

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ центр післядипломної освіти \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

\_\_\_\_\_ Дослідження ефективності використання інформаційних технологій \_\_\_\_\_  
\_\_\_\_\_ на основі штучного інтелекту для прогнозування курсу криптовалют \_\_\_\_\_  
(тема)

Виконав:  
студент (ка) 2 курсу, групи ІІЗЗдм-22-1

\_\_\_\_\_ Наумов А.Б. \_\_\_\_\_  
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного  
забезпечення  
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник доц. Чуприна А.С.  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри

\_\_\_\_\_ (підпис)

\_\_\_\_\_ З.В.Дудар \_\_\_\_\_  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ центр післядипломної освіти \_\_\_\_\_  
 Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
 Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
 Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення \_\_\_\_\_  
 Тип програми \_\_\_\_\_ освітньо-наукова програма \_\_\_\_\_  
 Освітня програма \_\_\_\_\_ Інженерія програмного забезпечення \_\_\_\_\_  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_\_\_» \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Наумову Антону Борисовичу \_\_\_\_\_  
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження ефективності використання інформаційних технологій на основі штучного інтелекту для прогнозування курсу криптовалют»

Затверджена наказом по університету від 22.04.2024р. № 60 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024

3. Вихідні дані до роботи встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки, перелік методів машинного навчання.

4. Перелік питань, що потрібно опрацювати в роботі

мета роботи, аналіз предметної області, методи та етапи досліджень, методологія досліджень, планування проведення експериментів, проведення експериментального дослідження, розробка ПЗ, тренування моделі, оцінка отриманих \_\_\_\_\_ результатів, \_\_\_\_\_ висновки.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	23.01 – 14.02.24	виконано
2	Аналіз та вибір API для дослідження	15.02 – 24.02.24	виконано
3	Аналіз та моделювання предметної області	17.02 – 28.02.24	виконано
4	Планування експериментів	25.02 – 28.02.24	виконано
5	Програмна реалізація кожного з обраних для дослідження API	25.02 – 01.04.24	виконано
6	Експериментальні дослідження	02.04 – 20.04.24	виконано
7	Аналіз результатів експериментальних досліджень та розробка рекомендацій	20.04 – 23.04.24	виконано
8	Написання та оформлення статті та тез доповіді	17.04 – 23.04.24	виконано
9	Підготовка пояснювальної записки	01.04 – 26.04.24	виконано
10	Підготовка презентації та доповіді	26.04 – 9.06.24	виконано
11	Нормоконтроль	10.06 – 11.06.24	виконано
12	Рецензування	11.06 – 14.06.24	виконано
13	Занесення диплома в електронний архів	16.06.2024	виконано
14	Попередній захист	16.06.2024	виконано
15	Допуск до захисту у зав. кафедри	19.06.2024	виконано

Дата видачі завдання 23 квітня 2024р.

Студент (ка) \_\_\_\_\_  
(підпис)

\_\_\_\_\_ Наумов А.Б.

Керівник роботи \_\_\_\_\_  
(підпис) (посада, прізвище, ініціали)

\_\_\_\_\_ доц. Чуприна А.С.

## РЕФЕРАТ / ABSTRACT

Звіт: 98 с., 26 рис., 13 табл., 49 джерел.

АЛГОРИТМИ ПРОГНОЗУВАННЯ, АНАЛІЗ НАСТРОЇВ, БІТКОЙН, КРИПТОВАЛЮТИ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗНИЙ АНАЛІЗ, РИНОК КРИПТОВАЛЮТ, BERT, FLAIR, LSTM, NLP, SPACY, TEXTBLOB, TENSORFLOW, VADER

Об'єктом дослідження є прогнозування курсу криптовалют з застосуванням алгоритмів машинного навчання.

Метою роботи є визначення найбільш ефективних методів машинного навчання серед обраних для найкращого прогнозування курсу криптовалют за допомогою аналізу оцінок новосних настроїв.

Методами дослідження є аналіз предметної області, визначення методик та етапів дослідження, встановлення методології досліджень, планування проведення експериментів, проведення експериментів, аналіз результатів, рекомендації до практичного застосування.

У результаті кваліфікаційної роботи було розроблене програмне забезпечення, що дозволяє виконувати збір новин з ресурсів Google та Yahoo, препроцесинг отриманих даних, поляризацію настроїв за допомогою бібліотек TextBlob, Vader, Flair, Spacy та Bert та навчання регресійної моделі з урахуванням отриманих метрик настроїв з застосуванням бібліотеки TensorFlow та високорівневого API Keras. Було проведено експеримент для порівняння результатів реалізованого алгоритму за заданими критеріями і знайдено найбільш ефективний підхід для прогнозування курсу криптовалют.

BERT, BITCOIN, CRYPTOCURRENCIES, CRYPTOCURRENCY MARKET, FLAIR, FORECASTING ALGORITHMS, LSTM, MACHINE LEARNING, NLP, SENTIMENT ANALYSIS, PREDICTIVE ANALYSIS, SPACY, TEXTBLOB, TENSORFLOW, VADER

The object of the research is forecasting the exchange rate of cryptocurrencies using machine learning algorithms.

The purpose of the work is to determine the most effective methods of machine learning among the selected ones for the best forecasting of the cryptocurrency exchange rate using the analysis of news sentiment estimates.

The methods are analysis of the subject area, determination of research techniques and stages, establishment of research methodology, planning of experiments, conducting experiments, analysis of results, recommendations for the practical application.

As a result of the qualification work, software was developed that allows you to collect news from Google and Yahoo resources, preprocess the received data, polarize sentiments using the TextBlob, Vader, Flair, Spacy and Bert libraries and train a regression model taking into account the received sentiment metrics using the library TensorFlow and Keras high-level API. An experiment was conducted to compare the results of the implemented algorithm according to the specified criteria and the most effective approach for forecasting the exchange rate of cryptocurrencies has been obtained.

Я, Наумов Антон Борисович, студент гр. ПЗЗдм-22-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження ефективності використання інформаційних технологій на основі штучного інтелекту для прогнозування курсу криптовалют», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	11
1.1 Аналіз предметної галузі дослідження	11
1.2 Постановка задачі	15
2 МЕТОДИ ТА ЕТАПИ ДОСЛІДЖЕННЯ	16
2.1 Обґрунтування методів дослідження	16
2.2 Етапи проведення наукового дослідження.	16
3 ТЕОРЕТИЧНІ ВІДОМОСТІ	18
3.1 Штучні нейронні мережі	18
3.2 Рекурентні нейронні мережі	19
3.3 LSTM моделі довго короткочасної пам'яті	24
4 ЕКСПЕРИМЕНТАЛЬНА СКЛАДОВА ДОСЛІДЖЕННЯ	29
4.1 Вибір криптовалют для дослідження	29
4.2 Збір даних.	33
4.4 Підготовка даних	35
4.5 Аналіз настроїв	36
4.6 Підготовка датасетів	40
4.7 План проведення експерименту.	42
5 РЕАЛІЗАЦІЯ МОДЕЛЕЙ	45
5.1 Аналіз настроїв за допомогою TextBlob	45
5.2 Аналіз настроїв за допомогою Vader.	45
5.3 Аналіз настроїв за допомогою Flair	46
5.4 Аналіз настроїв за допомогою Spacy	46
5.5 Аналіз настроїв за допомогою Bert	47
5.6 Модель прогнозування ціни	48
6 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ	51
6.1 Навчання з використанням датасету TextBlob	51
6.2 Навчання з використанням датасету Vader	53
6.3 Навчання з використанням датасету Flair	54
6.4 Навчання з використанням датасету Spacy.	55
6.5 Навчання з використанням датасету Bert	55
6.6 Оптимізація прогнозування моделі	56
6.7 Аналіз результатів	58
6.8. Аналіз подальшого розвитку дослідження	60
6.9. Практична цінність отриманих результатів	61

	7
ВИСНОВКИ	62
ДОДАТОК А	70
ДОДАТОК Б	71
ДОДАТОК В	82
ДОДАТОК Г	97
ДОДАТОК Д	98

## ПЕРЕЛІК СКОРОЧЕНЬ

ШІ - штучний інтелект

МН - машинне навчання

BTC - Bitcoin

ШНМ - штучна нейронна мережа

PHM - рекурентна нейронна мережа

NLP - обробка природної мови

GRU - Gated Recurrent Unit

ARIMA - Autoregressive Integrated Moving Average

LSTM - Long Short-Term Memory

VADER - Valence Aware Dictionary and Sentiment Reasoner

ANN - Artificial Neural Network

RNN - Recurrent Neural Network

TPS - Transactions Per Second

MSE - Mean Squared Error

MRSE - Mean Root Squared Error

MAE - Mean Absolute Error

## ВСТУП

В сучасному світі фінансів криптовалютні ринки займають особливе місце, ставши предметом пильної уваги як серед фахівців, так і серед широкого загалу. Спостерігається стрімкий розвиток цих ринків, який відбувається на фоні швидкого технологічного прогресу та зміни у споживацьких уподобаннях. Криптовалюти, такі як Bitcoin, Ethereum, та інші, стали не тільки об'єктом інвестування, але й новою формою фінансових активів, що впливають на глобальну економіку [1].

За останні десятиріччя криптовалютні ринки пройшли шлях від нішевих експериментів до визнаної складової фінансової системи. Ринки цифрових валют стали місцем, де зустрічаються як інноваційні технології, так і традиційні принципи фінансового управління. Зростання капіталізації ринків криптовалют та зростання інтересу соціуму до цих активів створюють нові виклики і можливості для дослідження їхньої природи, динаміки та впливу на глобальну фінансову систему.

Роль криптовалют у сучасному мирі фінансів надає їм значущість як для економічного розвитку окремих країн, так і для формування міжнародних фінансових відносин [2]. Їхній вплив поступово розширюється на різні сфери діяльності, від торгівлі та інвестування до фінансового посередництва та технологічних інновацій. Розмаїття криптовалют та ріст їхньої вартості стимулюють пошук нових способів використання цих активів та розвитку інструментів аналізу та управління ними.

Криптовалюти стали не тільки новою формою цифрових активів, але й предметом інвестицій та торгівлі, що створює величезний попит на інструменти для аналізу та прогнозування ціни на них. У цьому контексті, використання ІІІ стає важливим фактором для досягнення успішних інвестиційних стратегій та управління ризиками.

Однак, перед тим як розглядати роль ІІІ в прогнозуванні курсу криптовалют, важливо розуміти суть самого явища криптовалют та їхню динаміку на ринку. Криптовалюти є децентралізованими цифровими активами, які

базуються на технології блокчейн. Вони відрізняються від традиційних валют тим, що їхніми власниками може бути будь-хто, а трансакції здійснюються без посередництва банків чи інших фінансових установ. Ця характеристика робить криптовалюту особливо чутливими до змін на ринку та економічних факторів.

ШІ відіграє ключову роль у прогнозуванні курсу криптовалют завдяки своїм аналітичним можливостям та здатності швидко адаптуватися до нових умов [3]. Використання методів машинного навчання та алгоритмів аналізу дозволяє аналізувати величезні обсяги даних, включаючи цінову динаміку, обсяги торгів, новини та соціальні медіа, що допомагає робити точні прогнози щодо майбутнього руху цін на такі активи.

Прогнозування курсу криптовалют – дуже не проста задача. Існує дуже багато факторів, що мають вплив на динаміку курсу цифрових активів. У сучасному світі новини мають значний вплив на формування динаміки цін багатьох фінансових активів, у тому числі і криптовалют. Аналіз настроїв новин є актуальною темою дослідження ефективності МН [4]. настрої дозволяють кількісно оцінити емоційне забарвлення та суб'єктивність кожної новини та застосувати ці дані для аналізу.

Особливу актуальність цей підхід набуває на фоні бурхливого росту впливу соціальних мереж, ЗМІ та інформаційних технологій на свідомість сучасної людини та вибір, який ми робимо кожного дня. Цей вплив розповсюджується на всі верстви суспільства включаючи професійних інвесторів, трейдерів та учасників криптовалютного ринку.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз предметної галузі дослідження

Криптовалюти, такі як Bitcoin, Ethereum та інші, представляють собою новий тип цифрових активів, що використовують криптографічні методи для забезпечення безпеки та контролю транзакцій. Вони є предметом інтенсивного дослідження з боку як академічної, так і комерційної громадськості.

Криптовалюта – це програмне забезпечення яке має специфічні властивості використання, що дозволяє залучати їх у сучасні фінансові інструменти та навіть використовувати на валютних ринках. Сучасний огляд криптовалютних систем наведено у статтях [5-6]. У роботі [5] наведено порівняльний аналіз алгоритмів майнінгу, у [6] сформульовано основні виклики та можливості, а також проведено аналіз основних методів штучного інтелекту, які використовуються для вирішення найважливіших проблем криптовалют, що пов'язані з прогнозуванням цін, ризиків, загроз кібербезпеки та ряду інших. Основною ідеєю криптовалюти є можливість проведення анонімних транзакцій [4]; робота [5] показує особливості моделі децентралізованої конфіденційної платіжної системи. Крім того, існують інші типи криптовалют, але це дослідження орієнтується лише на дослідження факторів, що впливають на «монетний» тип криптовалюти (і це є основним обмеженням щодо типів криптовалют у даному дослідженні) [6-8].

