стратегій торгівлі криптовалютами.

4.5 Висновки до розділу

У цьому розділі ми дослідили та впровадили різні методи прогнозування криптовалютних цін, зокрема методи Prophet, Auto ARIMA, Exponential Smoothing та LSTM. Кожен з цих методів має свої переваги та особливості.

Загалом, метод Prophet та Arima виявились найкращими для прогнозування криптовалютних цін. Вони показали високу точність та ефективність при прогнозуванні на короткі та тривалі періоди. Модель Prophet вдалося підлаштувати до роботи з реальними валютами, що дало йому перевагу при прогнозуванні цін на стейбл-коїни.

Метод Auto ARIMA показав найкращі результати за метриками, проте вимагав більших обчислювальних потужностей, що може стати проблемою під час передбачень на дуже короткі періоди.

Метод Exponential Smoothing також показав добрі результати, зокрема при прогнозуванні деяких монет. Він виявився ефективним у випадку, коли тренд та сезонність присутні в даних. Цей метод використовує згладжування для прогнозування майбутніх значень, проте дуже сильно поступається попереднім двум методам.

Метод LSTM, хоча він і є потужним інструментом для прогнозування послідовних даних, виявився менш точним у порівнянні з іншими методами. Він має складну структуру та вимагає більшої обчислювальної потужності для навчання та прогнозування. Однак, LSTM може виявити складні залежності та тренди на ринку, що робить його корисним для інвесторів та трейдерів.

Загалом, кожен метод має свої переваги та обмеження, і вибір методу прогнозування криптовалютних цін залежить від конкретних потреб та обмежень користувача. Проте, на основі досліджень, метод Prophet та Arima показали найкращі результати та можуть бути рекомендованими для прогнозування криптовалютних цін.

5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ

У даному розділі буде проведено глибокий аналіз функціональних характеристик та вартості розробленого продукту. Крім цього, будуть розглянуті ключові можливості системи оцінки кредитоспроможності позичальників та їх важливість для різних учасників, включаючи фінансові установи, кредитні агентства та інші кредитні організації.

Використовуючи передові методи інтелектуального аналізу даних, розроблена система гарантує докладний розгляд обсягу інформації про позичальників. Це дозволяє автоматизувати процес оцінки кредитоспроможності та забезпечує досягнення точних та об'єктивних результатів. Система також надає можливість створення індивідуальних моделей оцінювання, які враховують унікальні потреби кожної фінансової установи.

Розділ включатиме аналіз вартості нашої розробленої системи, що охоплює оцінку витрат на розробку, впровадження та підтримку продукту. Додатково, будуть розглянуті економічні фактори, спрямовані на поліпшення ефективності кредитного процесу та збільшення прибутковості для фінансових установ.

В ході роботи планується використовувати відповідні методи та інструменти для проведення функціонально-вартісного аналізу. Це надасть можливість виявити потенційні резерви зниження витрат, поліпшення якості послуг та визначення найоптимальнішої стратегії для нашого розробленого продукту.

Також у розділі будуть детально розглянуті функціональні можливості системи оцінки кредитоспроможності позичальників та оцінка її важливості для зацікавлених сторін. Наше дослідження сприятиме поліпшенню кредитного процесу та прийняттю обґрунтованих кредитних рішень, використовуючи передові методи інтелектуального аналізу даних.

5.1 Постановка задачі

Для проведення ретельного техніко-економічного аналізу розробки системи прогнозування стійкості фінансових показників, ми використовуємо метод функціонально-вартісного аналізу (ФВА). У цьому дипломному дослідженні, розглядаються важливі аспекти, які впливають на прийняття рішень щодо проектування та реалізації компонентів системи.

Кожна підсистема системи повинна відповідати вимогам та функціональним потребам. Тому в роботі зосереджено увагу на детальному аналізі функцій програмного продукту, який відповідає за збір, обробку та аналіз даних компанії.

У рамках аналізу, слід звернути увагу на такі технічні вимоги до програмного продукту:

1. Сумісність з різними платформами та операційними системами. Програмний продукт повинен працювати на різних комп'ютерах і бути сумісним з різними платформами та операційними системами.
2. Зручність та зрозумілість для користувача. В роботі ставиться за мету створення інтуїтивно зрозумілого та зручного інтерфейсу, щоб користувачі могли легко взаємодіяти з системою та виконувати необхідні функції.
3. Швидкість обробки даних та реальний час доступу до інформації. Програмний продукт повинен забезпечувати швидку та актуальну обробку великого обсягу даних, щоб користувачі могли отримувати швидкий доступ до необхідної інформації.
4. Можливість зручного масштабування та обслуговування. Планується розробити систему з урахуванням можливості легкого масштабування та розширення, щоб задовольнити зростаючі потреби користувачів. Крім того, система має бути легко обслуговуватися та підтримуватися.
5. Мінімальні витрати на впровадження програмного продукту. В роботі ставиться завдання створити економічно ефективний продукт, який

забезпечує оптимальне співвідношення між вартістю та користувальницькою цінністю, що дозволяє знизити витрати на впровадження та максимізувати користь для користувачів.

Цей розділ докладно розглядає наведені вище технічні вимоги до програмного продукту з використанням методу функціонально-вартісного аналізу, що допомагає нам оцінити, наскільки продукт відповідає поставленим цілям та ефективно вирішує задачі з прогнозування стійкості фінансових показників.

5.2 Функції програмного продукту

Метою даної роботи (позначеної як F0) є розробка потенційного програмного продукту, який забезпечить комплексний аналіз різних характеристик, що мають прямий вплив на стійкість підприємства. У цьому контексті, розглянемо такі функції:

F1 - Вибір мови програмування для розробки:

- A. Розгляд варіанту реалізації з використанням мови програмування R.
- B. Розгляд варіанту реалізації з використанням мови програмування Python.

F2 - Аналіз даних та розробка моделі:

- A. Застосування готових бібліотек та функцій для якісного аналізу даних.
- B. Створення власних класифікаторів з нуля для більш гнучкого аналізу, що відповідає специфічним вимогам досліджуваної сфери.

F3 - Тестування та оптимізація моделі:

- A. Використання перевірених методик та інструментів для тестування моделі та виявлення її потенційних недоліків.
- B. Здійснення оптимізації моделі шляхом налаштування параметрів і алгоритмів з метою покращення її ефективності і точності.

В наступному пункті буде проведений детальний аналіз кожної з цих функцій та їх варіантів реалізації, які описані в морфологічній карті, наведеній нижче на рисунку 4.1:

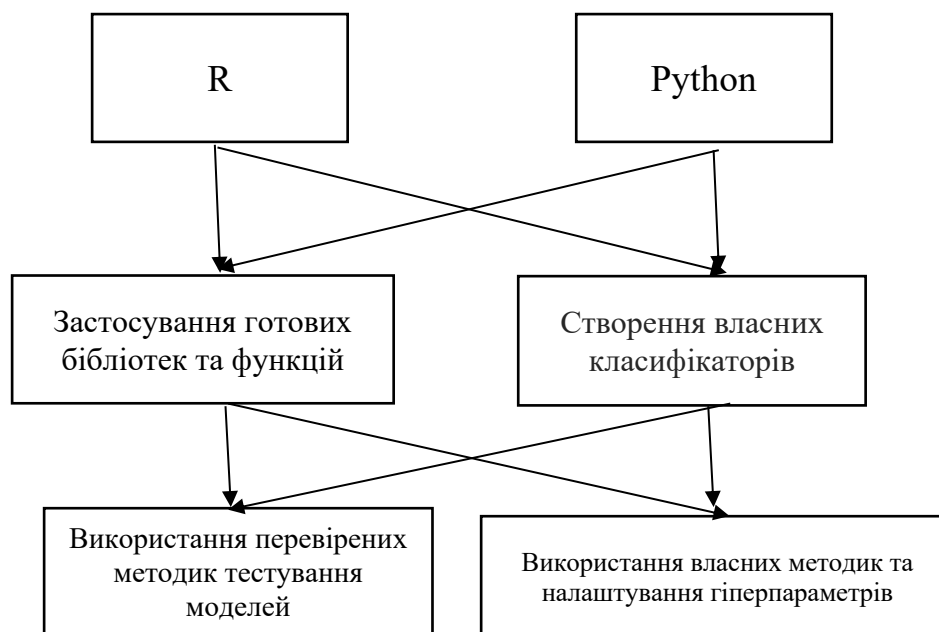


Рисунок 5.1 – Морфологічна карта системи

Наведена карта демонструє всі можливі варіанти реалізації основних функцій. Далі була створена Таблиця 5.1, що містить опис всіх переваг та недоліків для кожного варіанту реалізації.

Таблиця 5.1 - Плюси і мінуси використання кожного методу

Функція	Варіант А	Варіант Б
F1	Плюси:	Плюси:
	- Математична статистика	- Широкий спектр бібліотек
	- Висока швидкодія	- Простий та зрозумілий синтаксис
	Мінуси:	Мінуси:
	- Обмежений екосистемою	- Можливі проблеми з швидкодією

F2	Плюси:	Плюси:
	- Широкий вибір функцій та методів	- Багато готових бібліотек
	- Спеціалізований для аналізу даних	- Більша гнучкість в розробці
	Мінуси:	Мінуси:
	- Обмежений у використанні поза аналізом даних	- Більше коду для розробки з нуля
F3	Плюси:	Плюси:
	- Використання перевірених методик	- Гнучкість у тестуванні та оптимізації
	- Детальний аналіз недоліків моделі	- Можливість покращення точності
	Мінуси:	Мінуси:
	- Обмеженість вибору методик	- Затрати часу на тестування та оптимізацію

Отже, у функції F1 - Вибір мови програмування для розробки, варіант А (мова програмування R) має перевагу у використанні математичної статистики та високій швидкодії. Варіант Б (мова програмування Python), з свого боку, вирізняється широким спектром бібліотек та простим синтаксисом. Варіант А може мати обмеження у використанні поза аналізом даних, тоді як варіант Б може стикатися з проблемами швидкодії.