Криптовалюти – це електронний засіб обміну, яким можуть торгувати різні торгові групи через біржі цифрових валют. Зростаючий список альткойнів свідчить про поширення криптовалютних бірж, що з'явилися протягом останнього десятиліття, через те, що оскільки законодавчі органи багатьох країн у всьому світі тільки намагаються зрозуміти відповідні правила захисту інвесторів і виробити відповідне законодавство. Незважаючи на те, що немає центрального органу, який реєстрував би ці біржі, за різними оцінками, наразі на глобальному ринку діють до п'ятисот бірж, що використовують різні платформи, мають різне географічне охоплення і підпорядковуються різним нормативним документам. Криптовалютні ринки мають багато спільного як з валютними, так і з фондовими ринками, як-от книги лімітованих замовлень і відповідні алгоритми, а також біржі,

які можуть бути як централізованими, так і децентралізованими. Оскільки ці ринки все ще перебувають у стані зародження, те, який вид чи форму вони приймуть у майбутньому, буде залежати від їхнього сприйняття у майбутнього з боку споживачів, так і з боку інвестиційної спільноти, а також світових регуляторів.

Роботи [7,8] надають опис структури ринку криптовалют, робота [9] дає нетехнічний огляд концепції та поточної структури ринку криптовалют для дослідників у галузі економіки, фінансів, математики та інформатики.

Усі криптовалюти є програмним забезпеченням, і їхня ціна залежить від взаємопоеднання багатьох факторів [10]. Є дві головні частини, що містять найважливіші фактори: сам продукт (криптовалюта як продукт та їх механізм) та торговий ринок. Ніщо не живе поза навколишнім середовищем, і криптовалюта не є виключенням. Зв'язок між вибором факторів, моделями та алгоритмами блокчейну та функціонуванням екосистеми криптовалют описано в роботах [11-13]. Криптовалюта - дуже поширене програмне забезпечення, тому загальні правила, які стосуються будь-якого продукту в цій сфері, також можуть впливати на неї. Приклади таких функцій описані в [14-16].

У динамічному світі криптовалют, де ціни можуть швидко змінюватися, прогнозування курсу стає безцінним інструментом для інвесторів. Прогнози допомагають оцінити потенційний прибуток або збитки, дозволяючи приймати обґрунтовані інвестиційні рішення та знижувати ризики. Знання майбутніх цінових коливань дозволяє вибрати найвигідніші моменти для купівлі чи продажу криптовалют, максимізуючи прибуток. Фактори, які можуть вплинути на ціну криптовалюти та її використання в регресійні моделі, описані в роботі [17].

Це дуже актуальна тема, що привертає увагу багатьох дослідників [18-21]. Було проведено дослідження, чи може ціна золота використовувати як прогноз для ціни біткойна за допомогою трьох алгоритмів глибокого навчання CNN, LSTM і GRU [22]. Висновок полягає в тому, що прогнозована ціна моделі, яка використовує лише ціну золота, відхиляється від справжньої ціни біткойна, а точність прогнозу моделі LSTM є найкращою з трьох. В роботі [23] було

розширено спектр пояснювальних змінних на основі ринку криптовалют та індексу макроринку (індекс фондового ринку, ціна на сиру нафту, обмінний курс тощо) та пошукового індексу, загалом 40 пояснювальних змінних для прогнозування ціни біткойна. Що стосується дослідження прогнозування ціни біткойна, методи поділяються на часові ряди та машинне навчання. Кілька досліджень [24-27] дійшли висновку, що точність прогнозування ARIMA не така висока, як у машинного навчання.

LSTM, як контрольоване дослідження регресії випадкового лісу в цьому дослідженні, вивчалось як цільова модель багато разів у попередній літературі [25, 28-30] Було задіяно чотири алгоритми глибокого навчання (регресію Тейля-Сена, регресію Хубера, LSTM і GRU), щоб передбачити ціну біткойна. Точність алгоритму LSTM було найвищою.

Крім прогнозування ціни біткойна, існує багато досліджень [31-34], які використовують LSTM для прогнозування інших цифрових валют. Деякі з них використовували LSTM для прогнозування ціни ефіру з точністю 84,2%.

У роботі [35] використовували гібридний CNN-LSTM для проведення експериментів із прогнозуванням біткойнів (BTC), Ethereum (ETH) і Ripple (XRP) із найвищою ринковою вартістю на той час.

Використовуючи аналіз настроїв як опцію для обчислень, можна використовувати їх для прогнозування цін на біткойни та інші криптовалюти в різні проміжки часу. Однією з ключових характеристик крипторинку є те, що коливання його цін не залежить від інституційного регулювання грошей, а залежить від сприйняття та думки людей. Таким чином, аналіз зв'язку між соціальними медіа та веб-пошуком має вирішальне значення для прогнозування курсу криптовалюти.

Прогнозування цін на криптовалюти за допомогою аналізу настроїв є кульмінацією двох окремих процесів. Аналіз настрою можна виконати за допомогою двох методів: на основі правил або за допомогою методів машинного навчання. Метод на основі правил використовує лексикон слів і встановлені правила для класифікації настроїв. Наприклад, інструмент VADER - це

приклад аналізу сентиментів на основі лексики та правил, який не потребує навчання [36]. Він створює вектор оцінок настрою: позитивний, негативний і нейтральний, полярності яких потім нормалізуються між 0 і 1.

З іншого боку, методи машинного навчання аналізують сентименти на основі таких алгоритмів, як Random Forest, Support Vector Machines тощо. Для прогнозування настроїв [37] порівнює логістичну регресію, лінійні опорні векторні машини та Naïve Bayes.

Було проведено дослідження джерел [17, 38] що стверджують, що у прогнозуванні криптовалют на довгостроковий період часу майже немає сенсу.

У роботі [39] автори провели дослідження прогнозування курсу BTC із використанням алгоритму бустинга. У роботі [40] було виконано дослідження емоційної оцінки новин для прогнозування курсу біткоіна

Стенквіст і Льонн [41] досліджували суспільні настрої на підставі публікацій у Twitter, щоб передбачити ціну біткойна. 2,27 мільйона твітів, пов'язаних з біткойнами, було зібрано для аналізу настроїв, щоб вказати на зміну ціни в найближчому майбутньому. Це робиться за допомогою методу виключного приписування падіння або зростання на основі серйозності сукупної зміни настрою Twitter протягом періодів часу від 5 хвилин до 4 годин, а потім зміщення цих прогнозів вперед на 1, 2, 3 або 4 часових періодів відносно відповідного часу інтервалу BTC. Цей метод дав 83-відсоткову точність. Крім того, прогноз було зроблено лише тоді, коли середнє значення настрою було обмежено мінімальною зміною на 2,2 відсотка. У майбутньому цей аналіз можна покращити, додавши спеціальний лексикон для домену, що дасть більш репрезентативну думку.

Xin Huang [42] запропонував модель аналізу настроїв на основі LTSM. Дані, зібрані для визначення настроїв, були отримані з найпопулярнішої китайської соціальної медіа-платформи Sina-Weibo. Для прогнозування цінової тенденції на майбутнє використовувалася рекурентна нейронна мережа на основі довгострокової пам'яті (LSTM) разом із історичними цінами на криптовалюту. Результати показали 87-відсотковий рівень точності. Це на 15,4 вище, ніж традиційний метод авторегресії, який використовується.

Автори [43] запропонували модель із використанням інструментів машинного навчання та даних соціальних медіа для прогнозування цін на певну кількість криптовалют. У цій моделі використовувалися нейронні мережі (NN), опорні векторні машини (SVM) і випадковий ліс (RF). Результати цієї роботи показують, що прогнозувати криптовалюту можна за допомогою аналізу настроїв і за допомогою інструментів машинного навчання.

## 1.2 Постановка задачі

Ознайомившись з наведеними методами прогнозування курсу криптовалют, можна сказати, що існує багато параметрів, які що можна аналізувати, і багато методів покращення прогнозування курсу криптовалют.

Також було виявлено, що прогнозування курсу криптовалют за допомогою аналізу новинних настроїв є перспективною темою для дослідження. А також те, що точність прогнозування на довгостроковий період часу є досить низькою, тому було вирішено зупинитися на короткочасному прогнозуванні.

Також було виявлено, що рекурентна нейронна мережа на основі довгострокової пам'яті (LSTM) показує досить високу точність прогнозування.

Метою даної роботи є проведення дослідження для порівняння ефективності застосування поширених інструментів для аналізу настроїв та алгоритму PHM LSTM для прогнозування ціни закриття криптовалюти на поточний день.

Для проведення дослідження алгоритм PHM LSTM буде реалізований для прогнозування, будуть використані датасети з новинних стрічок, різні реалізації алгоритмів для аналізу настроїв та будуть визначені переваги та недоліки кожного з них.

## 2 МЕТОДИ ТА ЕТАПИ ДОСЛІДЖЕННЯ

### 2.1 Обґрунтування методів дослідження

Основна мета цього дослідження полягає у порівнянні різних інструментів аналізу настроїв для прогнозування курсу криптовалют, що відбувається за допомогою емпіричного аналізу.

Цей аналіз базується на проведенні експерименту, для якого необхідно створити стабільне середовище і підготувати відповідні датасети.

Метою експерименту є відповідь на конкретне питання, а саме визначення найефективніших методів прогнозування ціни закриття криптовалют на сьогоднішній день за допомогою аналізу настроїв зі стрічок новин.

### 2.2 Етапи проведення наукового дослідження.

Основною метою цього дослідження є порівняння різних інструментів аналізу настроїв для визначення оптимальних і найкращих з них для прогнозування курсу криптовалют.

Для цього дане наукове дослідження буде складатися з наступних етапів:

- вибір криптовалюти для аналізу
- дослідження РНМ;
- визначення метрик для порівняння;
- реалізація обраних алгоритмів та методів для порівняння;
- збір даних для проведення експерименту;
- складання плану експерименту;
- визначення найкращого методу для аналізу настроїв, згідно з обраними метриками;

- пошук методів оптимізації моделі;
- аналіз результатів дослідження;

## 3 ТЕОРЕТИЧНІ ВІДОМОСТІ

### 3.1 Штучні нейронні мережі

Нейронні мережі – це набір алгоритмів, які дуже нагадують людський мозок і призначені для розпізнавання шаблонів. Вони інтерпретують сенсорні дані за допомогою машинного сприйняття, маркування або кластеризації вихідних даних. Вони можуть розпізнавати числові шаблони, що містяться у векторах, у які мають бути переведені всі дані реального світу (зображення, звук, текст або часові ряди). ШНМ складаються з великої кількості взаємопов'язаних елементів обробки (нейронів), які працюють разом для вирішення проблеми.

ШНМ використовуються для прогнозного моделювання, адаптивного керування та інших програм, де їх можна навчати за допомогою набору даних. Вони також використовуються для вирішення завдань штучного інтелекту. Мережі можуть вчитися на досвіді та робити висновки зі складного та, здавалося б, непов'язаного набору інформації.

ШНМ складається із з'єднаних одиниць або вузлів, званих штучними нейронами, які вільно моделюють нейрони в мозку. Вони з'єднані краями, які моделюють синапси в мозку. Штучний нейрон отримує сигнали від підключених нейронів, потім обробляє їх і посиляє сигнал іншим підключеним нейронам

«Сигнал» – це дійсне число, а вихід кожного нейрона обчислюється за допомогою деякої нелінійної функції суми його входів, яка називається функцією активації. Нейрони та ребра зазвичай мають вагу, яка коригується в міру навчання. Вага збільшує або зменшує силу сигналу під час підключення.

Як правило, нейрони агреговані в шари. Різні шари можуть виконувати різні перетворення своїх вхідних даних. Мережа зазвичай називається глибокою нейронною мережею, якщо вона має принаймні 2 прихованих шари.

Сигнали проходять від першого шару (вхідного шару) до останнього (вихідного шару), можливо, проходячи через кілька проміжних шарів (прихованих шарів). (див рис. 3.1).

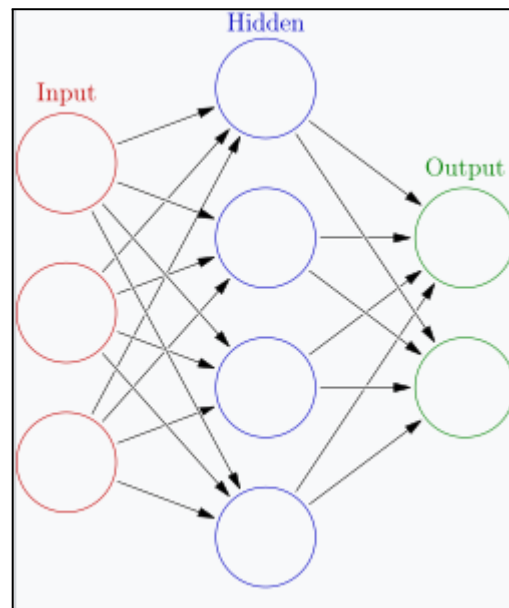


Рисунок 3.1 – Схема штучної нейронної мережі [46]

### 3.2 Рекурентні нейронні мережі

Рекурентна нейронна мережа - це будь-яка мережа, яка містить цикл у своїх мережевих з'єднаннях, що означає, що значення деякої одиниці прямо чи опосередковано залежить від її власних попередніх виходів як вхідних даних. Хоча такі мережі є потужними, їх важко обґрунтувати та навчити.

Однак у загальному класі рекурентних мереж існують обмежені архітектури, які виявилися надзвичайно ефективними у застосуванні до мови. Розглянемо клас рекурентних мереж, які називаються мережами Елмана або простими рекурентними мережами. Ці мережі корисні самі по собі та служать основою для більш складних підходів, таких як мережі довготривалої короткочасної пам'яті (LSTM), які обговорюються далі в цьому розділі. Коли ми будемо використовувати термін РНМ, ми матимемо на увазі ці простіші, більш обмежені мережі (хоча ви часто зустрічатимете термін РНМ для позначення будь-якої мережі з рекурентними властивостями, включаючи LSTM).