У функції F2 - Аналіз даних та розробка моделі, варіант А має перевагу у широкому виборі функцій та методів, а також спеціалізації на аналізі даних. З іншого боку, варіант Б надає можливість використання багатьох готових бібліотек і має більшу гнучкість у розробці. Зазначено, що варіант А може бути обмежений у

використанні поза аналізом даних, тоді як для варіанта Б може знадобитися більше коду для розробки з нуля.

У функції F3 - Тестування та оптимізація моделі обидва варіанти мають свої переваги. Вони дозволяють використовувати перевірені методики та забезпечують гнучкість у тестуванні та оптимізації моделі. Варіант А відзначається детальним аналізом недоліків моделі, тоді як варіант Б може вимагати більше часу на тестування та оптимізацію.

Після проведення ретельного аналізу можна зробити наступні висновки щодо вибору варіантів реалізації:

1. F1: Варіант Б є більш оптимальним варіантом для використання через наявну велику кількість бібліотек та простоту у використанні. Також слід врахувати високий рівень інтегрованості Python.
2. F2: Аналіз даних та розробка моделі, варіант А та варіант Б можуть бути використані обидва, залежно від конкретних потреб та вимог проекту. Обидва варіанти мають свої переваги, які можуть бути використані в різних ситуаціях.
3. F3: З метою спрощення розробки та оптимального використання ресурсів, варто вибрати варіант, що використовує вже існуючі методи оцінки точності та стандартне налаштування гіперпараметрів. Таким чином, пропонується обрати Варіант А, який є найоптимальнішим в даному контексті.

У результаті оцінки можна розглянути наступні варіанти реалізації програмного забезпечення: Варіант А для функцій F1 і F3. Варіант А виявився оптимальним в двох із цих випадків і відповідає потребам та обмеженням проекту. Остаточний вибір варіанту реалізації буде залежати від специфіки проекту та вимог замовника, тому для F2 було створено два сценарії:

$$1) F1(A) - F2(A) - F3(A)$$

$$2) F1(A) - F2(B) - F3(A)$$

5.3 Вибір параметрів

На підставі проведеного попереднього аналізу було визначено, що для оцінки програмного продукту необхідно враховувати наступні кількісні параметри:

1. X1 - Час виконання програми, вимірюваний у мілісекундах. Цей показник відображає швидкодію програмного продукту та його ефективність при виконанні обчислень.
2. X2 - Обсяг використаної оперативної пам'яті, вимірюваний у мегабайтах. Цей показник визначає ресурсомісткість програмного продукту та його вимоги до пам'яті під час виконання.
3. X3 - Час, необхідний для навчання моделі, враховуючи складність алгоритмів та обсяг даних, які потрібно обробити. Цей параметр оцінює швидкість процесу навчання моделі.
4. X4 - Обсяг програмного продукту, вимірюваний у гігабайтах. Цей показник визначає розмір програмного коду, бази даних та інших компонентів продукту, що впливають на його потужність та займане місце.

Значення цих параметрів будуть вимірюватися кількісно з урахуванням вимог замовника та умов експлуатації програмного продукту. Кожен параметр буде оцінюватися за числовою шкалою, що дозволить здійснити об'єктивну оцінку. Отримані значення будуть враховуватися при розрахунку коефіцієнта технічного рівня програмного продукту. Детальні результати оцінки параметрів представлені у таблиці 5.2, де наведені числові значення для кожного параметра згідно вимог замовника та умов експлуатації.

Таблиця 5.2 - Параметри продукту

Параметр	Позначення	Одиниці виміру	Значення		
			Погано	Посередньо	Добре
Час виконання	X1	Операція/ мс	0.5	2	20
Обсяг оперативної пам'яті	X2	Мб	8000	6500	450
Час для навчання	X3	Год	1	0.4	0.2
Очікуваний обсяг продукту	X4	ГБ	1000	150	30

На основі таблиці вище, було побудовано графіки для кожного параметра. (Рис. 5.2)

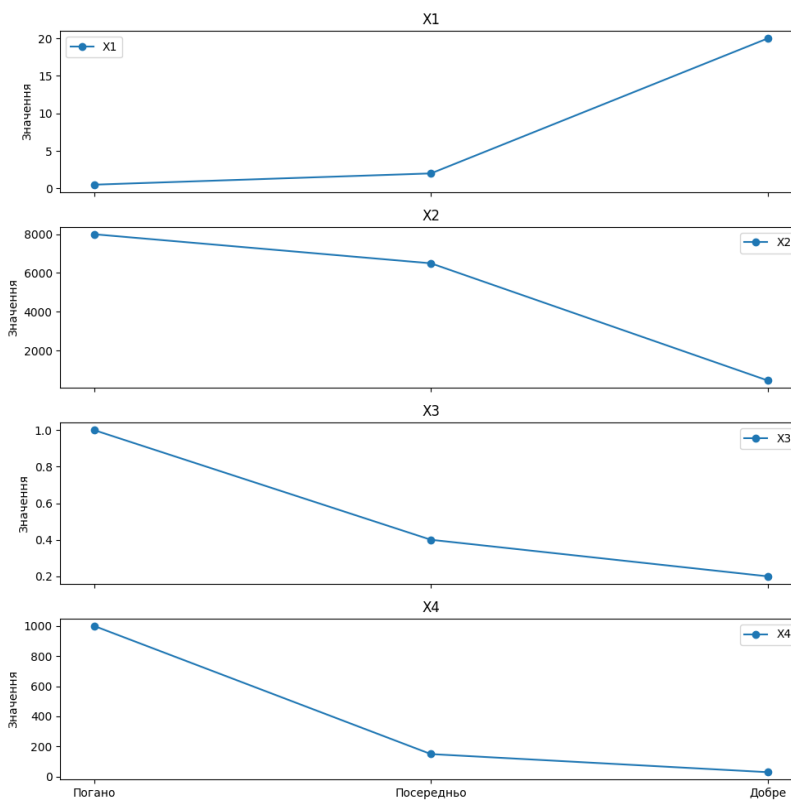


Рисунок 5.2 - Візуалізація для кожного доступного параметра

5.4 Експертний аналіз

З метою досягнення найточніших результатів у моделюванні адаптивного прогнозування та обчисленні прогнозних значень, було проведено інтенсивне обговорення та аналіз, на основі яких кожен експерт надав свою безапеляційну оцінку ступеня важливості кожного параметра. Цей процес виконувався за допомогою методу попарного порівняння, залучаючи 7 висококваліфікованих фахівців у складі експертної комісії.

Початковим етапом процесу було встановлення вищого порядку важливості параметрів шляхом їх належного ранжування. Далі, був проведений аналіз адекватності експертних оцінок з метою подальшого практичного використання отриманих даних. Попарне порівняння параметрів було здійснено для визначення їх пріоритетності, а отримані результати були піддані аналізу та обробці для визначення коефіцієнтів значимості кожного параметра.

Цей методологічний підхід дозволяє систематично оцінити та врахувати вагомість кожного параметра у процесі розробки програмного продукту. Він надає непереконливу інформацію, яка є цінною для прийняття стратегічних рішень та налаштування адаптивних прогнозуючих моделей. Результати експертної оцінки параметрів, отримані від провідних фахівців, детально фіксуються у таблиці 5.3, надаючи цінний посібник для подальшої роботи.

Процес встановлення вищого порядку важливості параметрів через їх ранжування відображає наше прагнення до ясності та чіткості у визначенні пріоритетів. Аналіз адекватності експертних оцінок підтверджує нашу прозорість і відповідальність у використанні отриманих даних для подальшої роботи над програмним продуктом.

Попарне порівняння параметрів, яке було здійснено, сприяє виявленню їхньої пріоритетності та забезпечує нам цінну інформацію для прийняття стратегічних рішень. Аналіз та обробка отриманих результатів у підсумку дозволяють нам

визначити коефіцієнти значимості кожного параметра, що стане основою для подальшої роботи над розробкою програмного продукту.

Результати експертної оцінки параметрів, які отримані від провідних фахівців, внесуть вагому вклад у нашу роботу та нададуть нам цінний посібник для подальшого вдосконалення програмного продукту. Завдяки цим результатам ми зможемо зосередитися на найважливіших параметрах, що дозволить нам досягти найкращих результатів у моделюванні та прогнозуванні.

Таблиця 5.3 - Результати оцінки параметрів від експертів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Швидкодія	Операція/ мс	2	2	1	1	2	2	1	11	-6,5	42,25
X2	Обсяг оперативної пам'яті	Мб	4	4	4	4	3	4	3	26	8,5	72,25
X3	Час для навчання	Год	3	3	3	3	4	3	4	23	5,5	30,25
X4	Очікуваний обсяг продукту	ГБ	1	1	2	2	1	1	2	10	-7,5	56,25
	Сума		10	10	10	10	10	10	10	70	0	201

Для перевірки достовірності оцінок буде визначено наступні параметри:

1. Обчислено суму рангів кожного з параметрів та загальну суму рангів (Формула 5.1):

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, (5.1)$$

де N - кількість експертів,
 n - кількість параметрів.