Прихований рівень містить повторюване з'єднання як частину свого входу. Тобто значення активації прихованого шару залежить від поточного введення, а також значення активації прихованого шару з попереднього кроку часу. (див. рис. 3.2)

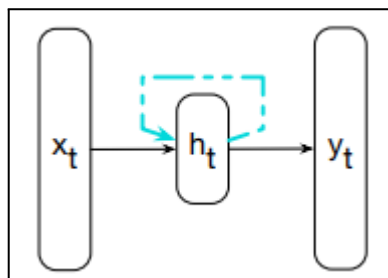


Рисунок 3.2 – Структура РНМ [44]

У звичайних мережах прямого зв'язку, вхідний вектор, що представляє поточний вхід,  $x_t$ , множиться на вагову матрицю, а потім пропускається через нелінійну функцію активації для обчислення значень для шару прихованих одиниць. Потім цей прихований шар використовується для обчислення відповідного результату,  $y_t$ . На відміну від нашого попереднього віконного підходу, послідовності обробляються шляхом надання в мережу одного елемента за раз. Ми будемо використовувати нижні індекси для представлення часу, тому  $x_t$  означатиме вхідний вектор  $x$  у момент часу  $t$ .

Ключова відмінність від прямої мережі полягає в повторюваному зв'язку, показаному на малюнку пунктирною лінією. Це посилення доповнює вхідні дані для обчислення на прихованому шарі значенням прихованого шару з попереднього моменту часу. Прихований рівень із попереднього часового кроку забезпечує форму пам'яті або контексту, який кодує попередню обробку та інформує про рішення, які мають бути прийняті в наступні моменти часу. Важливо те, що цей підхід не накладає обмеження фіксованої довжини на цей попередній контекст; контекст, втілений у попередньому прихованому шарі, може містити інформацію, що тягнеться до початку послідовності. Додавання цього тимчасового виміру робить РНМ більш складними, ніж неповторювані архітектури.

Давайте подивимося, як застосувати РНМ до задачі моделювання мови. Згадаємо, що мовні моделі передбачають наступне слово в послідовності з урахуванням деякого попереднього контексту. Наприклад, якщо попередній контекст «Дякуємо за все», і ми хочемо щоб дізнатися, наскільки ймовірно, що наступне слово буде «риба», ми б обчислили за формулою 3.1:

$$P(\text{fish}|\text{Thanks for all the}) \quad (3.1)$$

Мовні моделі дають нам можливість призначити таку умовну ймовірність кожному можливе наступне слово, що дає нам розподіл по всьому словниковому запасу. Ми також можемо призначити ймовірності цілим послідовностям, комбінуючи ці умовні ймовірності з правилом ланцюга.

Моделі мови *n*-gram у розділі обчислюють підрахунок ймовірності наданого слова його появи з *n*-1 попередніми словами. Таким чином, контекст має розмір *n*-1.

Для мовних моделей прямого зв'язку з контекстом є розмір вікна.

Моделі мови обробляють вхідну послідовність один слово за раз, намагаючись передбачити наступне слово з поточного слова та попередній прихований стан. Таким чином, РНМ не мають проблеми обмеженого контексту, ніж *n*-gram моделі або фіксований контекст, який мають мовні моделі прямого зв'язку, оскільки прихований стан може в принципі представляти інформацію про всі попередні слова назад до початку послідовності.

Порівняння традиційної ШНМ з РНМ можемо побачити на рисунку 3.3.

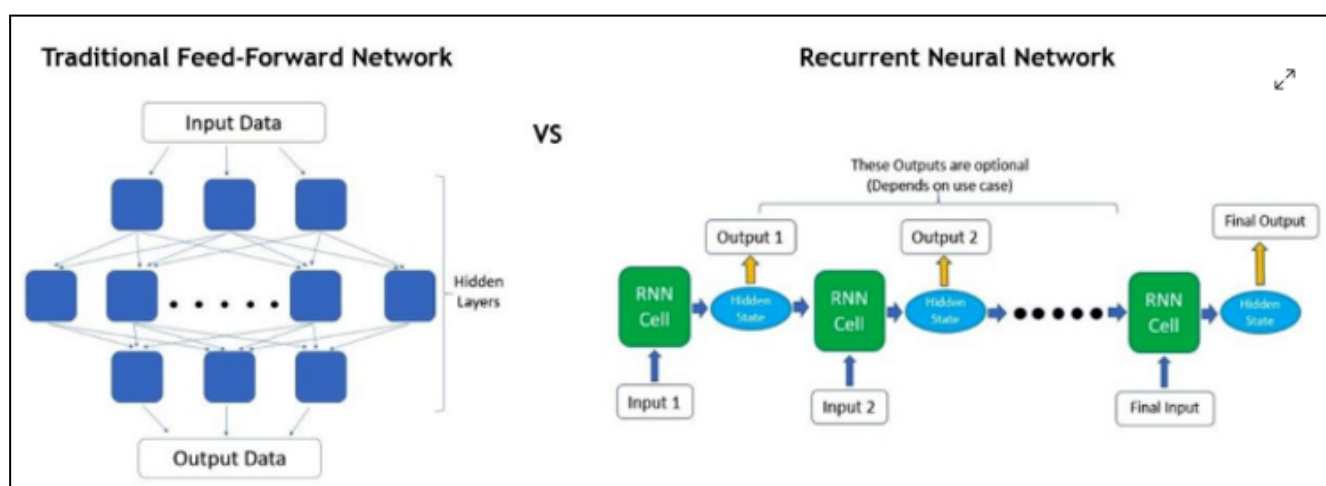


Рисунок 3.3 – Порівняння традиційної ШНМ з РНМ [43]

Моделі РНМ в основному використовуються в області обробки природної мови (NLP) і розпізнавання мовлення. Рекурентні мережі дозволяють нам

оперувати послідовністю векторів: послідовністю на вході, виході або, в найзагальнішому випадку, обома. Можна виділити такі типи РНМ (див. рис. 3.4.):

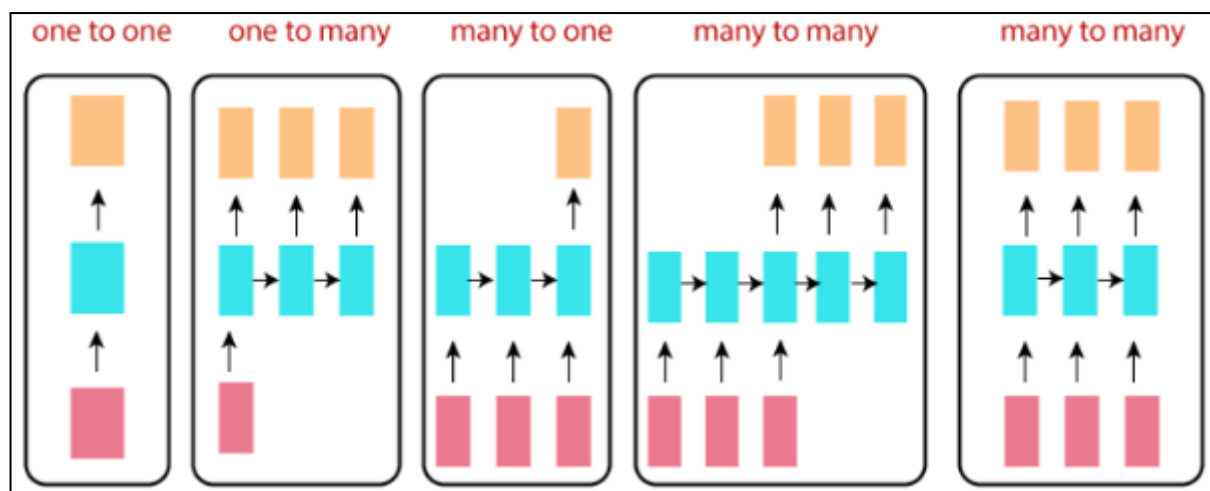


Рисунок 3.4 – Різні типи РНМ [43]

Один до одного – також називаються простими нейронними мережами. Вони мають справу з фіксованим розміром вхідних даних і фіксованим розміром виходу, де вони не залежать від попередньої інформації/виходу. Наприклад класифікація зображень.

Один до багатьох – має справу з фіксованим розміром інформації як вхідних даних, що дає послідовність даних як вихідні дані. Наприклад: Субтитри зображень приймають зображення як вхідні дані та виводять речення зі слів.

Багато-до-одного – приймає послідовність інформації як вхід і виводить фіксований розмір виходу. Наприклад: аналіз настроїв, де будь-яке речення класифікується як виражає позитивні чи негативні настрої.

Багато до багатьох – приймає послідовність інформації як вхідну інформацію та обробляє періодичні виходи як послідовність даних. Наприклад: машинний переклад, де RNN читає будь-яке речення англійською мовою, а потім виводить речення французькою.

Двонаправлений багато-до-багатьох – має синхронізований послідовний вхід і вихід. Наприклад: класифікація відео, де ми хочемо позначити кожен кадр відео.

На відміну від нейронних мереж прямого зв'язку, РНМ можуть використовувати свій внутрішній стан (пам'ять) для обробки послідовностей вхідних даних. Це робить їх застосовними для таких завдань, як несегментоване, пов'язане розпізнавання рукописного тексту або розпізнавання мовлення. В інших нейронних мережах усі входи незалежні один від одного. Але в РНМ усі входи пов'язані один з одним (див рис. 3.5).

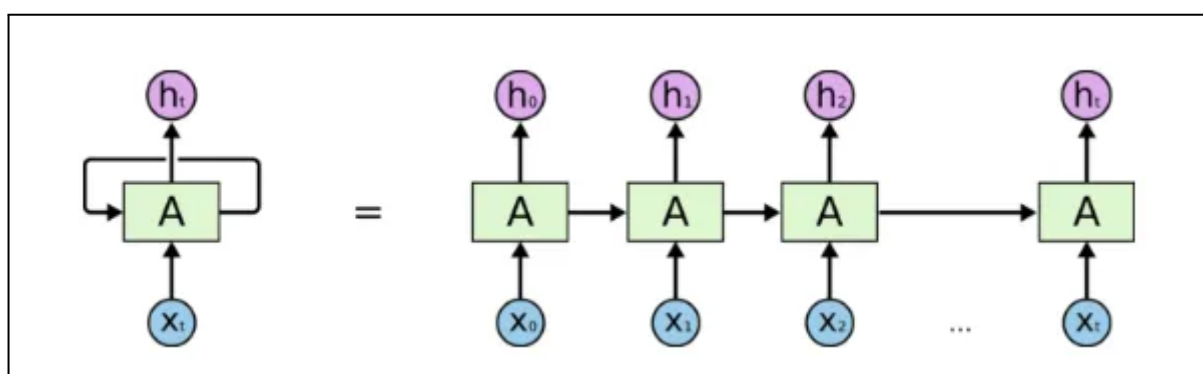


Рисунок 3.5 – Розгорнута РНМ [47]

Спочатку він бере  $X(0)$  із послідовності введення, а потім виводить  $h(0)$ , який разом із  $X(1)$  є входом для наступного кроку. Отже,  $h(0)$  і  $X(1)$  є вхідними даними для наступного кроку. Подібним чином,  $h(1)$  з наступного є вхідними даними з  $X(2)$  для наступного кроку і так далі. Таким чином вона продовжує запам'ятовувати контекст під час навчання.

Формула поточного стану така (формула 3.2):

$h_t = f(h_{t-1}, x_t)$	(3.2)
-------------------------	-------

де  $h_t$  - новий стан,

$h_{t-1}$  - попередній стан,

$x_t$  - поточний вхід.

Тепер у нас є стан попереднього введення замість самого введення, оскільки вхідний нейрон застосував би перетворення до нашого попереднього введення. Тому кожен наступний вхід називається часовим кроком.

Застосування функції активації формула 3.3:

$h_t = \tanh (W_{hh}h_{t-1} + W_{xh}x_t)$	(3.3)
---	-------

де  $W$  – вага,

$h$  – єдиний прихований вектор,

$W_{hh}$  – вага в попередньому прихованому стані,

$W_{hx}$  – вага в поточному стані введення,

$\tanh$  – функція активації, яка реалізує нелінійність, яка зміщує активації до діапазону  $[-1, 1]$ .

Тепер, коли поточний стан обчислено, ми можемо обчислити вихідний стан за формулою 3.4:

$y_t = W_{hy}h_t$	(3.4)
-------------------	-------

де  $y_t$  – вихідний стан,

$W_{hy}$  вага у вихідному стані.

### 3.3 LSTM моделі довго короткочасної пам'яті

LSTM є вдосконаленою версією звичайної РНМ, яка була розроблена, щоб полегшити фіксацію довготривалих залежностей у даних послідовності. Регулярна РНМ функціонує таким чином, що на активацію прихованого стану впливають інші локальні активації, найближчі до них, що відповідає «короткочасній пам'яті», тоді як на ваги мережі впливають обчислення, які відбуваються протягом усього довгого часу послідовностей, що відповідає «довготривалій пам'яті». Тому РНМ було перероблено таким чином, щоб вона мала стан активації, який також може діяти як ваги та зберігати інформацію на великих відстанях, звідси й назва «довго короткочасна пам'ять».