2. Обчислено середню суму рангів за формулою (5.2):

$$T = \frac{1}{n} R_{ij} = 17,5 \quad (5.2)$$

3. Обчислено відхилення суми рангів кожного параметра від середньої суми рангів за формулою (5.3):

$$\Delta_i = R_i - T \quad (5.3)$$

4. Обчислено загальну суму квадратів відхилень за формулою (5.4):

$$S = \sum_{i=1}^N \Delta_i^2 = 201. \quad (5.4)$$

Обраховано коефіцієнт узгодженості за формулою (5.5):

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 201}{7^2(4^3 - 4)} = 0,82 > W_k = 0,67. \quad (5.5)$$

Отриманий коефіцієнт узгодженості, який перевищує встановлене нормативне значення 0.67, свідчить про достовірність проведеного ранжування параметрів. Це дозволяє нам мати впевненість у відповідності наших експертних оцінок та встановленого порядку важливості.

За допомогою отриманих результатів ранжування, ми здійснимо попарне порівняння всіх параметрів. Цей процес дозволить нам більш детально проаналізувати їхню пріоритетність та занести отримані результати до таблиці 5.4. Ця

таблиця стане цінним інструментом для подальшого використання та прийняття рішень у процесі розробки програмного продукту.

Отримані результати ранжування та попарного порівняння параметрів дозволять нам більш об'єктивно визначити їх значущість і встановити пріоритети в розробці. Це важливий крок, який допоможе нам ефективно використовувати ресурси та налаштовувати адаптивні прогнозуючі моделі.

Таблиця 5.4 - Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	<	<	<	<	<	0,5
X1 і X3	<	>	>	<	<	<	<	<	0,5
X1 і X4	>	>	<	<	>	>	<	>	1,5
X2 і X3	>	>	>	>	<	>	<	>	1,5
X2 і X4	>	>	>	>	>	>	>	>	1,5
X3 и X4	>	>	>	>	>	>	>	>	1,5

За метою точного відтворення відносної переваги між параметрами, числове значення a_{ij} обчислюється згідно формули (5.6). Ця формула враховує ступінь наділення переваги одного параметра в порівнянні з іншим.

$$a_{ij} = \{1.5 \text{ при } X_i > X_j \ 1.0 \text{ при } X_i = X_j \ 0.5 \text{ при } X_i < X_j \}. \quad (5.6)$$

З отриманих числових оцінок утворюється матриця $A = \|a_{ij}\|$

Для кожного з параметрів розраховується вагомість K_{bi} згідно з формулами:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (5.7)$$

$$b_i = \sum_{j=1}^N a_{ij} \quad (5.8)$$

Продовжуючи ітераційний процес, відносні оцінки розраховуються знову і знову до того моменту, коли наступні значення відрізняються від попередніх менше

ніж на 2%. На кожному наступному етапі розрахунку відносні оцінки обчислюються згідно з формулами, які наведені нижче (5.9-5.10):

$$K_{Bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (5.9)$$

$$b'_i = \sum_{j=1}^N a_{ij} b_j \quad (5.10)$$

З аналізу таблиці 5.5 очевидно, що відмінність між значеннями коефіцієнтів вагомості не перевищує 2%, що свідчить про достатню точність отриманих результатів. Отже, додаткові ітерації не є необхідними.

Таблиця 5.5 - Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер		Четверта ітер	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2	b_i^3	K_{Bi}^3
X1	1	0,5	0,5	1,5	3,5	0,21	12,25	0,18	42,875	0,14	150,0625	0,12
X2	1,5	1	1,5	1,5	5,5	0,34	30,25	0,44	166,375	0,53	915,0625	0,55
X3	1,5	0,5	1	1,5	4,5	0,28	20,25	0,3	91,125	0,29	410,0625	0,270
X4	0,5	0,5	0,5	1	2,5	0,16	6,25	0,09	15,625	0,05	39,0625	0,026
Всього:					16	1	69	1	316	1	1514,25	1

5.5 Аналіз якості реалізації варіантів функцій

З метою оцінки якості виконання основних функцій програмного продукту, ми досліджуємо абсолютні значення параметрів, таких як об'єм пам'яті (X2), час для навчання (X3) та потенційний об'єм програмного коду (X4), які відповідають вимогам

технічних умов його функціонування. Особлива увага приділяється також швидкодії мови програмування (X1), яка повинна мати задовільний рівень продуктивності.

Для кожного варіанта реалізації програмного продукту ми обчислюємо показники якості, використовуючи коефіцієнт технічного рівня (КК_j), який враховує вагомність кожного параметра (Формула 5.11). Цей коефіцієнт дозволяє нам оцінити вплив кожного параметра на загальну якість програмного продукту. Чим вище значення коефіцієнта технічного рівня, тим краща якість виконання функцій програмного продукту.

$$K_K(j) = \sum_{i=1}^n K_{Vi,j} B_{i,j}, \quad (5.11)$$

де n – кількість параметрів;

K_{Vi} – коефіцієнт вагомності i -го параметра;

B_i – оцінка i -го параметра в балах.

Кожен показник рівня якості обчислюється за допомогою формули (5.11), яка враховує вагомність параметра (K_{Vi}) та його оцінку (B_i) у балах. Ці обчислення представлені в таблиці 5.6, яка надає оцінку якості різних варіантів реалізації основних функцій програмного продукту. Значення показників рівня якості вказують на те, наскільки задовольняється кожен варіант вимогам та враховується вагомність його параметрів.

Таблиця 5.6 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Осно вні функції	Варіа нт реалізац ії функції	Параме три	Абсолю тне значення параметра	Баль на оцінка парамет ра	Коефіці єнт вагомості параметра	Коефіці єнт рівня якості
F2	A	X3	56	23	0,27	6,21
	B	X2	70	26	0,55	14,3
F1	A	X1	30	11	0.12	1,32
F3	A	X4	30	10	0,026	0,26

$$K_K = K_{Ty}[F_{1k}] + K_{Ty}[F_{2k}] + \dots + K_{Ty}[F_{zk}], \quad (5.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 6,21 + 1,32 + 0,26 = 7,79 ;$$

$$K_{K2} = 14,3 + 1,32 + 0,26 = 15,88.$$

Отже, кращим в даному випадку є варіант 2, оскільки він має вищий рівень якості.

5.6 Економічний аналіз розробки

Для оцінки витрат на розробку програмного продукту, першим кроком є розрахунок обсягу роботи для кожного варіанту виконання основних завдань, який включає розробку проекту програмного продукту і програмної оболонки. Зробимо

порівняння між функцією 1 та функцією 2, де є відмінність у рівні мови програмування. Оцінка загального обсягу роботи буде проведена за допомогою формули (5.13).

$$T_0 = T_P \cdot K_P \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (5.13),$$

де T_P – трудомісткість розробки ПП;

K_P – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для даного конкретного завдання, яке відноситься до категорії розрахункового характеру і має ступінь новизни А, був проведений розрахунок трудомісткості, що складає 30 людино-днів. В процесі розрахунків були використані поправочні коефіцієнти: $K_P = 1.6$, який враховує вид нормативно-довідкової інформації, та $K_{СК} = 1.3$, який враховує складність контролю вхідних і вихідних даних. Ці коефіцієнти дозволили отримати загальну трудомісткість програмування для даного завдання.

Під час розрахунків було також враховано використання стандартних модулів, що забезпечило зниження трудомісткості завдання і було враховано за допомогою коефіцієнта $K_{СТ} = 0.9$. Загальний коефіцієнт $K_M = 1.0$ використовувався для визначення загальної трудомісткості.

$$T_1 = 30 \cdot 1.6 \cdot 1.3 \cdot 0.9 \cdot 1 = 56,16 \text{ людино-днів, або } 1347,84 \text{ людино-годин}$$

Для другого завдання було проведено аналогічні обрахунки із використанням $K_M=1.2$ та $K_{СТ}=0.8$, $K_P=1.6$:

$$T_2 = 30 \cdot 1.6 \cdot 1.3 \cdot 0.8 \cdot 1.2 = 59,904 \text{ людино-днів, або } 1437,696 \text{ людино-годин}$$

Як видно із обрахунків, найбільшу трудомісткість має функція під номером 2.

В розробці буде брати участь один програміст з окладом 35 000 та один спеціаліст із роботи з даними із окладом 45 000.

Середня зарплата буде обрахована за такою формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (5.14)$$

де M – місячний оклад працівників;

T_m – кількість робочих днів місяць;

t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{35000 + 45000}{2 \cdot 21 \cdot 8} = 238,1 \text{ грн.} \quad (5.15)$$

Далі обраховується середня зарплата за формулою (5.16)

$$C_{\text{ЗП}} = C_{\text{ч}} \cdot T_i \cdot K_d, \quad (5.16)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

K_d – норматив, який враховує додаткову заробітну плату (20%).

Зарплата робітників із соц. внеском, що становить 22% дорівнює:

$$1. C_{\text{ЗП}} = 238,1 \cdot 1347,84 \cdot 1,22 = 391\,523,3 \text{ грн.}$$

$$2. C_{\text{ЗП}} = 238,1 \cdot 1437,696 \cdot 1,22 = 417\,624,8 \text{ грн.}$$

Далі буде обраховано ціну використання ЕОМ. (C_M)

Так як одна ЕОМ обслуговує одного програміста з окладом 35 000 грн., при коефіцієнті зайнятості 0,2, отримаємо наступні розрахунки для однієї машини:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 35\,000 \cdot 0,2 = 84\,000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{ЗП} = C_{Г} \cdot (1 + K_{З}) = 84000 \cdot (1 + 0.2) = 100800 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 30 000 грн.

$$C_{А} = K_{ТМ} \cdot K_{А} \cdot Ц_{ПР} = 1,2 \cdot 0,25 \cdot 30\ 000 = 90000 \text{ грн.},$$

де $K_{ТМ}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$K_{А}$ – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_{Р} = K_{ТМ} \cdot Ц_{ПР} \cdot K_{Р} = 1.2 \cdot 30000 \cdot 0.08 = 2880 \text{ грн.},$$

де $K_{Р}$ – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{ЕФ} &= (D_{К} - D_{В} - D_{С} - D_{Р}) \cdot t_{з} \cdot K_{В} = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.35 = \\ &= 627,2 \text{ години,} \end{aligned}$$

де $D_{К}$ – календарна кількість днів у році;

$D_{В}$, $D_{С}$ – відповідно кількість вихідних та святкових днів;

$D_{Р}$ – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_{\text{З}} \cdot C_{\text{ЕН}} = 627,2 \cdot 0,3 \cdot 0,2 \cdot 4,86889 = 183,226 \text{ грн.},$$

де $N_{\text{С}}$ – середньо-споживана потужність приладу;

$K_{\text{З}}$ – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = C_{\text{ПР}} \cdot 0,67 = 30000 \cdot 0,67 = 20100 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}, \quad (5.17)$$

$$C_{\text{ЕКС}} = 90000 + 2880 + 183,226 + 20100 = 113\,163,226 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 113\,163,226 / 627,2 = 180,42 \text{ грн/год.}$$

У даному контексті, оскільки всі процеси, пов'язані з розробкою програмного продукту, виконуються на ЕОМ (електронно-обчислювальних машинах), вартість машинного часу буде впливати на витрати, і ця вартість залежатиме від обраного варіанту реалізації.