LSTM спеціально розроблені, щоб уникнути проблеми довгострокової залежності. Запам'ятовування інформації протягом тривалого часу – це практично

їхня типова поведінка, а не те, що їм важко навчитися. Архітектуру LSTM відображено на рисунку 3.6.

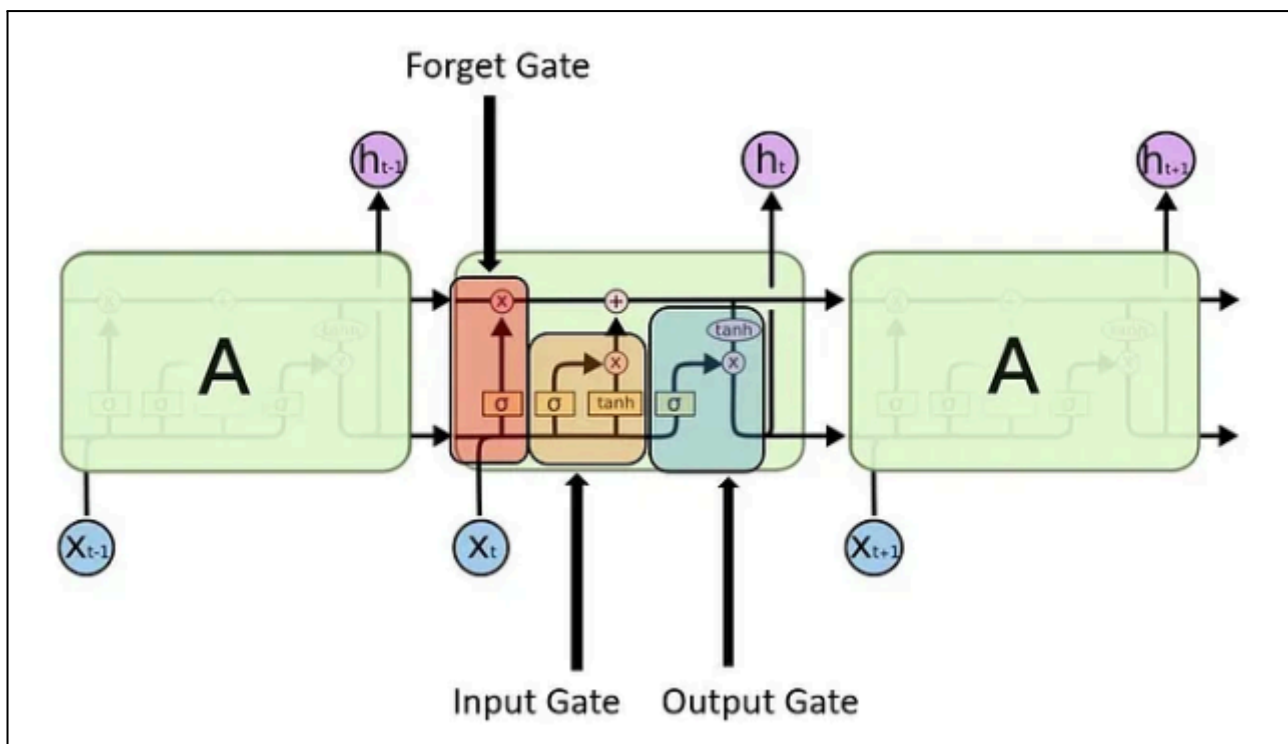


Рисунок 3.6 – Архітектура LSTM [43]

Ключем до LSTM є стан вузлу, горизонтальна лінія, що проходить через верхню частину діаграми. (див. рис. 3.7)

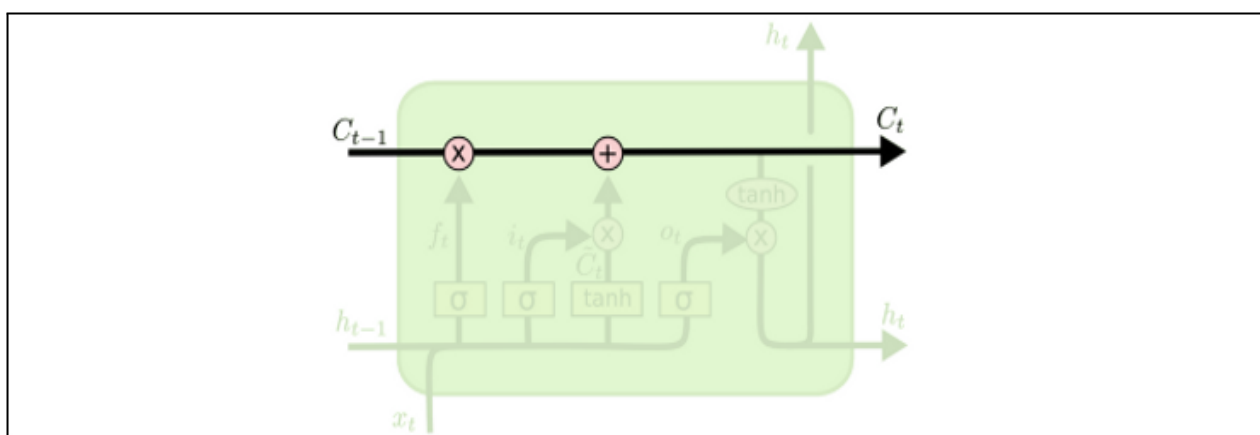


Рисунок 3.7 – Чорна лінія відображає стан вузлу [43]

Стан вузлу схожий на конвеєр. Він проходить прямо вниз по всьому ланцюгу, лише з деякими незначними лінійними взаємодіями. Інформація дуже легко і просто тече разом із ними без змін.

LSTM дійсно має здатність видаляти або додавати інформацію про стан вузлу, що ретельно регулюється структурами, які називаються брамами. Брама – це необов’язковий спосіб пропускання інформації. Вони складаються з шару сигмоподібної нейронної мережі та операції потокового множення.

Сигмоподібний шар виводить числа від нуля до одиниці, описуючи, скільки кожного компонента має пройти. Значення нуль означає «нічого не пропускати», тоді як значення один означає «пропускати все».

Якщо ви подивитесь на архітектуру LSTM, то побачите, що вона має три ворота для брами захисту та контролю стану комірки:

Шлюз забуття вирішує, яку частину минулої інформації LSTM має запам’ятати (яку інформацію пропустити на конкретному кроці зв’язку). Це рішення приймається сигмоїною функцією активації (формула 3.5).

$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$	(3.5)
---	-------

Вона переглядає попередній стан  $h_{t-1}$  і поточний вхідний сигнал  $X_t$  і виводить число від 0 (опустити цю інформацію) до 1 (зберегти цю інформацію) для кожного числа в стані комірки  $C_{t-1}$ .

$h_{t-1}$  – це прихований стан із попередньої комірки або результат попередньої комірки, а  $x_t$  – це вхід на цьому конкретному кроці часу. Дані вхідні дані множаться на вагові матриці та додається зміщення. Після цього до цього значення застосовується сигмоїдна функція. Сигмоїдна функція виводить вектор зі значеннями в діапазоні від 0 до 1, що відповідає кожному числу в стані комірки. Сигмоїдна функція відповідає за прийняття рішень, які значення зберегти, а які відкинути. Якщо «0» виводиться для певного значення в стані комірки, це означає, що шлюз забуття хоче, щоб стан комірки повністю забув цю частину інформації. Подібним чином «1» означає, що шлюз забуття хоче запам’ятати весь цей фрагмент інформації. Цей векторний вихід із сигмоїдної функції множиться на стан комірки.

Вхідний шлюз відповідає за додавання інформації до стану комірки. Це додавання інформації є триетапним процесом, як видно з діаграми вище.

Регулювання того, які значення потрібно додати до стану комірки, за допомогою сигмоїдної функції (формула 3.6).

$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$	(3.6)
---	-------

Це дуже схоже на шлюз забуття та діє як фільтр для всієї інформації з  $h_{t-1}$  та  $x_t$ .

Створення вектора, що містить усі можливі значення, які можна додати (як сприймаються з  $h_{t-1}$  і  $x_t$ ) до стану клітинки. Це робиться за допомогою функції  $\tanh$ , яка виводить значення від -1 до +1.

Множення значення регуляторного фільтра (сигмоїдний затвор) на створений вектор (функція  $\tanh$ ), а потім додавання цієї корисної інформації до стану комірки за допомогою операції додавання.

Настав час оновити старий стан клітинки,  $C_{t-1}$ , на новий стан клітинки  $C_t$ . Попередні кроки вже вирішили, що робити, нам просто потрібно це зробити.

Ми множимо старий стан на  $f_t$ , забуваючи те, що вирішили забути раніше. Потім додаємо його  $i_t \cdot \tilde{C}$ . Це нові значення кандидатів, масштабовані за тим, наскільки ми вирішили оновити кожне значення стану.

Вихідний шлюз вирішує, яка частина поточної комірки потрапляє на вихід.

Функціонування вихідного вентиля знову можна розбити на три етапи:

Створення вектора після застосування функції  $\tanh$  до стану клітинки, таким чином масштабуючи значення в діапазоні від -1 до +1.

Створення фільтра з використанням значень  $h_{t-1}$  і  $x_t$  таким чином, щоб він міг регулювати значення, які мають бути виведені з вектора, створеного вище. Цей фільтр знову використовує сигмоподібну функцію.

Помноження значення цього регуляторного фільтра на вектор, створений на кроці 1, і надсилання його як вихідних даних, а також надсилання його прихований стан наступної комірки див. у формулі 3.7.

$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh (C_t)$	(3.7)
---	-------

LSTM є дуже перспективним рішенням для проблем, пов'язаних із послідовністю та часовими рядами. Однак єдиним недоліком, який я бачу в них, є труднощі в їх навчанні. Багато часу і системних ресурсів йде на навчання навіть простої моделі.

## 4 ЕКСПЕРИМЕНТАЛЬНА СКЛАДОВА ДОСЛІДЖЕННЯ

### 4.1 Вибір криптовалют для дослідження

Перед тим як розпочати дослідження та аналіз прогнозу курсів криптовалют, необхідно правильно визначити, яка саме криптовалюта варта уваги.

Для вирішення задачі многокритеріального вибору в рамках магістерського дослідження оберемо наступні криптовалюти:

Bitcoin (BTC) – це криптовалюта, яка була створена анонімною особою або групою людей під псевдонімом Сатоші Накамото у 2008 році. Bitcoin є першою та найвідомішою криптовалютою в світі. Він базується на технології блокчейн, яка забезпечує безпеку та надійність транзакцій.

Ethereum (ETH) – це децентралізована блокчейн-платформа з відкритим вихідним кодом, створена Віталіком Бутеріном. Основною криптовалютою на цій платформі є Ether (ETH).

Solana (SOL) – це блокчейн-платформа та криптовалюта, створена з метою покращення швидкості та масштабованості транзакцій у світі криптовалют. Основною метою Solana є забезпечення високої швидкості обробки транзакцій та виконання смарт-контрактів.

MATIC (MATIC) – це криптовалюта та платформа для створення децентралізованих додатків (DApps) на блокчейні Ethereum. Основною метою MATIC є покращення швидкості та масштабованості мережі Ethereum.

Gelato (GEL) є криптовалютою, яка використовується для функціонування інфраструктури DeFi (децентралізованих фінансів). Вона отримала свою назву від італійського слова "gelato," що означає морозиво. Основною ідеєю за цією криптовалютою є забезпечення автоматизації різних операцій у децентралізованих фінансових протоколах.

Arbitrum (ARB) – це масштабний розширювальний шар для блокчейну Ethereum, який дозволяє покращити продуктивність та швидкість транзакцій на платформі Ethereum.

Виберемо множину критеріїв для задачі багатокритеріального вибору.

**Капіталізація:** Капіталізація визначається як загальна ринкова вартість криптовалюти, що показує, скільки грошей інвестори вже вклали у дану криптовалюту. Цей показник буде відображати надійність цієї криптовалюти.

**Волатильність:** Візьмемо середнє значення за останній місяць, для короткострокового прогнозування, цього буде достатньо.

**Обсяг торгів / в день (середній):** Цей критерій вказує на обсяг торгів криптовалютою протягом дня. Більший обсяг може свідчити про більшу ліквідність і довіру ринку.

**Транзакції в секунду (TPS)** – це кількість операцій, які мережа може обробляти кожну секунду.

**Історичні дані (місяців):** Для оцінки криптовалюти важливо досліджувати її історичні дані. Наявність даних за більший період надає можливість навчити модель більш тонко прогнозувати курси валют.

Для заповнення таблиці 4.1 були взяті історичні дані з сайту <https://finance.yahoo.com/> який розміщує корисні інструменти для аналізу ринку криптовалют та отримувати історичні дані у форматі CSV.

Таблиця 4.1 – Векторний опис альтернатив (Таблиця виконана самостійно)

	Капіталізація, USD	Стабільність за останній місяць, %	Обсяг торгів / в день (середній за ост місяць), USD	Кількість транзакцій в секунду	Історичні дані, міс.
<b>BTC</b>	775 821 726 845	0,10	18 047 358 995,00	7	129
<b>ETH</b>	259 300 223 439	0,09	17 068 510 656,00	30	101
<b>SOL</b>	26 815 560 650	0,00	2 090 151 459,00	65000	44
<b>MATIC</b>	7 447 803 791	0,05	580 052 071,00	65000	56
<b>ARB</b>	1 383 455 156	0,08	315 842 978,00	40000	9
<b>GEL</b>	92 420 917	0,12	243413.1935	15000	27

Зробимо використання на множині альтернатив принципу Парето. Для цього знайдемо спочатку усі недоміновані або оптимальні за Парето альтернативи. Візуалізація цього аналізу приведена в таблиці 4.2.

Таблиця 4.2 – Пошук недомінованих альтернатив за принципом Парето (Таблиця виконана самостійно)

	Капіталізація, USD	Стабільність за останній місяць, %	Обсяг торгів / в день (середній за ост місяць), USD	Кількість транзакцій в секунду	Історичні дані, міс.
<b>BTC</b>	775 821 726 845	0,10	18 047 358 995,00	7	129
<b>ETH</b>	259 300 223 439	0,09	17 068 510 656,00	30	101
<b>SOL</b>	26 815 560 650	0,00	2 090 151 459,00	65000	44
<b>MATIC</b>	7 447 803 791	0,05	580 052 071,00	65000	56
<b>ARB</b>	1 383 455 156	0,08	315 842 978,00	40000	9
<b>GEL</b>	92 420 917	0,12	243413.1935	15000	27

Виконаємо нормування за еталоном. В нашому випадку еталоном для кожної шкали виберемо максимальне значення. Та розрахуємо нормовані показники за формулою 4.1.

$F = F_{\text{ізмєр}} / F_{\text{єталон}}$	(4.1)
--	-------

де  $F_{\text{ізмєр}}$  – це оцінка, що зважується,

$F_{\text{єталон}}$  – це максимальна оцінка за критерієм.

Таким чином, отримуємо нормовані значення, що приведені у таблиці 4.3.

Таблиця 4.3 – Результат нормування оцінок за шкалами за еталоном  
(Таблиця виконана самостійно)

	Капіталізація, USD	Стабільність за останній місяць, %	Обсяг торгів / в день (середній за ост місяць), USD	Кількість транзакцій в секунду	Історичні дані, міс.
<b>BTC</b>	1,00	0,83	1,00	0,00011	1
<b>ETH</b>	0,33	0,75	0,95	0,00046	0,78
<b>SOL</b>	0,03	0,00	0,12	1,00	0,34
<b>MATIC</b>	0,01	0,42	0,03	1,00	0,43
<b>GEL</b>	0,00012	1,00	0,00001	0,23	0,21

Оскільки серед наших критеріїв для вибору альтернативи задля проведення дослідження ефективності використання інформаційних технологій на основі ШІ для прогнозування курсу криптовалюти однозначно більш вагомими є стабільність (бо робить прогноз більш точним) та історичні дані (бо дозволяють навчити

модель більш досконало), а інші є вторинними за важливістю, вибір загорткової моделі впав на лінійну адитивної згортку з ваговими коефіцієнтами.

Лінійна адитивна згортка з ваговими коефіцієнтами – це метод прийняття рішень, який використовується для порівняння альтернативних варіантів за допомогою числових коефіцієнтів (вагових коефіцієнтів), що відображають важливість кожного критерію при виборі оптимальної альтернативи.

Визначимо вагові коефіцієнти пропорційним методом. Як вже зазначено вище, для аналізу в рамках магістерського дослідження найголовнішими критеріями є стабільність та довжина історичних даних. Надамо ваги 0,35 кожному них. А вторинним критеріям – призначемо вагу 0,1 кожному. Отже маємо  $0,35 + 0,35 + 0,10 + 0,10 + 0,10 = 1,00$ , тобто загальна сума вагів дорівнює одному що задовольняє умови визначення вагових коефіцієнтів пропорційним методом.

Результат нормування оцінок за шкалами за еталоном з визначеними ваговими коефіцієнтами наведено у таблиці 4.4.

Таблиця 4.4 – Результат нормування оцінок за шкалами за еталоном з визначеними ваговими коефіцієнтами (Таблиця виконана самостійно)

	Капіталізація, USD	Стабільність за останній місяць, %	Обсяг торгів / в день (середній за ост місяць), USD	Кількість транзакцій в секунду	Історичні дані, міс.
<b>BTC</b>	1,00	0,83	1,00	0,00011	1
<b>ETH</b>	0,33	0,75	0,95	0,00046	0,78
<b>SOL</b>	0,03	0,00	0,12	1,00	0,34
<b>MATIC</b>	0,01	0,42	0,03	1,00	0,43
<b>GEL</b>	0,00012	1,00	0,00001	0,23	0,21
<b>Вага</b>	<b>0,1</b>	<b>0,35</b>	<b>0,1</b>	<b>0,1</b>	<b>0,35</b>

Перемножимо кожне значення шкали и коефіцієнтом ваги та отримаємо суму за кожної з альтернатив, що наведено у таблиці 4.5.

Таблиця 4.5 – Остаточні розрахунки (Таблиця виконана самостійно)

	Капіталі- зація, USD	Стабільність за останній місяць, %	Обсяг торгів / в день (середній за ост місяць), USD	Кількість транзакці й в секунду	Історичні дані, міс.

<b>Рез-т BTC</b>	0,10000	0,29167	0,10000	0,00001	0,35000	<b>0,84</b>
<b>Рез-т ETH</b>	0,03342	0,26250	0,09458	0,00005	0,27403	<b>0,66</b>
<b>Рез-т SOL</b>	0,00346	0,00000	0,01158	0,10000	0,11938	<b>0,23</b>
<b>Рез-т MATIC</b>	0,00096	0,14583	0,00321	0,10000	0,15194	<b>0,40</b>
<b>Рез-т GEL</b>	0,00001	0,35000	0,00000	0,02308	0,07326	<b>0,45</b>

Як ми бачимо з цього рисунку, найоптимальнішою альтернативою для дослідження є криптовалюта BTC з результатом 0,84.

#### 4.2 Збір даних

Дослідження впливу новостних настроїв на прогнозування курсу BTC потребує наявності даних зі стрічок новин. Для збору даних було обрано метод скраблінгу, тобто – це процес збирання, вилучення та очищення даних з різних джерел в інтернеті. Мета скраблінгу даних - отримати доступ до інформації, яка зазвичай не є структурованою і розподілена за у різних веб-ресурсах, а потім перевести її у структурований формат для подальшого аналізу або використання. Однак при використанні скраблінгу даних необхідно враховувати правові обмеження та політику конфіденційності, щоб уникнути порушення авторських прав або правил використання даних веб-сайтів.

В якості ресурсів новин було обрано пошуковий сайт Google <https://www.google.com/> (див. рис. 4.1) та Yahoo <https://news.search.yahoo.com/> (див. рис. 4.2).

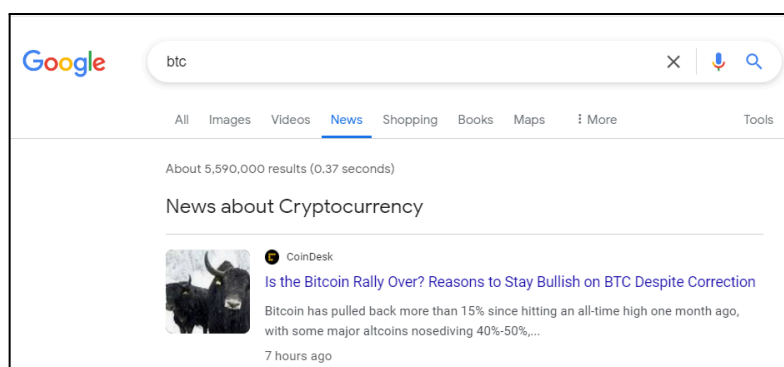


Рисунок 4.1 – Пошуковий сайт новин Google

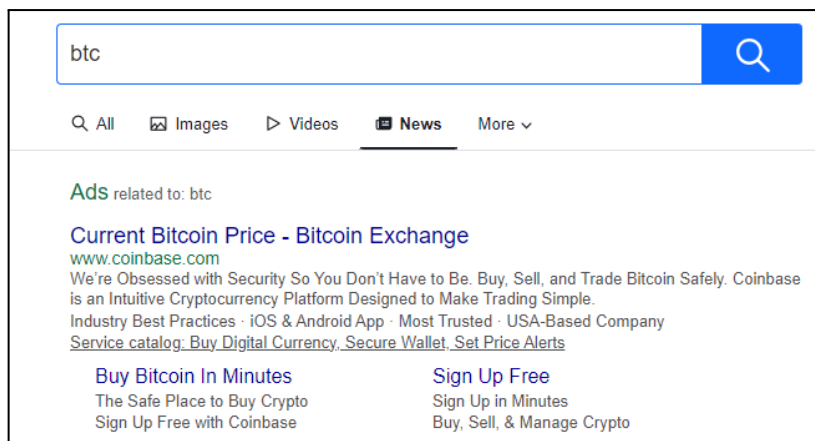


Рисунок 4.2 – Пошуковий сайт новин Yahoo

Для того, щоб автоматизувати процес скраблінгу, було розроблено 2 програми з використанням бібліотек requests, urllib, BeautifulSoup та pandas мовою python.

Було зібрано 29900 новин зі сайту Google та 24000 новин Yahoo.

Після фільтрування дублікатів отримано 3634 унікальні новини починаючи з 2014 року, року коли почалися торги BTC (див рис 4.3).

	Title	Source	Description	Link	Date
0	Bitcoin 'Halving' Due Next Year Spurs Predicti...	Bloomberg.com	Bitcoin's rebound is just the start of a rally...	https://www.bloomberg.com/news/articles/2023-0...	Apr 23, 2023
1	Crypto winter is over — and bitcoin could hit ...	CNBC	Crypto winter is over — and bitcoin could hit ...	https://www.cnn.com/2023/04/24/bitcoin-btc-pr...	Apr 24, 2023
2	Bitcoin could hit \$100000 by end-2024, Standar...	Reuters	Top cryptocurrency bitcoin could reach \$100000...	https://www.reuters.com/markets/us/bitcoin-cou...	Apr 24, 2023
3	BTC Presents "Brand New" Spring Recital	City of Pittsburg, KS	Break The Ceiling Athletics & Dance Present Th...	https://www.pittks.org/event/btc-presents-bran...	Apr 3, 2023
4	Bitcoin's Lightning Network: What It Is And Ho...	Forbes	It's a network of payment channels that allows...	https://www.forbes.com/sites/digital-assets/20...	Apr 27, 2023

Рисунок 4.3 – Приклад датасету новин (Рисунок виконаний самостійно)

Крім того було отримано дані котировок з сайту <https://finance.yahoo.com/>. Датасет було вибрано з 17 вересня 2014 по 31 березня 2024р. Приклад датасету наведено на рисунку 4.4.

	Date	Close	Volume	Change, %
0	9/17/2014	457.334015	2.105680e+07	NaN
1	9/18/2014	424.440002	3.448320e+07	-7.192558
2	9/19/2014	394.795990	3.791970e+07	-6.984264
3	9/20/2014	408.903992	3.686360e+07	3.573492
4	9/21/2014	398.821014	2.658010e+07	-2.465855
...	...	...	...	...
3479	3/27/2024	69455.343750	4.082711e+10	-0.760835
3480	3/28/2024	70744.953130	3.437490e+10	1.856746
3481	3/29/2024	69892.828130	2.523085e+10	-1.204503
3482	3/30/2024	69645.304690	1.713024e+10	-0.354147
3483	3/31/2024	71333.648440	2.005094e+10	2.424203

Рисунок 4.4 – Приклад датасету BTC (Рисунок виконаний самостійно)

### 4.3 Підготовка даних

Обробка тексту в галузі NLP охоплює широкий спектр завдань та методів для роботи з текстовими даними.

Нижче наведено кілька основних етапів обробки тексту, що було виконано для проведення дослідження з використанням сентиментів.

Спочатку було об'єднано вміст колонок Title та Description в колонку Text з метою спрощення оцінки сентиментів.

Далі було видалено усі спеціальні символи та цифри, залишаючи тільки символи алфавіту.

Також було виконано приведення до нижнього регістру.

Видалення стоп-слів - це базова техніка обробки природної мови (NLP), яка полягає у виключенні з найпоширеніших і неінформативних слів. Було використано модуль stopwords бібліотеки NLTK для виконання цієї процедури.

Лематизація - це процес приведення словоформи до її базової чи нормальної форми, що називається лемою. Лема являє собою канонічну форму слова, яка є його словниковою формою або основою. Наприклад, лемою для слова "бігав" буде слово "бігати", для "кішок" - "кішка".

Лематизація у обробці тексту широко використовується для стандартизації тексту перед його аналізом чи обробкою. Це дозволяє знизити розмірність

текстових даних, зменшити кількість різних словоформ, покращити точність аналізу тексту та отримання ознак.

Процес лематизації включає видалення закінчень і суфіксів слова, щоб отримати його базову форму. Однак, на відміну від стеммінгу, лематизація враховує морфологічну інформацію та зазвичай використовує словник або граматичні правила для визначення леми. Це дозволяє уникнути створення некоректних слів, які могли б з'явитися під час стеммінгу.

Для виконання лематизації було використано модуль `WordNetLemmatizer` бібліотеки `NLTK`.

Нормалізація даних - це процес приведення даних до стандартного формату або діапазону, щоб покращити їхню узгодженість та обробку. У контексті машинного навчання та аналізу даних нормалізація зазвичай означає масштабування числових ознак таким чином, щоб вони перебували у певному діапазоні або мали стандартний розподіл.

Найбільш поширеним методом нормалізації даних є Мін-Макс нормалізація (`Min-Max Scaling`). Вона перетворює значення ознак до заданого діапазону, зазвичай від 0 до 1, шляхом віднімання мінімального значення та поділу на різницю між максимальним та мінімальним значеннями.