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T, \quad (5.18)$$

$$1. C_{\text{М}} = 180,42 \cdot 1347,84 = 243\,177,3 \text{ грн.}$$

$$2. C_{\text{М}} = 180,42 \cdot 1437,696 = 259\,389,11 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{ЗП} \cdot 0,67, (5.19)$$

1. $C_H = 391\,523,3 \cdot 0,67 = 262\,320,611$ грн.
2. $C_H = 417\,624,8 \cdot 0,67 = 279\,808,616$ грн.

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{ЗП} + C_M + C_H, \quad (5.20)$$

1. $C_{ПП} = 391\,523,3 + 243\,177,3 + 262\,320,611 = 597\,021,211$ грн.
2. $C_{ПП} = 417\,624,8 + 259\,389,11 + 279\,808,616 = 956\,822,526$ грн.

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

В цьому розділі ми здійснимо розрахунок та вибір оптимального варіанту розробки програмного продукту згідно з формулою (5.21).

Застосовуючи формулу (5.21), ми розраховуємо коефіцієнт техніко-економічного рівня ($K_{ТЕРj}$) для кожного варіанту:

$$K_{ТЕРj} = K_{Kj} / C_{Фj}, (5.21)$$

$$K_{ТЕР1} = 7,79 / 597\,021,21 = 0,13 \cdot 10^{-4},$$

$$K_{ТЕР2} = 15,88 / 956\,822,526 = 0,16 \cdot 10^{-4}.$$

Таким чином, з обрахунків видно, що кращим варіантом є другий, який має коефіцієнт рівний $0.16 \cdot 10^{-4}$.

Після проведення функціонально-вартісного аналізу ми приходимо до висновку, що серед залишених варіантів найкращим є другий. Він має найкращий показник коефіцієнта якості.

Обраний варіант включає такі параметри:

1. Використання мови програмування Python.
2. Створення власних моделей для класифікації.
3. Використання готових метрик для оцінки точності.

Цей варіант реалізації забезпечує швидку та надійну розробку системи і не вимагатиме значних витрат ресурсів компанії.

5.8 Висновки до розділу

В даному розділі ми провели детальний функціонально-вартісний аналіз програмного продукту для розробки системи прогнозування стійкості фінансових показників. Аналіз включав визначення основних функцій, оцінку їх вагомості та визначення параметрів вартості розробки.

На основі проведеного аналізу було обрано найоптимальніший варіант реалізації програмного продукту, який відповідає поставленим цілям та вимогам. Обраний варіант використовує мову програмування Python для розробки і використовує готові бібліотеки та функції для аналізу даних та візуалізації, що забезпечує швидкість та ефективність розробки.

Здійснений аналіз дозволив зрозуміти, які функції є найбільш важливими для розробки програмного продукту і які параметри впливають на вартість розробки. Це

допомогло прийняти обґрунтовані рішення та забезпечити ефективну розробку системи прогнозування стійкості фінансових показників.

В цілому, проведений функціонально-вартісний аналіз був важливим кроком у процесі розробки програмного продукту, що дозволив визначити оптимальні варіанти реалізації та встановити основні параметри для успішної реалізації системи прогнозування стійкості фінансових показників.

ВИСНОВКИ

Під час роботи над прогнозуванням криптовалютних курсів досліджено різні методи, проаналізовано особливості часових рядів та проблеми, пов'язані з прогнозуванням курсів криптовалют.

З'ясувано, що часові ряди мають унікальні властивості, які роблять їх привабливими для прогнозування, такі як часова впорядкованість, закономірності та тенденції. Прогнозування фінансових часових рядів має свої особливості, пов'язані з високим ступенем невизначеності та ризику на фінансових ринках.

У розділі прогнозування курсу криптовалют розглянуто різні методи прогнозування, включаючи гібридні методи та методи машинного навчання. Виявилось, що метод Prophet показав найкращі результати у прогнозуванні криптовалютних курсів, забезпечуючи високу точність та ефективність.

Також був проведений функціонально-вартісний аналіз, що дозволив зробити вибір найкращого методу для реалізації моделі та оцінити вартість розробки.

Аналізуючи результати нашої роботи, ми прийшли до висновку, що прогнозування криптовалютних курсів є важливим інструментом для інвесторів та трейдерів. Використання відповідних методів, таких як Prophet та Auto ARIMA, може допомогти приймати обґрунтовані рішення та покращувати результати.

Дослідження також виявило проблеми, пов'язані з волатильністю ринку криптовалют, обмеженою історичною інформацією та проблемами зі збором та обробкою даних. Однак, незважаючи на ці виклики, прогнозування криптовалютних курсів має потенціал приносити значну користь, якщо використовується належним чином.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Definition of Time Series Analysis. *Toppr*. Режим доступу: <https://www.toppr.com/guides/fundamentals-of-business-mathematics-and-statistics/time-series-analysis/definition-of-time-series-analysis/> (Дата звернення 16.05.2023)
2. Marco Peixeiro. The Complete Guide to Time Series Analysis and Forecasting. Aug 7, 2019. Режим доступу: <https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775> (Дата звернення 16.05.2023)
3. Leo Smigel. What Is Stationarity? A Visual Guide. April 27, 2022. Режим доступу: <https://analyzingalpha.com/stationarity> (Дата звернення 16.05.2023)
4. Chunhang Chen. Robustness properties of some forecasting methods for seasonal time series: A Monte Carlo study. June 1997. Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S0169207097000149> (Дата звернення 16.05.2023)
5. Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and Practice - Chapter 4 Time series features. Режим доступу: <https://otexts.com/fpp3/> (Дата звернення 16.05.2023)
6. Francesca Lazzeri. Introduction to feature engineering for time series forecasting. Oct 5, 2021. Режим доступу: <https://medium.com/data-science-at-microsoft/introduction-to-feature-engineering-for-time-series-forecasting-620aa55fcab0> (Дата звернення 16.05.2023)
7. Sequence data vs time series data. Stack Exchange Network. Режим доступу: <https://datascience.stackexchange.com/questions/30182/sequence-data-vs-time-series-data> (Дата звернення 16.05.2023)
8. What are the biggest differences between time series and non-time series regression? Quora. Режим доступу: <https://www.quora.com/What-are-the-biggest-differences-between-time-series-and-non-time-series-regression> (Дата звернення 16.05.2023)

9. Lithmee Mandula. Difference Between Time Series and Cross Sectional Data. June 11, 2018. Режим доступу: <https://www.differencebetween.com/difference-between-time-series-and-cross-sectional-data/> (Дата звернення 16.05.2023)
10. M. Mallikarjuna & R. Prabhakara Rao. Evaluation of forecasting methods from selected stock market returns. Режим доступу: <https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0157-x> (Дата звернення 16.05.2023)
11. Will Kenton. What Is Risk Management in Finance, and Why Is It Important? March 01, 2021. Режим доступу: <https://www.investopedia.com/terms/r/riskmanagement.asp> (Дата звернення 16.05.2023)
12. Why is financial forecasting so critical to your business? *Business Growth Services*. Режим доступу: <https://www.businessgrowthservices.co/why-is-financial-forecasting-so-critical-for-your-business/> (Дата звернення 16.05.2023)
13. 9 Key Features of Financial Management System by SolutionDot. Saudi ERP & Website Solution Blog. November 29, 2016. Режим доступу: <https://solutiondots.com/blog/9-key-features-of-financial-management-system/> (Дата звернення 16.05.2023)
14. Jason Brownlee. 10 Challenging Machine Learning Time Series Forecasting Problems. March 1, 2017. Режим доступу: <https://machinelearningmastery.com/challenging-machine-learning-time-series-forecasting-problems/> (Дата звернення 16.05.2023)
15. Forecasting Financial Time Series - Part I. Quantstart. January 2023. Режим доступу: <https://www.quantstart.com/articles/Forecasting-Financial-Time-Series-Part-1/> (Дата звернення 16.05.2023)
16. Jon Farland. Navigating the challenges of time series forecasting. April 12th, 2023. Режим доступу: <https://h2o.ai/blog/navigating-the-challenges-of-time-series-forecasting/> (Дата звернення 16.05.2023)

17. Bitcoin Price Prediction 2023-2030 BTC Forecast [Research]. Software Testing Help. April 26, 2023. Режим доступу: <https://www.softwaretestinghelp.com/> (Дата звернення 16.05.2023)
18. Viktor Manahov, Hanxiong Zhang. Forecasting Cryptocurrency Markets using Artificial Intelligence and Machine Learning Tools. 06 October 2022. Режим доступу: <https://www.frontiersin.org/research-topics/37594/forecasting-cryptocurrency-markets-using-artificial-intelligence-and-machine-learning-tools> (Дата звернення 16.05.2023)
19. Predictions News. Cointelegraph. Режим доступу: <https://cointelegraph.com/tags/predictions> (Дата звернення 16.05.2023)
20. Helder Sebastião & Pedro Godinho. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. 06 January 2021. Режим доступу: <https://jfin-swufe.springeropen.com/articles/10.1186/s40854-020-00217-x> (Дата звернення 16.05.2023)
21. Vasily Derbentsev, Olga Stepanenko, Natalia Datsenko, Vitaly Bezkorovainyi. Forecasting cryptocurrency prices time series using machine learning approach. January 2019. Режим доступу: https://www.researchgate.net/publication/333464180_Forecasting_cryptocurrency_prices_time_series_using_machine_learning_approach (Дата звернення 16.05.2023)
22. What is fundamental analysis of cryptocurrency markets? Bit2me Academy. October 11, 2019. Режим доступу: <https://academy.bit2me.com/en/what-is-cryptocurrency-fundamental-analysis/> (Дата звернення 16.05.2023)
23. Mohadesa Najumi. How to predict cryptocurrency price using fundamental analysis. 22 December 2021. Режим доступу: <https://capital.com/how-to-predict-cryptocurrency-prices> (Дата звернення 16.05.2023)
24. How to perform crypto fundamental analysis. SwissBorg Academy. March 23, 2021. Режим доступу: <https://academy.swissborg.com/en/learn/crypto-fundamental-analysis> (Дата звернення 16.05.2023)

25. Temitope Olatunji. What Is Crypto Technical Analysis? 6 Technical Indicators Explained. Jun 10, 2022. Режим доступу: <https://www.makeuseof.com/crypto-technical-analysis-explained/> (Дата звернення 16.05.2023)
26. Methods You Can Use to Analyze Crypto Markets. Paxful University. Режим доступу: <https://paxful.com/university/cryptocurrency-market-analysis/> (Дата звернення 16.05.2023)
27. Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Anticipating Cryptocurrency Prices Using Machine Learning. 04 Nov 2018. Режим доступу: <https://www.hindawi.com/journals/complexity/2018/8983590/> (Дата звернення 16.05.2023)
28. Helder Sebastião & Pedro Godinho. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. 06 January 2021. Режим доступу: <https://jfin-swufe.springeropen.com/articles/10.1186/s40854-020-00217-x> (Дата звернення 16.05.2023)
29. Jesus Guzman. Top 5 Most Volatile Cryptocurrencies Today. November 3, 2021. Режим доступу: <https://www.interactivecrypto.com/discover-the-most-volatile-cryptos-of-all> (Дата звернення 16.05.2023)
30. Why Is The Market So Unpredictable? Let's Talk Volatility. Stash. Oct 19, 2018. Режим доступу: <https://www.stash.com/learn/volatility-market-unpredictable/> (Дата звернення 16.05.2023)
31. Today's Cryptocurrency Prices by Market Cap. CoinMarketCap. Режим доступу: <https://coinmarketcap.com/> (Дата звернення 16.05.2023)
32. Hamza Ahmad. Forecasting Future Prices of Cryptocurrency using Historical Data. Dec 5, 2019. Режим доступу: <https://towardsdatascience.com/forecasting-future-prices-of-cryptocurrency-using-historical-data-83604e72bc68> (Дата звернення 16.05.2023)
33. Chalita Lertlumprasert. CryptoCurrency Price Prediction with Python. Jan 6, 2018. Режим доступу: <https://towardsdatascience.com/demystifying-cryptocurrency-price-prediction-5fb2b504a110> (Дата звернення 16.05.2023)

34. Paul Tune. Time Series Modeling of Bitcoin Prices. Feb 21, 2021. Режим доступа: <https://towardsdatascience.com/time-series-modeling-of-bitcoin-prices-5133edfec30b> (Дата звернення 16.05.2023)
35. Kale-ab Tessera. IcoOmen: Using Machine Learning to Predict ICO Prices. Dec 13, 2018. Режим доступа: <https://towardsdatascience.com/icoomen-using-machine-learning-to-predict-ico-prices-29fa4cec6d86> (Дата звернення 16.05.2023)
36. Ioannis E. Livieris, Niki Kiriakidou, Stavros Stavroyiannis and Panagiotis Pintelas. An Advanced CNN-LSTM Model for Cryptocurrency Forecasting. 26 January 2021. Режим доступа: <https://www.mdpi.com/2079-9292/10/3/287/pdf> (Дата звернення 16.05.2023)
37. Justin Кюерпер. How Cryptocurrencies Affect the Global Market. October 29, 2021. Режим доступа: <https://www.thebalancemoney.com/how-cryptocurrencies-affect-the-global-market-4161278> (Дата звернення 16.05.2023)
38. Kelsie Nabben. Cryptocurrency has an impact on economies. That’s why some are afraid of it – and some welcome it. January 31, 2022. Режим доступа: <https://theconversation.com/cryptocurrency-has-an-impact-on-economies-thats-why-some-are-afraid-of-it-and-some-welcome-it-175911> (Дата звернення 16.05.2023)
39. Connor Sephton. What do interest rate hikes mean for crypto? 16 June 2022. Режим доступа: <https://currency.com/interest-rates-forecast> (Дата звернення 16.05.2023)
40. Alex Miles. Best 11 crypto trading bots for 2023. April 28, 2023. Режим доступа: <https://tokentax.co/blog/best-crypto-trading-bot> (Дата звернення 16.05.2023)
41. Best 10 Crypto Trading Bots For 2023 [Review And Comparison]. Software Testing Help. May 10, 2023. Режим доступа: <https://www.softwaretestinghelp.com/best-crypto-trading-bots/> (Дата звернення 16.05.2023)
42. Moritz Putzhammer. Crypto Trading Bots: The Ultimate Beginner's Guide. 28 June 2022. Режим доступа: <https://www.trality.com/blog/crypto-trading-bots> (Дата звернення 16.05.2023)

43. Jayit Saha, Smit Patel, Frank Xing, Erik Cambria. Does Social Media Sentiment Predict Bitcoin Trading Volume? 2022. Режим доступу: <https://sentic.net/social-media-sentiment-and-bitcoin-trading-volume.pdf> (Дата звернення 16.05.2023)
44. Prophet. Facebook GitHub. Режим доступу: <https://facebook.github.io/prophet/> (Дата звернення 18.05.2023)
45. Nir Barazida. How to use DagsHub with PyCaret. Jan 16, 2023. Режим доступу: <https://dagshub.com/blog/how-to-use-pycaret-with-dagshub/> (Дата звернення 18.05.2023)
46. Jason Brownlee. Time Series Forecasting With Prophet in Python. August 26, 2020. Режим доступу: <https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/> (Дата звернення 18.05.2023)
47. Sophia Yang. Time series analysis using Prophet in Python — Part 2: Hyperparameter Tuning and Cross Validation. Jul 9, 2020. Режим доступу: <https://medium.com/analytics-vidhya/time-series-analysis-using-prophet-in-python-part-2-hyperparameter-tuning-and-cross-validation-88e7d831a067> (Дата звернення 18.05.2023)
48. How to build ARIMA models in Python for time series prediction. Just Into Data. August 25, 2022. Режим доступу: <https://www.justintodata.com/arima-models-in-python-time-series-prediction/> (Дата звернення 18.05.2023)
49. An Introduction to Exponential Smoothing for Time Series Forecasting in Python. SimpliLearn. Feb 15, 2023. Режим доступу: <https://www.simplilearn.com/exponential-smoothing-for-time-series-forecasting-in-python-article> (Дата звернення 18.05.2023)
50. Exponential Smoothing: Definition of Simple, Double and Triple. Statistics How To. Режим доступу: <https://www.statisticshowto.com/exponential-smoothing/> (Дата звернення 18.05.2023)
51. Jason Brownlee. How to Develop LSTM Models for Time Series Forecasting. November 14, 2018. Режим доступу: <https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/> (Дата звернення 18.05.2023)

52. Ogulcan Ertunc. 3- Time Series Forecasting Using LSTM. Jun 20, 2021. Режим доступу: <https://medium.com/analytics-vidhya/3-time-series-forecasting-using-lstm-e14b93f4ec7c> (Дата звернення 18.05.2023)

ДОДАТОК А ЛІСТИНГ

```

#!/usr/bin/env python
# coding: utf-8

# In[ ]:

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.metrics import MeanSquaredError
from sklearn.preprocessing import MinMaxScaler
import warnings
import logging
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from keras.utils import plot_model
from statsmodels.tsa.arima.model import ARIMA
import math
from fbprophet import Prophet
from prophet import Prophet

# In[ ]:

for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# In[ ]:

# Завантаження даних з таблиці у об'єкт DataFrame
df = pd.read_csv('шлях_до_файлу.csv')

# Перевірка та зміна формату дати
df['Дата'] = pd.to_datetime(df['Дата'])

# Вибір необхідних стовпців
df = df[['Дата', 'Значення']]

# Перейменування стовпців на "ds" (для дати) та "y" (для значення)
df = df.rename(columns={'Дата': 'ds', 'Значення': 'y'})

# Опціонально: обрізання аномалій або видалення пропущених значень
df = df.loc[df['y'].notnull()]

# Ініціалізація моделі Prophet та її навчання на даних
model = Prophet()
model.fit(df)

# Генерація майбутніх дат для прогнозу
future_dates = model.make_future_dataframe(periods=365)