Нормалізація даних допомагає алгоритмам машинного навчання працювати ефективніше та покращує їхню збіжність, особливо при використанні алгоритмів, таких як градієнтний спуск, які чутливі до масштабування ознак.

Саме таку нормалізацію було проведено над датасетами у ході виконання дослідження за допомогою модулю `MinMaxScaler` з бібліотеки `sklearn`.

#### 4.4 Аналіз настроїв

Аналіз настроїв – це процес визначення тональності чи емоційного забарвлення тексту. Метою аналізу настроїв є визначення того, чи є текст позитивним, негативним чи нейтральним. Цей процес є важливим у багатьох областях, таких як соціальні медіа, маркетинг, огляди продуктів, фінансова аналітика та інші.

Для аналізу sentimentів використовуються різні методи та інструменти:

Правила та евристика. Прості правила та евристика можуть використовуватись для визначення тональності тексту на основі ключових слів, виразів або шаблонів. Наприклад, поява слів "добре", "погано", "дивовижно" може вказувати на позитивний або негативний відгук.

Машинне навчання. Методи машинного навчання, такі як класифікація тексту, використовуються автоматичної класифікації текстових даних на позитивні, негативні чи нейтральні. Для цього можуть застосовуватись різні алгоритми, включаючи наївний Байєсівський класифікатор, метод опорних векторів (SVM), випадковий ліс та нейронні мережі.

Глибоке навчання. З розвитком глибокого навчання стали з'являтися моделі, здатні ефективно обробляти природну мову та аналізувати sentimentи. Прикладами таких моделей є рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) та трансформери, такі як BERT.

Словникові методи: Цей метод ґрунтується на використанні словників із позитивними та негативними словами для визначення тональності тексту. Методи, такі як аналіз настроїв лексикону (Lexicon-based sentiment analysis), спираються такі словники і можуть застосовувати правила обчислення загальної тональності тексту.

Гібридні підходи: У деяких випадках комбінуються різні методи та інструменти для покращення якості аналізу sentimentів. Наприклад, можна використовувати словникові методи для попереднього визначення тональності тексту, а потім застосовувати методи машинного навчання для більш точної класифікації.

Бібліотека vaderSentiment (Valence Aware Dictionary and sEntiment Reasoner) – це інструмент аналізу sentimentів у тексті, розроблений компанією Georgia Tech, що спеціалізується на NLP.

Основою vaderSentiment є словник, що містить тисячі слів з відповідними оцінками тональності (позитивної, негативної або нейтральної). Ці оцінки було створено експертами з урахуванням їх сприйняття sentimentів слів. Принцип

роботи бібліотеки vaderSentiment заснований на аналізі тональності тексту з використанням лексикону, який містить тисячі слів з відповідними оцінками позитивної, негативної або нейтральної тональності. Загалом принцип роботи vaderSentiment базується на простих лексичних правилах та обліку інтенсивності та контексту слів, що дозволяє йому швидко та ефективно аналізувати sentimenti в тексті.

Наступною обраною бібліотекою для аналізу sentimentів було обрано TextBlob. TextBlob є інструментом для обробки тексту мовою Python, який надає простий і зручний інтерфейс для виконання різних завдань NLP

Розробка бібліотеки TextBlob була пов'язана з необхідністю створення інструменту для виконання базових завдань обробки тексту з мінімальними зусиллями. Вона була розроблена Шоном Ріверсом (Sean Robertson) і складається з простого інтерфейсу, що дозволяє швидко приступити до роботи з текстовими даними без необхідності глибоких знань обробки природної мови

TextBlob побудований на основі бібліотек NLTK (Natural Language Toolkit) та Pattern, що дозволяє використовувати їх можливості для вирішення різних завдань обробки тексту. Бібліотека надає широкий спектр функцій для роботи з текстом, включаючи виділення ключових слів, визначення мови тексту, вилучення фраз, аналіз тональності та багато іншого. Принцип роботи функції аналізу sentimentу в бібліотеці TextBlob ґрунтується на аналізі тональності тексту з використанням попередньо навченої моделі на основі правил та машинного навчання.

Наступним інструментом було обрано бібліотеку BERT (Bidirectional Encoder Representations from Transformers) – це модель для представлення тексту, розроблена компанією Google у 2018 році. Вона є трансформерною моделлю, яка навчається на великих обсягах текстових даних і здатна генерувати контекстуалізовані уявлення слів. Отримані результати класифікації використовують для оцінки тональності тексту. Залежно від завдання, результати можуть бути представлені у вигляді числових оцінок, ймовірностей чи міток класів.

Ще одним інструментом було обрано бібліотеку Flair для Python. Вона є інструментом NLP) роботи з текстовими даними. Flair розроблена в компанії Zalando Research і вперше випущена у 2018 році. Вона є однією з провідних бібліотек для обробки природної мови та активно підтримується та розвивається спільнотою дослідників та розробників.

Принцип роботи Flair Sentiment ґрунтується на використанні контекстуалізованих ембеддингів слів, отриманих за допомогою мовної моделі Flair, для аналізу тональності тексту. Залежно від завдання, результати можуть бути представлені у вигляді числових оцінок, ймовірностей чи міток класів.

Бібліотека spaCy – це відкрита бібліотека NLP мовою Python. Вона надає простий та ефективний спосіб виконання різних завдань обробки тексту, таких як токенізація, розмітка частин мови, синтаксичний аналіз, виділення іменованих сутностей та векторне представлення тексту. spaCy була розроблена в 2015 році Метью Хонзалом (Matthew Honnibal) та Йорґеном Ван Дер Вальтом (Jorgensen van der Walt) у рамках їхньої роботи в компанії Explosion AI. Це було зроблено у відповідь на брак ефективних та простих у використанні інструментів для обробки тексту на Python.

Принцип роботи spaCy можна розбити на кілька ключових кроків:

Попередня обробка тексту: spaCy приймає на вхід текст і виконує попередню обробку, включаючи токенізацію (розподіл тексту на окремі слова або токени), видалення стоп-слів, лематизацію (приведення слів до їх базової форми) та визначення частин мови.

Синтаксичний аналіз: Після попередньої обробки spaCy аналізує структуру речень та зв'язку між словами, будуючи синтаксичне дерево залежностей. Це дозволяє зрозуміти граматичну структуру тексту та виділити ключові елементи, такі як сутності та їхні стосунки.

Виділення сутностей: spaCy виявляє та класифікує іменовані сутності в тексті, такі як імена людей, організації, розташування, дати та числа. Це допомагає зрозуміти про що йдеться в тексті та отримати важливу інформацію.

Морфологічний аналіз: spaCy також проводить морфологічний аналіз тексту, визначаючи форму слова, його частину мови, відмінок, число та інші граматичні характеристики.

Векторне уявлення: Однією з особливостей spaCy є можливість використання векторних моделей, що передбачаються, для представлення слів і текстових фрагментів у вигляді числових векторів. Це дозволяє ефективно працювати з текстовими даними та застосовувати їх у різних завданнях машинного навчання та аналізу тексту.

Таким чином, spaCy відома своєю високою продуктивністю та оптимізованими алгоритмами, що дозволяє обробляти великі обсяги тексту швидко та ефективно та є потужним інструментом для обробки та аналізу текстових даних, що володіє широкими можливостями і високою продуктивністю.

#### 4.5 Підготовка датасетів

Для наглядності дослідження було вирішено використовувати одну модель на різних датасетах. Задля цього було проведено оцінку новосних сентиментів для кожного з вибраних інструментів аналізу (Vader, TextBlob, Bert, Flair та Spacy) та отримано 2 оцінки – полярність та сентимент.

Полярність зазвичай відноситься до міри тональності чи емоційного забарвлення тексту. Вона визначає, наскільки текст висловлює позитивні, негативні чи нейтральні емоції чи думки.

Сентимент відображає округлене значення полярності, що визначається наступною функцією до якої передається значення полярності:

```
def analysis(score):  
    if score < 0:  
        return -1  
    elif score == 0:  
        return 0  
    else:  
        return 1
```

Рисунок 4.5 – Функція отримання сентименту (Рисунок виконаний самостійно)

У результаті отримано датасети на чотирьох часових інтервалах – з самого початку – з 2014 року, з 2021 року, з 2023 року та з початку 2024 по 31 березня 2024 року за кожним з обраних інструментів – що загалом становить 20 різних датасетів.

Формат отриманого в результаті датасету дивись на рисунку 4.6

Title	Source	Description	Link	Date	Text	Polarity	Sentiment
Venture capitalist Draper wins U.S. bitcoin au...	Reuters	Venture capitalist Tim Draper, sole winner of ...	<a href="https://www.reuters.com/article/idUSKBN0F7199/">https://www.reuters.com/article/idUSKBN0F7199/</a>	7/2/2014	venture capitalist draper win u bitcoin auctio...	0.8979	1

Рисунок 4.6 – Датасет після оцінки сентименту (Рисунок виконано самостійно)

Після цього датасети з сентиментами було згруповано за датою, при цьому агрегуюча функція була використана с обчисленням середньої зваженої та з отриманням максимального показника. Тим самим кількість датасетів було збільшено вдвічі (див. рис. 4.7 – 4.8).

Date	Close	Volume	Polarity	Sentiment
1/1/2024	44167.33203	1.842698e+10	0.5423	1.0
1/2/2024	44957.96875	3.933527e+10	0.8591	1.0
1/3/2024	42848.17578	4.634232e+10	0.6597	1.0
1/4/2024	44179.92188	3.044809e+10	0.8271	1.0
1/5/2024	44162.69141	3.233603e+10	0.8779	1.0

Рисунок 4.7 – Результуючий датасет після мержу об'єднання за максимальним показником (Рисунок виконаний самостійно).

Date	Close	Volume	Polarity	Sentiment
1/1/2024	44167.33203	1.842698e+10	0.372300	1.000000
1/2/2024	44957.96875	3.933527e+10	0.330760	0.600000
1/3/2024	42848.17578	4.634232e+10	0.110233	-0.333333
1/4/2024	44179.92188	3.044809e+10	0.449333	0.666667
1/5/2024	44162.69141	3.233603e+10	0.590300	1.000000

Рисунок 4.8 – Результуючий датасет після об'єднання за середньозваженими показниками (Рисунок виконаний самостійно)

Також було отримано суму новин за кожною датою та змержено з датасетом BTC та для наглядності було побудовано графік ціни та обсягу торгів за весь період (див. рис. 4.9)



Рисунок 4.9 – Ціна BTC та обсяг торгів з 2014 року (Рисунок виконано самостійно)

Та побудовано графік ціни BTC та кількості новин (див. рис. 4.10)

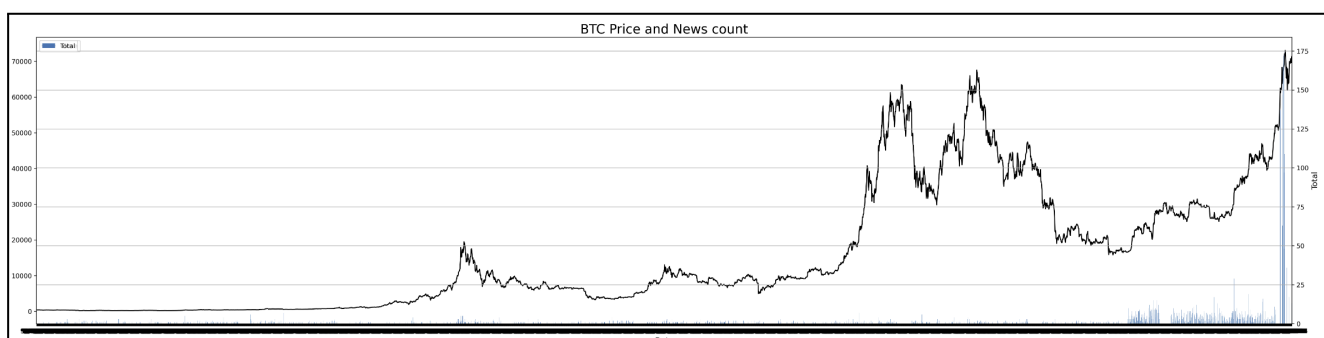


Рисунок 4.10 – ціна BTC та кількість новин з 2014 року (Рисунок виконано самостійно)

Можемо зробити висновок, що обсяг торгів розподілений досить рівномірно, а кількість отриманих новин різко зростає після 2023 року, а у березні 2024 має нерівномірний сплеск. Взагалі ми отримали 40 різних датасетів – 5 різних інструментів з 2-ми типами показників оцінки настроїв (максимальні та середньозважені) а також на 4-х різних часових інтервалах.

#### 4.6 План проведення експерименту

Для проведення експерименту було використано середовище – фізичний пристрій на базі Windows. Технічні характеристики даного пристрою наведені у таблиці 4.6.

Таблиця 4.6 – Характеристики пристрою (Таблиця виконана самостійно)

Характеристика	Значення
Name	Dell XPS 15 9520
CPU	12th Gen Intel(R) Core(TM) i9-12900HK 2.50 GHz
RAM	64 GB: 2 x 32 GB, DDR5, 4800 MT/s, dual-channel
OS	Windows
SDD	1 TB, M.2, PCIe NVMe, SSD

Для реалізації обраних моделей була обрана мова програмування Python версії 3.11.5 та середовище розробки Jupyter Notebook версії 6.5.4. Також буде необхідно використовувати наступні бібліотеки:

- pandas 2.0.3;
- requests 2.3.0;
- urllib 3.11.5;
- BeautifulSoup 4.12.2;
- nltk 3.8.1;
- vaderSentiment 3.3.2;
- textblob 0.15.3;
- transformers 4.32.1 з bert-large-uncased моделлю;
- flair 0.13.1;
- spacy 3.7.4;

- numpy 1.24.3;
- matplotlib 3.7.2;
- sklearn 1.4.1.post1;
- keras 2.15.0;
- scipy 1.11.1.