```

```

# Виклик методу прогнозу для отримання результатів
forecast = model.predict(future_dates)

# In[ ]:

# Disable chained assignment warning
pd.options.mode.chained_assignment = None

# Set maximum number of columns to display
pd.options.display.max_columns = 999

# Define root directory for cryptocurrency price history data
RootDir = "/kaggle/input/cryptocurrencypricehistory"

# Specify the number of historical data points to consider
History = 60

# Function to read cryptocurrency data
def read_data():
    coin_no = 0 # Counter for the number of coins

    # Count the number of files in the root directory
    for name in os.listdir(RootDir):
        coin_no += 1

    max_length, min_length = 0, 1000000 # Initialize variables for maximum and
minimum lengths

    # Iterate through the files in the root directory
    for name in os.listdir(RootDir):
        df = pd.read_csv(RootDir + "/" + name, parse_dates=['Date']) # Read
the CSV file into a DataFrame
        length = df.shape[0] # Get the number of rows in the DataFrame
        if max_length < length:
            max_length = length # Update maximum length if necessary
        if min_length > length:
            min_length = length # Update minimum length if necessary

        data = np.zeros((coin_no, max_length)) # Create an array to store the data
lengths = np.zeros(coin_no, dtype=int) # Create an array to store the
lengths of each coin's data

        i = 0 # Initialize index variable
        # Iterate through the files in the root directory again
        for name in os.listdir(RootDir):
            short_name = name[5:-4] # Extract the short name of the coin
            df = pd.read_csv(RootDir + "/" + name, parse_dates=['Date']) # Read
the CSV file into a DataFrame
            length = df.shape[0] # Get the number of rows in the DataFrame
            lengths[i] = length # Store the length of the current coin's data
            print(i, short_name, length) # Print the index, short name, and length
of the current coin's data
            data[i, 0:length] = df['Close'].values # Store the closing prices of
the current coin as a sequence
            i += 1 # Increment the index

    return coin_no, lengths, data

# Call the read_data function to retrieve the cryptocurrency data
coin_no, lengths, data = read_data()

```

```

# Print the number of coins obtained
print("Got", coin_no, "coins.")

# In[ ]:

for coin in os.listdir('/kaggle/input/cryptocurrencypricehistory'):
    df = pd.read_csv('/kaggle/input/cryptocurrencypricehistory/' + coin)
    display(df)

# In[ ]:

# Visualize time series data
def visualize_data(data, lengths, coin_index):
    coin_data = data[coin_index] # Select the data for a specific coin
    length = lengths[coin_index] # Get the length of the data for the coin

    # Create a figure and axes
    fig, ax = plt.subplots()

    # Plot the time series data
    ax.plot(range(length), coin_data[:length], label='Coin
{}'.format(coin_index))

    # Set the title and labels
    ax.set_title('Time Series Data')
    ax.set_xlabel('Time')
    ax.set_ylabel('Price')

    # Display the legend
    ax.legend()

    # Show the plot
    plt.show()

# In[ ]:

coin_names = [i.split('_')[1].split('.')[0] for i in
os.listdir('/kaggle/input/cryptocurrencypricehistory')]
print(coin_names)
# Calculate the number of rows and columns for subplots
num_coins = len(coin_names)
num_rows = 2
num_cols = math.ceil(num_coins / num_rows)

# Create subplots for the visualizations
fig, axs = plt.subplots(num_cols, num_rows, figsize=(16, 56))

# Flatten the axs array
axs = axs.flatten()

# Visualize data for each coin
for i in range(num_coins):
    ax = axs[i] # Select the appropriate subplot

    coin_data = data[i]
    length = lengths[i]

```

```

# Plot the time series data
ax.plot(range(length), coin_data[:length], label='Coin {}'.format(i))

# Set the title of the subplot with coin name and length
ax.set_title('{} (Length: {})'.format(coin_names[i], lengths[i]),
fontsize=16)

# Set the x-axis label
ax.set_xlabel('Time')

# Set the y-axis label
ax.set_ylabel('Price')

# Display the legend
ax.legend()

# Hide any extra subplots
for i in range(num_coins, num_rows * num_cols):
    axs[i].axis('off')

# Adjust the spacing between subplots
fig.tight_layout()

# Show the plot
plt.show()

# In[ ]:

# Function to scale the cryptocurrency data
def scale_data(data, lengths):
    coin_no = data.shape[0] # Get the number of coins
    shift = np.zeros(coin_no) # Initialize an array to store the shift values
    factor = np.zeros(coin_no) # Initialize an array to store the scaling
factors

# Iterate through each coin
for i in range(coin_no):
    max_val = data[i, :lengths[i]].max() # Find the maximum value in the
coin's data
    min_val = data[i, :lengths[i]].min() # Find the minimum value in the
coin's data
    shift[i] = min_val # Store the minimum value as the shift
    factor[i] = max_val - min_val # Calculate the scaling factor
    data[i, 0:lengths[i]] = (data[i, 0:lengths[i]] - shift[i]) / factor[i]
# Scale the coin's data

    return shift, factor

# Call the scale_data function to scale the data and retrieve the shift and
factor
shift, factor = scale_data(data, lengths)

# In[ ]:

data.shape

# In[ ]:

```

```

# Function to create input-output sequences for training, validation, and
testing
def create_sequences(data, lengths, start, end):
    x = [] # List to store input sequences
    y = [] # List to store output sequences

    for i in range(start, end): # Iterate over the specified coins
        for j in range(History, lengths[i]): # Iterate over the available data
points for each coin
            x.append(data[i, j-History:j]) # Append the input sequence of
historical data
            y.append(data[i, j]) # Append the corresponding output value

    return np.array(x)[: , :, np.newaxis], np.array(y) # Convert the lists to
numpy arrays and add a new axis

# Create input-output sequences for training
x_train, y_train = create_sequences(data, lengths, 0, 18)
print("Got", y_train.shape[0], "training sequences.")

# Create input-output sequences for validation
x_val, y_val = create_sequences(data, lengths, 18, 22)
print("Got", y_val.shape[0], "validation sequences.")

# Create input-output sequences for testing
x_test, y_test = create_sequences(data, lengths, 22, 23)
print("Got", y_test.shape[0], "test sequences.")

# In[ ]:

def build_lstm ():
    # Build an LSTM model
    model = Sequential()
    model.add(LSTM(128, return_sequences=True, input_shape= (History, 1)))
    model.add(LSTM(64, return_sequences=False))
    model.add(Dense(25))
    model.add(Dense(1))

    # Compile the model
    model.compile(optimizer='adam', loss='mean_squared_error',
metrics=[MeanSquaredError()])
    model.summary()

    return model

model = build_lstm()

# In[ ]:

plot_model(model)

# In[ ]:

hist = model.fit(x_train, y_train, validation_data = (x_val, y_val),
batch_size=32, epochs=5)

```

```

# In[ ]:

def show_stats (hist):
    plt.plot(hist.history['loss'])
    plt.plot(hist.history['val_loss'])
    plt.title("Model loss")
    plt.ylabel("Loss")
    plt.xlabel("Epoch")
    plt.legend(["Training loss", "Validation loss"])
    plt.show()

show_stats (hist)

# In[ ]:

predictions = model.predict(x_test)

# Calculate RMSE
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print("Root mean square error on test data:", rmse)

# Calculate MAE
mae = np.mean(np.abs(predictions - y_test))
print("Mean absolute error on test data:", mae)

# Calculate DW statistic
residuals = predictions - y_test
dw = np.sum(np.diff(residuals)**2) / np.sum(residuals**2)
print("Durbin-Watson statistic:", dw)

# Calculate MAPE
mape = np.mean(np.abs((predictions - y_test) / y_test)) * 100
print("Mean absolute percentage error on test data:", mape)

# Set the figure size
plt.figure(figsize=(10, 6))

plt.plot(predictions * factor[22] + shift[22])
plt.plot(y_test * factor[22] + shift[22])
plt.legend(["Predictions", "Real data"])
plt.show()

# In[ ]:

get_ipython().system('pip install pmdarima')

# In[ ]:

# Define the directory path of the cryptocurrency price data
data_dir = '/kaggle/input/cryptocurrencypricehistory'

# Get the list of filenames in the directory
filenames = os.listdir(data_dir)

```

```

# Create empty lists to store the metrics
rmse_list = []
mae_list = []
dw_list = []
mape_list = []

with warnings.catch_warnings():
    warnings.filterwarnings("ignore")

# Loop through each cryptocurrency dataset
for filename in filenames:
    # Read the CSV file
    df = pd.read_csv(os.path.join(data_dir, filename))

    # Preprocess the dataframe
    scaler = MinMaxScaler()
    df['Close'] = scaler.fit_transform(df['Close'].values.reshape(-1, 1))

    arima_yhat = []
    for i in range(60):
        # Create ARIMA model
        arima_model = ARIMA(df['Close'][:-60+i], order=(3, 0, 1))

        # Fit the ARIMA model
        arima_model_fit = arima_model.fit()

        # Forecast using ARIMA
        arima_forecast = arima_model_fit.forecast(steps=1)
        arima_yhat.append(arima_forecast.values[0])

    # Scale back the predictions and test data
    arima_yhat = arima_yhat
    real_data = df['Close'][-60:].values

    # Calculate RMSE
    rmse = np.sqrt(np.mean(((arima_yhat - real_data) ** 2)))
    rmse_list.append(rmse)

    # Calculate MAE
    mae = np.mean(np.abs(arima_yhat - real_data))
    mae_list.append(mae)

    # Calculate DW statistic
    residuals = arima_yhat - real_data
    dw = np.sum(np.diff(residuals)**2) / np.sum(residuals**2)
    dw_list.append(dw)

    # Calculate MAPE
    mape = np.mean(np.abs((arima_yhat - real_data) / real_data)) * 100
    mape_list.append(mape)

    fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15, 6))
    fig.suptitle(f'ARIMA Predictions for {filename}\nRMSE: {rmse:.2f}, MAE:
{mae:.2f}, DW: {dw:.2f}, MAPE: {mape:.2f}')

    # Plot predictions and real data for the last 18 days in the first
subplot
    axes[0].plot(arima_yhat)
    axes[0].plot(real_data)
    axes[0].legend(["Predictions", "Real data"])

    preds = df['Close'].values