Для проведення експерименту числові дані повинні бути зібрані для всіх метрик кожного методу:

MSE – або середньоквадратична помилка. Це метрика, яка використовується для вимірювання точності прогнозування. Чим нижче значення MSE, тим краще точність прогнозування.

MRSE – або середнє квадратичне кореневе відхилення. Ця метрика є модифікацією MSE, відрізняється від MSE тим, що вона виражається в одиницях вимірювання вихідної змінної, а не в квадратних одиницях.

MAE – або середня абсолютна помилка. Чим нижче значення MAE, тим краще точність прогнозування.

## 5 РЕАЛІЗАЦІЯ МОДЕЛЕЙ

### 5.1 Аналіз настроїв за допомогою TextBlob

Для отримання оцінок настроїв за допомогою TextBlob скористаємось бібліотекою textblob. Для встановлення та імпорту бібліотеки виконаємо наступний код (див. рис 5.1.)

```
!pip install -U textblob
!python -m textblob.download_corpora
from textblob import TextBlob
```

Рисунок 5.1 – Підготовка TextBlob до використання (Рисунок виконаний самостійно)

Після встановлення необхідних залежностей та імпорту бібліотеки, скористуємось наступною конструкцією для отримання оцінки полярності (див. рис. 5.2)

```
def getPolarity(review):
    return TextBlob(review).sentiment.polarity
```

Рисунок 5.2 – Отримання оцінки полярності тексту використовуючи TextBlob (Рисунок виконаний самостійно)

### 5.2 Аналіз настроїв за допомогою Vader

Для початку роботи з бібліотекою Vader необхідно встановити усі залежності (див. рис. 5.3)

```
!pip install vaderSentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

Рисунок 5.3 – Підготовка Vader до використання (Рисунок виконаний самостійно)

Після цього скористаємось наступною конструкцією для отримання оцінки полярності настроїв (див. рис. 5.4)

```
analyzer = SentimentIntensityAnalyzer()
text = data['Text']
data['Polarity'] = [analyzer.polarity_scores(x)['compound'] for x in text]
```

Рисунок 5.4 – Отримання оцінки полярності тексту використовуючи Vader  
(Рисунок виконаний самостійно)

### 5.3 Аналіз настроїв за допомогою Flair

Наведемо програмну реалізацію підготовки flair до використання (див. рис.5.5)

```
!pip install flair
from flair.models import TextClassifier
from flair.data import Sentence
classifier = TextClassifier.load('en-sentiment')
```

Рисунок 5.5 – Підготовка Flair до використання (Рисунок виконаний самостійно)

Після цього скористаємось наступною функцією для отримання оцінки полярності (див. рис. 5.6)

```
def getPolarity(review):
    sentence = Sentence(review)
    classifier.predict(sentence)
    if sentence.labels[0].value == 'POSITIVE':
        return 1 * sentence.labels[0].score
    elif sentence.labels[0].value == 'NEGATIVE':
        return -1 * sentence.labels[0].score
    else:
        return 0
```

Рисунок 5.6 – Отримання оцінки полярності тексту використовуючи Flair (Рисунок виконаний самостійно)

### 5.4 Аналіз настроїв за допомогою Spacy

Для підготовки бібліотеки spacy до використання після встановлення та імпорту необхідних залежностей, треба ініціалізувати та об'єкт nlp (див. рис. 5.7)

```
!pip install spacytextblob
!python -m textblob.download_corpora
!python -m spacy download en_core_web_sm
import spacy
from spacytextblob.spacytextblob import SpacyTextBlob
nlp = spacy.load('en_core_web_sm')
nlp.add_pipe("spacytextblob")
```

Рисунок 5.7 – Підготовка Spacy до використання (Рисунок виконаний самостійно)

Скористаємось наступним методом задля отримання оцінки полярності (див. рис. 5.8)

```
def getPolarity(review):
    return nlp(review)._blob.polarity
```

Рисунок 5.8 – Отримання оцінки полярності тексту використовуючи Spacy  
(Рисунок виконаний самостійно)

### 5.5 Аналіз настроїв за допомогою Bert

Для початку роботи з Bert після встановлення модулю transformers та імпорту бібліотеки pipeline, налаштуємо модель bert-large-uncased на завдання sentiment-analysis (див. рис. 5.9)

```
!pip install transformers
from transformers import pipeline
sentiment_pipeline = pipeline(model="bert-large-uncased", task="sentiment-analysis")
```

Рисунок 5.9 – Підготовка Bert до використання (Рисунок виконаний самостійно)

Після його використаємо наступний метод для визначення оцінки настрою.

```
def getPolarity(review):
    sentiment = sentiment_pipeline(review)
    if sentiment[0]['label'] == 'LABEL_1':
        return 1 * sentiment[0]['score']
    elif sentiment[0]['label'] == 'LABEL_0':
        return -1 * sentiment[0]['score']
    else:
        return 0
```

Рисунок 5.10 – Отримання оцінки полярності тексту використовуючи Bert  
(Рисунок виконаний самостійно)

## 5.6 Модель прогнозування ціни

Додамо усі необхідні залежності для побудови моделі (див. рис. 5.11)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from os.path import join

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import root_mean_squared_error, mean_squared_error, mean_absolute_percentage_error

from keras.models import Sequential, load_model
from keras.layers import Dense, LSTM
from scipy.ndimage import shift
```

Рисунок 5.11 – Додавання залежностей для моделі прогнозування ціни (Рисунок виконаний самостійно)

Задамо необхідні параметри для побудови моделі (див. рис. 5.12)

```
look_back = 5
num_units = 256
epochs=80
batch_size=128
train_test_split = 0.80
```

Рисунок 5.12 – Параметри налаштування (Рисунок виконаний самостійно)

Налаштування конфігурацій моделі. Наша модель використовує один шар LSTM, оскільки він дає найкращий результат.

Look\_back встановлено на 5 днів, а дані розділені на 80% / 20% тренування та тестування. Look\_back - це параметр, який визначає кількість попередніх разів, які слід використовувати як вхідну змінну для прогнозування наступного періоду, в нашому випадку це 5 днів.

Крок `epochs` – це коли всі навчальні дані використовуються одночасно та визначається як загальна кількість ітерацій усіх навчальних даних в одному циклі для навчання моделі машинного навчання, задамо 80, це збільшить навантаження для тренування, але ми можемо це собі дозволити.

Параметр `batch_size` в моделі LSTM в бібліотеці Keras визначає кількість зразків, які одночасно використовуються для навчання в одній ітерації. Використання пакетних (батчевих) операцій дозволяє прискорити процес навчання та оптимізує використання ресурсів обчислювального пристрою, особливо при роботі з великими наборами даних. Зазвичай використовують значення, кратні 2 або 32, 64, 128 тощо. Встановимо 128.

Параметр `num_units` в моделі визначає кількість нейронів (одиниць) у шарі LSTM. Цей параметр контролює складність моделі і може впливати на її здатність до вивчення складних залежностей у вхідних даних. Встановимо 256, бо нам дозволяє фізичний пристрій.

Розділимо набір даних на навчальний та тестувальний та створимо вікна (див. рис. 5.13)

```
train_set, test_set = np.split(dataset, [int(train_test_split * len(dataset))])
x_train, y_train = create_dataset(train_set, pred_col, look_back=look_back)
x_test, y_test = create_dataset(test_set, pred_col, look_back=look_back)

x_train = np.reshape(x_train, (x_train.shape[0], look_back, num_of_features))
x_test = np.reshape(x_test, (x_test.shape[0], look_back, num_of_features))
```

Рисунок 5.13 – Розділення датасету (Рисунок виконаний самостійно)

У моделі `Sequential` з бібліотеки Keras `Dense` шар використовується для додавання повністю зв'язаного (повністю з'єднаного) шару до нейронної мережі. `Dense` шар складається з набору нейронів, кожен з яких пов'язаний з кожним нейроном попереднього шару. Кожен нейрон у `Dense` шарі виконує лінійну комбінацію вхідних даних та їх ваг, після чого застосовується активаційна функція для визначення виходу.

Основна ідея `Dense` шарів полягає в тому, щоб нейронна мережа могла вивчити складні нелінійні залежності у вхідних даних, обчислюючи ваги для

кожного з'єднання між нейронами. Це дозволяє моделі вирішувати різноманітні завдання, включаючи класифікацію, прогнозування та генерацію.

Він також використовує оптимізатор Adam і mean\_squared\_error для обчислення втрат.

Наведений нижче код створює модель. Має шар LSTM і шар Dense (див. рис. 5.14)

```
model = Sequential()
model.add(LSTM(num_units, input_shape=(look_back,num_of_features)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
history= model.fit(x_train, y_train,validation_split=0.30, epochs=epochs, batch_size=batch_size,shuffle=False)
```

Рисунок 5.14 - Створення моделі (Рисунок виконаний самостійно)

Отримаємо прогнозовані результати та розрахуємо метрики (див. рис. 5.15)

```
trainPredict = model.predict(x_train)
testPredict = model.predict(x_test)

train_MSE = mean_squared_error(y_train, trainPredict)
test_MSE = mean_squared_error(y_test, testPredict)

train_RMSE = np.sqrt(mean_squared_error(y_train, trainPredict))
test_RMSE = np.sqrt(mean_squared_error(y_test, testPredict))

train_MAE = mean_absolute_error(y_train, trainPredict)
test_MAE = mean_absolute_error(y_test, testPredict)
```

Рисунок 5.15 – Код прогнозування моделі та отримання метрик (Рисунок виконаний самостійно)

Для розрахунку метрик MSE, RMSE та MAE використовувалася бібліотека sklearn.

## 6 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

### 6.1 Навчання з використанням датасету TextBlob

Запустимо нашу модель для датасету TextBlob з агрегованими оцінками сентиментів за максимумом з початку 2014 року та побудуємо графік loss функції (див. рис. 6.1) для оцінки якості навчання моделі.

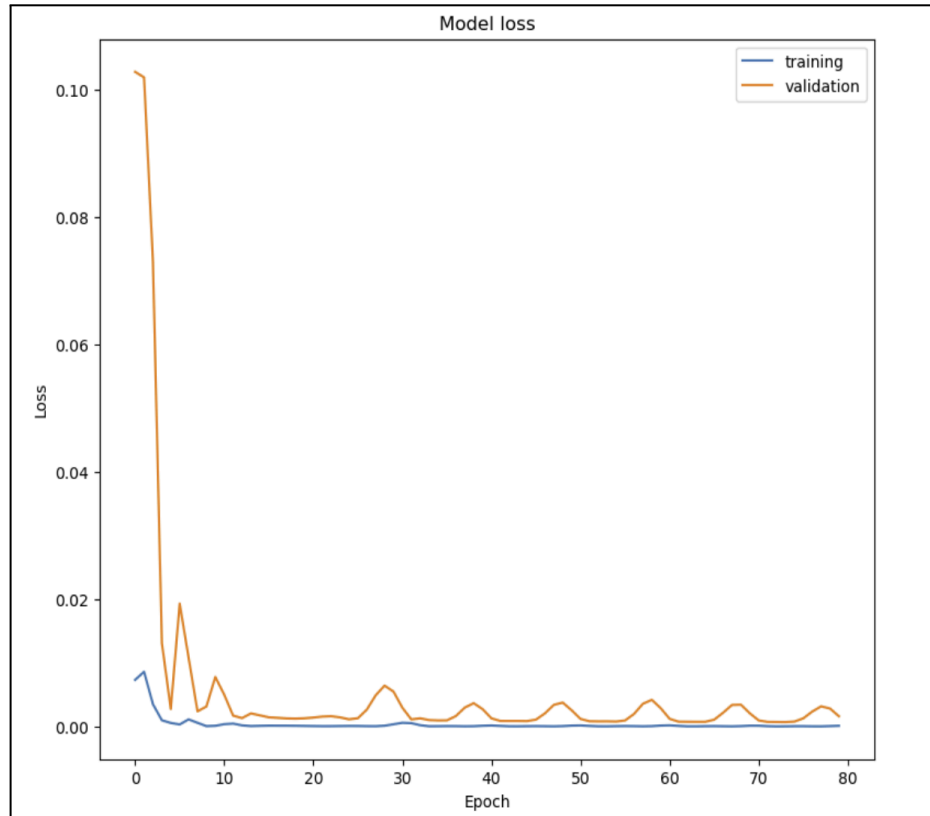


Рисунок 6.1 – Графік loss для моделі датасету TextBlob max з 2014 року (Рисунок виконаний самостійно)

Як ми бачимо, значення loss функції для тренувального датасету не перевищує 0.005, для тестового в середньому не більше 0.02. Аналізуючи графік, можна побачити, що значення функції для тренувального та тестового датасетів зменшуються до певної точки стабільності та мають невеликий розрив [2]. Виходячи з цього, можна зробити висновок, що модель навчена добре.

Побудуємо графік прогнозу ціни BTC та порівняємо з даними з тестового датасету (див. рис. 6.2). Прогнозовану ціну зображено зеленим кольором на графіку, справжню – чорним.



Рисунок 6.2 – Графік прогнозу ціни BTC датасету TextBlob max з 2014 року  
(Рисунок виконаний самостійно)

Навчання моделі на фізичному пристрої, обраному для експерименту зайняло 163 секунди. Отримано наступні метрики: MSE - 0.00063, RMSE – 0.02505 та MAE – 0.1801.