```

```

    preds[-60:] = arima_yhat
    # Plot predictions and real data for the entire data range in the second
subplot
    axes[1].plot(preds)
    axes[1].plot(df['Close'])
    axes[1].legend(["Predictions", "Real data"])

    # Adjust the layout
    plt.tight_layout()

    # Show the plots
    plt.show()

# Calculate the mean metrics
mean_rmse = np.mean(rmse_list)
mean_mae = np.mean(mae_list)
mean_dw = np.mean(dw_list)
mean_mape = np.mean(mape_list)

# Print the mean metrics
print("Mean Root Mean Squared Error:", mean_rmse)
print("Mean Absolute Error:", mean_mae)
print("Mean Durbin-Watson statistic:", mean_dw)
print("Mean Absolute Percentage Error:", mean_mape)

# In[ ]:

get_ipython().system('pip install prophet')

# In[ ]:

# Define the directory path of the cryptocurrency price data
data_dir = '/kaggle/input/cryptocurrencypricehistory'

# Get the list of filenames in the directory
filenames = os.listdir(data_dir)

logging.getLogger("prophet").setLevel(logging.WARNING)
logging.getLogger("cmdstanpy").disabled = True

# Create empty lists to store the metrics
rmse_list = []
mae_list = []
dw_list = []
mape_list = []

with warnings.catch_warnings():
    warnings.filterwarnings("ignore")

    # Loop through each cryptocurrency dataset
    for filename in filenames:
        # Read the CSV file
        df = pd.read_csv(os.path.join(data_dir, filename))

        # Preprocess the dataframe
        df = df.rename(columns={"Date": "ds", "Close": "y"})

        # Preprocess the dataframe

```

```

scaler = MinMaxScaler()
df['y'] = scaler.fit_transform(df['y'].values.reshape(-1, 1))

prophet_yhat = []
for i in range(59):
    # Create a new Prophet model
    model = Prophet()

    # Fit the model with the data
    model.fit(df[-60+i:])

    # Forecast using Prophet
    future = model.make_future_dataframe(periods=1, freq='D')
    forecast = model.predict(future)
    yhat = forecast['yhat'].values[0]

    # Scale back the predictions and true values
    real_data = df['y'].values[-60+i:]
    prophet_yhat.append(yhat)

# Scale back the predictions and test data
real_data = df['y'][-59:].values

# Calculate RMSE
rmse = np.sqrt(np.mean(((prophet_yhat - real_data) ** 2)))
rmse_list.append(rmse)

# Calculate MAE
mae = np.mean(np.abs(prophet_yhat - real_data))
mae_list.append(mae)

# Calculate DW statistic
residuals = prophet_yhat - real_data
dw = np.sum(np.diff(residuals)**2) / np.sum(residuals**2)
dw_list.append(dw)

# Calculate MAPE
mape = np.mean(np.abs((prophet_yhat - real_data) / real_data)) * 100
mape_list.append(mape)

# Set the figure size and title
plt.figure(figsize=(10, 6))
plt.title(f'Prophet Predictions for {filename}\nRMSE: {rmse:.2f}, MAE:
{mae:.2f}, DW: {dw:.2f}, MAPE: {mape:.2f}')

# Plot predictions and true values
plt.plot(prophet_yhat)
plt.plot(real_data)
plt.legend(["Predictions", "Real data"])

# Show the plot
plt.show()

# Calculate the mean metrics
mean_rmse = np.mean(rmse_list)
mean_mae = np.mean(mae_list)
mean_dw = np.mean(dw_list)
mean_mape = np.mean(mape_list)

# Print the mean metrics
print("Mean Root Mean Squared Error:", mean_rmse)
print("Mean Absolute Error:", mean_mae)
print("Mean Durbin-Watson statistic:", mean_dw)

```

```

print("Mean Absolute Percentage Error:", mean_mape)

# In[ ]:

df.shape[0]-1

# In[ ]:

# Define the directory path of the cryptocurrency price data
data_dir = '/kaggle/input/cryptocurrencypricehistory'

# Get the list of filenames in the directory
filenames = os.listdir(data_dir)

# Disable logging output from Prophet and cmdstanpy
logging.getLogger("prophet").setLevel(logging.WARNING)
logging.getLogger("cmdstanpy").disabled = True

# Create an empty list to store the RMSE for each model
rmse_list = []
with warnings.catch_warnings():
    warnings.filterwarnings("ignore")
    # Loop through each cryptocurrency dataset
    for filename in filenames:
        # Read the CSV file
        df = pd.read_csv(os.path.join(data_dir, filename))

        # Preprocess the dataframe
        df = df.rename(columns={"Date": "ds", "Close": "y"})

        # Preprocess the dataframe
        scaler = MinMaxScaler()
        df['y'] = scaler.fit_transform(df['y'].values.reshape(-1, 1))

        # Perform exponential smoothing
        prophet_yhat = []
        for i in range(59):
            # Print the data for the current iteration
            print(df['y'][-60 + i:])
            print(i)
            # Create a new Exponential Smoothing model
            model = ExponentialSmoothing(df['y'][-60 + i:], trend='add')

            # Fit the model with the data
            model_fit = model.fit()

            # Forecast using Exponential Smoothing
            yhat = model_fit.forecast(steps=1)

            # Scale back the prediction
            prophet_yhat.append(yhat)

        # Scale back the predictions and true values
        real_data = df['y'].values[-60:]
        prophet_yhat = np.array(prophet_yhat)

        # Calculate RMSE
        rmse = np.sqrt(np.mean(((prophet_yhat - real_data) ** 2)))
        rmse_list.append(rmse)

```

```

# Set the figure size and title
plt.figure(figsize=(10, 6))
plt.title(f'Exponential Smoothing Predictions for {filename}\nRMSE:
{rmse:.2f}, MAE: {mae:.2f}, DW: {dw:.2f}, MAPE: {mape:.2f}')

# Plot predictions and true values
plt.plot(prophet_yhat)
plt.plot(real_data)
plt.legend(["Predictions", "Real data"])

# Show the plot
plt.show()

# Calculate the mean RMSE
mean_rmse = np.mean(rmse_list)
print("Mean Root Mean Squared Error:", mean_rmse)

# In[ ]:

# Define the directory path of the cryptocurrency price data
data_dir = '/kaggle/input/cryptocurrencypricehistory'

# Get the list of filenames in the directory
filenames = os.listdir(data_dir)

# Disable logging output from Prophet and cmdstanpy
logging.getLogger("prophet").setLevel(logging.WARNING)
logging.getLogger("cmdstanpy").disabled = True

# Create empty lists to store the metrics
rmse_list = []
mae_list = []
dw_list = []
mape_list = []

with warnings.catch_warnings():
    warnings.filterwarnings("ignore")

# Loop through each cryptocurrency dataset
for filename in filenames:
    # Read the CSV file
    df = pd.read_csv(os.path.join(data_dir, filename))

    # Preprocess the dataframe
    df = df.rename(columns={"Date": "ds", "Close": "y"})

    # Preprocess the dataframe
    scaler = MinMaxScaler()
    df['y'] = scaler.fit_transform(df['y'].values.reshape(-1, 1))

    # Perform exponential smoothing
    prophet_yhat = []
    for i in range(59):
        # Create a new Exponential Smoothing model
        model = ExponentialSmoothing(df['y'][-60 + i:], trend='add')

        # Fit the model with the data
        model_fit = model.fit()

```

```

# Forecast using Exponential Smoothing
yhat = model_fit.forecast(steps=1)

# Scale back the prediction
prophet_yhat.append(yhat)

# Scale back the predictions and true values
real_data = df['y'].values[-60:]
prophet_yhat = np.array(prophet_yhat)

# Calculate RMSE
rmse = np.sqrt(np.mean(((prophet_yhat - real_data) ** 2)))
rmse_list.append(rmse)

# Calculate MAE
mae = np.mean(np.abs(prophet_yhat - real_data))
mae_list.append(mae)

# Calculate DW statistic
residuals = prophet_yhat - real_data
dw = np.sum(np.diff(residuals) ** 2) / np.sum(residuals ** 2)
dw_list.append(dw)

# Calculate MAPE
mape = np.mean(np.abs((prophet_yhat - real_data) / real_data)) * 100
mape_list.append(mape)

# Set the figure size and title
plt.figure(figsize=(10, 6))
plt.title(f'Exponential Smoothing Predictions for {filename}\nRMSE:
{rmse:.2f}, MAE: {mae:.2f}, DW: {dw:.2f}, MAPE: {mape:.2f}')

# Plot predictions and true values
plt.plot(prophet_yhat)
plt.plot(real_data)
plt.legend(["Predictions", "Real data"])

# Show the plot
plt.show()

# Calculate the mean metrics
mean_rmse = np.mean(rmse_list)
mean_mae = np.mean(mae_list)
mean_dw = np.mean(dw_list)
mean_mape = np.mean(mape_list)

# Print the mean metrics
print("Mean Root Mean Squared Error:", mean_rmse)
print("Mean Absolute Error:", mean_mae)
print("Mean Durbin-Watson statistic:", mean_dw)
print("Mean Absolute Percentage Error:", mean_mape)

```

ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ НА ПРИКЛАДІ КРИПТОВАЛЮТ

Гриценко Олександра
Володимирівна
КА-95

Керівник: Савастьянов Володимир Володимирович

**МЕТА РОБОТИ: РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
ПРОГНОЗУВАННЯ КУРСІВ КРИПТОВАЛЮТ ІЗ ВИКОРИСТАННЯМ
ЧАСОВИХ РЯДІВ.**

**ОБ'ЄКТ ДОСЛІДЖЕННЯ: ЗАСТОСУВАННЯ МЕТОДІВ ПРОГНОЗУВАННЯ
ЧАСОВИХ РЯДІВ ДЛЯ ПЕРЕДБАЧЕННЯ КУРСІВ КРИПТОВАЛЮТ.**

**ПРЕДМЕТ ДОСЛІДЖЕННЯ: МЕТОДИ ПРОГНОЗУВАННЯ ЧАСОВИХ
РЯДІВ ТА ЇХ ЗАСТОСУВАННЯ.**

ПОСТАНОВКА ЗАДАЧІ:

ЗАДАЧА ПОЛЯГАЄ В ПРОГНОЗУВАННІ КУРСУ КРИПТОВАЛЮТ З ВИКОРИСТАННЯМ МЕТОДІВ АНАЛІЗУ ЧАСОВИХ РЯДІВ ТА МАШИННОГО НАВЧАННЯ. НА ОСНОВІ ДОСТУПНИХ ДАНИХ ПРО КУРС КРИПТОВАЛЮТ НЕОБХІДНО РОЗРОБИТИ СИСТЕМУ ПРОГНОЗУВАННЯ, ЯКА ЗДАТНА ПЕРЕДБАЧАТИ МАЙБУТНІ ЗМІНИ ЦІН НА КРИПТОВАЛЮТИ ТА ПОРІВНЯТИ ДЕКІЛЬКА МЕТОДІВ.

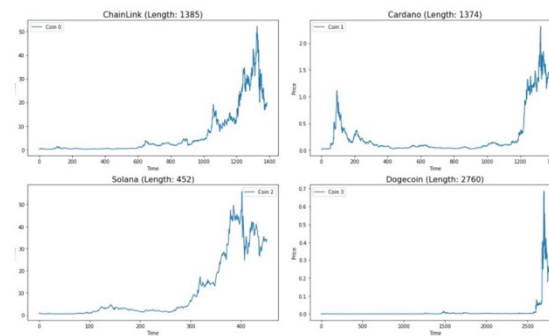
АКТУАЛЬНІСТЬ ТЕМИ РОБОТИ ПОЛЯГАЄ У РОСТІ ПОПУЛЯРНІСТІ АЛГОРИТМІЧНОЇ ТОРГІВЛІ НА РИНКУ КРИПТОВАЛЮТ. ДАНА ТЕМА ДИПЛОМНОГО ПРОЕКТУ Є АКТУАЛЬНОЮ В НАШ ЧАС ТОМУ, ЩО НА ОНЛАЙН ПЛОЩАДКАХ ДЛЯ ТОРГІВЛІ ВСЕ БІЛЬШЕ ТРАНЗАКЦІЙ ВІДБУВАЮТЬСЯ НА ОСНОВІ ПРОГНОЗІВ КОМП'ЮТЕРІВ.

SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap	
0	1	Chainlink	LINK	2017-09-21 23:59:59	0.207892	0.155292	0.189132	0.169680	2.126270e+06	5.938800e+07
1	2	Chainlink	LINK	2017-09-22 23:59:59	0.181675	0.139869	0.169826	0.150503	1.423220e+06	5.267605e+07
2	3	Chainlink	LINK	2017-09-23 23:59:59	0.157440	0.126297	0.149562	0.131129	1.186670e+06	4.589515e+07
3	4	Chainlink	LINK	2017-09-24 23:59:59	0.190250	0.130275	0.131110	0.170278	1.025740e+06	5.959730e+07
4	5	Chainlink	LINK	2017-09-25 23:59:59	0.182637	0.158580	0.170636	0.182198	1.021370e+06	6.376930e+07
...	
1380	1381	Chainlink	LINK	2021-07-02 23:59:59	18.520256	17.305501	18.220669	18.283561	6.865419e+08	7.990091e+09
1381	1382	Chainlink	LINK	2021-07-03 23:59:59	18.886003	17.898075	18.285458	18.538818	5.090763e+08	8.101641e+09
1382	1383	Chainlink	LINK	2021-07-04 23:59:59	19.894503	18.028957	18.548805	19.262838	6.802681e+08	8.418044e+09
1383	1384	Chainlink	LINK	2021-07-05 23:59:59	19.246891	17.814752	19.246891	18.371589	6.986191e+08	8.028560e+09
1384	1385	Chainlink	LINK	2021-07-06 23:59:59	20.240997	18.351877	18.351877	20.080466	1.156986e+09	8.775356e+09

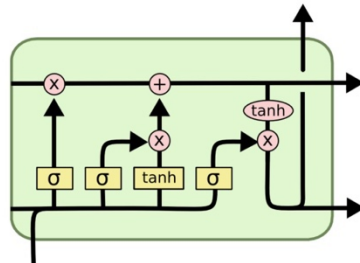
1385 rows × 10 columns

ЗОБРАЖЕННЯ ОДНІЄЇ ІЗ ТАБЛИЦЬ ІЗ ДАНИМИ

ЦІНИ НА КРИПТОВАЛЮТИ



СТРУКТУРА КОМІРКИ LSTM



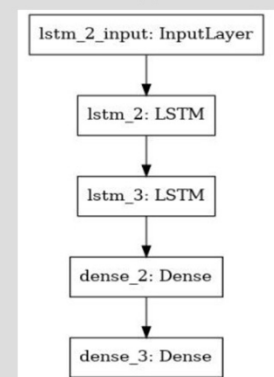
СТРУКТУРА МОДЕЛІ

Model: "sequential_3"

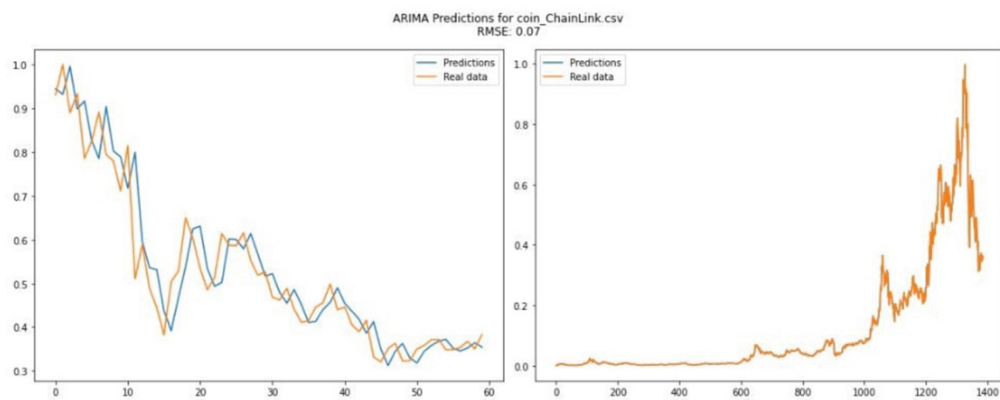
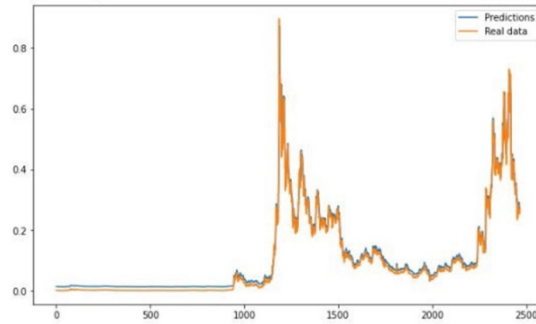
Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 60, 128)	66560
lstm_7 (LSTM)	(None, 64)	49408
dense_6 (Dense)	(None, 25)	1625
dense_7 (Dense)	(None, 1)	26

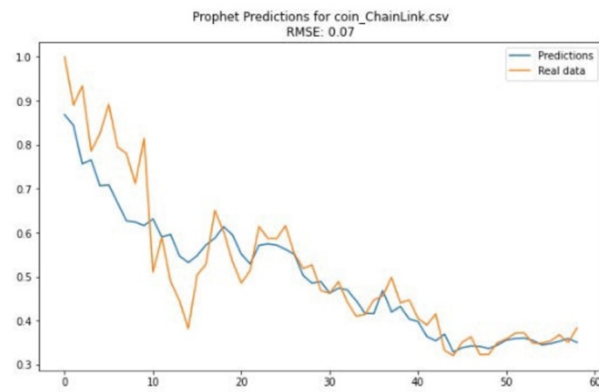
Total params: 117,619
 Trainable params: 117,619
 Non-trainable params: 0

ВІЗУАЛІЗАЦІЯ СТРУКТУРИ МОДЕЛІ



ГРАФІК РОБОТИ МОДЕЛІ НА ДАНИХ ДЛЯ ТЕСТУВАННЯ

ГРАФІК РОБОТИ МОДЕЛІ АІРМА НА
ТЕСТОВИХ ДАНИХ



РЕЗУЛЬТАТИ РОБОТИ МОДЕЛІ PROPNET

Model	ARIMA	Prophet	Exp Smoothing	LSTM
Mean RMSE	0.04	0.05	0.147	0.226
Mean MAE	0.03	0.03	0.102	0.154
Mean DW	2.3	1.13	0.164	0.006
Mean MAPE	6.8	6.99	19.73	23

РЕЗУЛЬТАТИ РОБОТИ МОДЕЛЕЙ

ВИСНОВОК

Прогнозування криптовалютних курсів має важливість та потенціал, але стикається зі складнощами, такими як висока невизначеність ринків та обмежена історична інформація. Проте, використання методів, таких як Prophet та Auto ARIMA, може допомогти зробити обґрунтовані рішення та покращити результати.