Також було проведено експеримент для датасетів з 2021, 2023 та 2023 років для середньозваженої та максимальної поляризації після агрегування даних. Усього було використано 8 датасетів для дослідження прогнозу ціни BTC за допомогою аналізу настроїв бібліотекою TextBlob та отримано наступні результати (див. табл. 6.1)

Таблиця 6.1 – Результати експерименту на базі датасетів TextBlob (Таблиця виконана самостійно).

		Датасет 2014		Датасет 2021		Датасет 2023		Датасет 2024	
		max	ave	max	ave	max	ave	max	ave
MSE	train	0.00063	0.00033	0.00107	0.00121	0.00101	0.00061	0.01093	0.01374
	test	0.00087	0.00055	0.00092	0.00123	0.02594	0.00894	0.09253	0.15626
RMSE	train	0.02505	0.01828	0.03277	0.03479	0.03185	0.02469	0.10453	0.1172
	test	0.02957	0.02335	0.0304	0.03504	0.16107	0.09456	0.30419	0.3953
MAE	train	0.01801	0.01268	0.02383	0.02493	0.02321	0.14437	0.06172	0.06626
	test	0.0179	0.01561	0.02076	0.02417	0.14437	0.01781	0.30369	0.3949

Як ми бачимо з цієї таблиці, найкраще за все модель навчається на даних більшої довжини [48]. Також ми бачимо, що середньозважені показники

полярності більш точно передають оцінку настроїв та дають кращий результат прогнозу.

Отримано майже в 2 разі кращі результати ніж за максимальними оцінками за той самий період, а саме: MSE - 0.00033, RMSE – 0.01828 та MAE – 0.1268 для тренувального датасету та MSE - 0.00055, RMSE – 0.02335 та MAE – 0.01561 для тестового.

## 6.2 Навчання з використанням датасету Vader

Запустимо нашу модель для датасетів, що їх було підготовлено з використанням бібліотеки Vader для аналізу настроїв. Усього маємо 8 датасетів, а саме 4 інтервали часу та за кожним отримуємо прогноз за середньо-зваженими та максимальним показниками поляризації. Отримані результати наведено в таблиці 6.2.

Таблиця 6.2 – Результати експерименту на базі датасетів Vader (Таблиця виконана самостійно).

		Датасет 2014		Датасет 2021		Датасет 2023		Датасет 2024	
		max	ave	max	ave	max	ave	max	ave
MSE	train	0.00035	<i>0.00034</i>	0.00154	0.00112	0.00052	0.00047	0.00699	0.00739
	test	0.00046	<i>0.00042</i>	0.00296	0.0013	0.00411	0.00585	0.05842	0.05646
RMSE	train	0.02505	<i>0.01853</i>	0.03926	0.03339	0.02276	0.02168	0.08362	0.08599
	test	0.02957	<i>0.02046</i>	0.05438	0.0361	0.06411	0.07646	0.2417	0.23762
MAE	train	0.01414	<i>0.01342</i>	0.03055	0.02376	0.01665	0.04998	0.05092	0.0525
	test	0.01307	<i>0.01274</i>	0.04725	0.02482	0.04998	0.01611	0.24112	0.23683

Як ми бачимо з цієї таблиці найкраще за все модель навчається так само на даних більшої довжини. Також ми бачимо, що середньозважені показники полярності також більш тонко передають оцінку настроїв та дають кращий результат прогнозу.

Отримано майже найкращі з результатів за середньозваженими оцінками за той самий період, а саме: MSE - 0.00034, RMSE – 0.01853 та MAE – 0.1342 для

тренувального датасету та MSE - 0.00042, RMSE – 0.02046 та MAE – 0.01237 для тестового.

### 6.3 Навчання з використанням датасету Flair

Запустимо нашу модель для датасетів, що було підготовлено з використанням бібліотеки Flair для аналізу сентиментів. Так само маємо 8 датасетів, а саме 4 інтервали часу (з даними з початку 2014, з 2021, 2023 та 2024 років відповідно) та за кожним отримаємо прогноз за показниками поляризації. Отримані результати наведено в таблиці 6.3.

Таблиця 6.3 – Результати експерименту на базі датасетів Flair (Таблиця виконана самостійно)

		Датасет 2014		Датасет 2021		Датасет 2023		Датасет 2024	
		max	ave	max	ave	max	ave	max	ave
MSE	train	0.00054	<i>0.00052</i>	0.00117	0.00114	0.00064	0.00046	0.01127	0.00767
	test	0.00121	<i>0.001</i>	0.00133	0.00159	0.01204	0.00539	0.06982	0.06097
RMSE	train	0.02505	<i>0.02289</i>	0.0342	0.03382	0.02521	0.02143	0.10616	0.08756
	test	0.02957	<i>0.03165</i>	0.03652	0.03988	0.10973	0.07341	0.26423	0.24693
MAE	train	0.0142	<i>0.01414</i>	0.02485	0.02447	0.01808	0.09557	0.06938	0.05536
	test	0.02665	<i>0.0229</i>	0.02502	0.02935	0.09557	0.01554	0.26389	0.24646

Як ми бачимо з цієї таблиці, у нас виявлено такий самий тренд - найкраще всього модель навчається на даних більшої довжини. І також ми бачимо, що середньозважені показники поляризації також більш точно передають оцінку сентиментів та дають кращий результат прогнозу. Отримано майже найкращі результати за середньозваженими оцінками за той самий період, а саме: MSE - 0.00052, RMSE – 0.02289 та MAE – 0.01414 для тренувального датасету та MSE - 0.001, RMSE – 0.03165 та MAE – 0.0229 для тестового.

#### 6.4 Навчання з використанням датасету Spacy

Запустимо нашу модель для датасетів, що було підготовлено з використанням бібліотеки Spacy що було підготовлено за такої ж саме логікою, та отримуємо результати у таблиці 6.4.

Таблиця 6.4 – Результати експерименту на базі датасетів Spacy (Таблиця виконана самостійно).

		Датасет 2014		Датасет 2021		Датасет 2023		Датасет 2024	
		max	ave	max	ave	max	ave	max	ave
MSE	train	0.00056	<i>0.00024</i>	0.00113	0.00109	0.0007	0.0006	0.01019	0.01477
	test	0.00077	<i>0.00037</i>	0.00107	0.00109	0.01951	0.00926	0.0707	0.17658
RMS E	train	0.02505	<i>0.01544</i>	0.03369	0.03304	0.02648	0.02453	0.10094	0.12152
	test	0.02957	<i>0.01933</i>	0.0327	0.03298	0.13969	0.09624	0.26589	0.42021
MAE	train	0.01731	<i>0.00785</i>	0.02427	0.02353	0.01949	0.12352	0.05957	0.06796
	test	0.01672	<i>0.0125</i>	0.02224	0.02261	0.12352	0.01776	0.26547	0.41976

Як ми бачимо з цієї таблиці в нас отримано такий самий тренд - найкраще всього модель навчається на даних більшої довжини. І також середньо взважені показники полярності більш тонко передають оцінку настроїв та дають кращий результат прогнозу.

Отримано найкращі результати за середньозваженими оцінками за той самий період, а саме: MSE - 0.00024, RMSE – 0.01544 та MAE – 0.00785 для тренувального датасету та MSE - 0.00037, RMSE – 0.01933 та MAE – 0.0125 для тестового.

#### 6.5 Навчання з використанням датасету Bert

Застосуємо такий саме підхід до датасетів Bert та отримуємо результати у таблиці 6.5.

Таблиця 6.5 – Результати експерименту на базі датасетів Bert (Таблиця виконана самостійно).

		Датасет 2014		Датасет 2021		Датасет 2023		Датасет 2024	
		max	ave	max	ave	max	ave	max	ave
MSE	train	0.00047	<i>0.00046</i>	0.00266	0.00323	0.00065	0.00068	0.01031	0.01299
	test	0.00063	<i>0.00057</i>	0.00131	0.002	0.01515	0.0141	0.05069	0.06156
RMSE	train	0.02505	<i>0.02146</i>	0.05155	0.05687	0.02542	0.02601	0.10156	0.11399
	test	0.02957	<i>0.02377</i>	0.03619	0.04469	0.1231	0.11876	0.22515	0.24811
MAE	train	0.01428	<i>0.01455</i>	0.03821	0.04039	0.01774	0.10713	0.06153	0.06878
	test	0.01517	<i>0.0139</i>	0.02512	0.03054	0.10713	0.0181	0.22475	0.24781

Так само в нас отриманий аналогічний тренд як і у попередніх експериментах - на краще всього модель навчається на даних більшої довжини. І також сереньо взважені показники полярності більш точно передають оцінку настроїв та дають кращий результат прогнозу.

Отримано майже найкращі результати за середньозваженими оцінками за той самий з 2014 року, а саме: MSE - 0.00046, RMSE – 0.02146 та MAE – 0.01455 для тренувального датасету та MSE - 0.00057, RMSE – 0.02377 та MAE – 0.0139 для тестового.

## 6.6 Оптимізація прогнозування моделі

Для оптимізації прогнозування моделі підготуємо ще два типи датасетів на основі тих, що показали найкращі результати для кожного з інструментів аналізу настроїв, а саме: візьмемо датасети, що містять дані з початку торгів BTC та середньозважену оцінку полярності настроїв після агрегації новин. Після цього залишимо такий набір фіч, що буде застосовано для тренування моделі і прогнозу.

Тут Close, Volume та Polarity – ціна закриття торгів, обсяг торгів та полярність для першого випадку (див. рис. 6.3)

Close	Volume	Polarity
457.334015	21056800	0.000000
424.440002	34483200	0.000000
394.795990	37919700	0.000000

Рисунок 6.3 – Приклад спрощеного датасету, що містить полярність (рисунок виконано самостійно).

Та Close, Volume та Sentiment – ціна закриття торгів, обсяг торгів та полярність після функції analysis (див. рис. 4.5) для другого випадку - приклад наведено на рисунку 6.4.

Close	Volume	Sentiment
457.334015	21056800	0.000000
424.440002	34483200	0.000000
394.795990	37919700	0.000000

Рисунок 6.4 – Приклад спрощеного датасету, що містить сентменти (рисунок виконано самостійно).

Це дасть нам можливість зрозуміти, який саме показник – полярність чи сентмент (полярність після обробки) дозволить нам отримати кращий прогноз. Всього отримано 10 датасетів – по 2 типу для 5 різних інструментів аналізу. Використаємо таку ж саму модель без додаткових налаштувань, для того щоб більш ясно побачити вплив цих показників. Отримані результати в таблиці 6.6.

Таблиця 6.6 – Результати прогнозування ціни BTC на оптимізованих датасетах (Таблиця виконана самостійно)

		Original		Polarity		Sentiment	
		train	test	train	test	train	test
TextBlob	MSE	0.00033	0.00055	<i>0.0002</i>	<i>0.00031</i>	0.00028	0.00046
	RMSE	0.01828	0.02335	<i>0.01429</i>	<i>0.01757</i>	0.01671	0.02146
	MAE	0.01268	0.01561	<i>0.00693</i>	<i>0.01136</i>	0.00938	0.01283

Кінець таблиці 6.6.

		Original		Polarity		Sentiment	
		train	test	train	test	train	test
Vader	MSE	0.00034	0.00042	0.00024	0.00034	0.0009	0.00204
	RMSE	0.01853	0.02046	0.01555	0.01852	0.03	0.04516
	MAE	0.01414	0.01274	0.00976	0.01137	0.01395	0.03489
Flair	MSE	0.00052	0.001	0.00025	0.00039	0.00028	0.00039
	RMSE	0.02289	0.03165	0.01591	0.0197	0.01677	0.01978
	MAE	0.0142	0.0229	0.00797	0.01247	0.0114	0.01287
Spacy	MSE	0.00024	0.00037	0.00025	0.00041	0.00067	0.00117
	RMSE	0.01544	0.01933	0.01592	0.0202	0.00067	0.00117
	MAE	0.00785	0.0125	0.00778	0.01242	0.01727	0.02348
Bert	MSE	0.00046	0.00057	0.00029	0.00053	0.00028	0.00037
	RMSE	0.02146	0.02377	0.01689	0.02299	0.01674	0.01911
	MAE	0.01455	0.0139	0.00985	0.01793	0.0121	0.01327

У цій таблиці у колонці Original наведено результати прогнозування на датасетах до оптимізації, а у колонках Polarity та Sentiment результати відповідних датасетів після оптимізації.

Отримано кращі результати за прогнозуванням с оцінками Polarity TextBlob, а саме: MSE - 0.0002, RMSE – 0.01429 та MAE – 0.00693 для тренувального датасету та MSE - 0.00031, RMSE – 0.01757 та MAE – 0.01136 для тестового.

Це суттєво, майже в 2 рази за деякими показниками покращує результати експеримента.

## 6.7 Аналіз результатів

Як ми бачимо з результатів експерименту, найкраще значення прогнозування показало використання моделі на оптимізованім датасеті з використанням значення Polarity з використанням інструменту аналізу сентиментів TextBlob.

Для порівняння з результатами прогнозування до оптимізації, давайте візьмемо середньо-зважені та найкращі показники до оптимізації та розрахуємо

абсолютне (див. табл. 6.7) та відносне відхилення (див. табл. 6.8) після оптимізації за полярністю.

Таблиця 6.7 – Абсолютне відхилення результатів експерименту після оптимізації за полярністю. (Таблиця виконана самостійно)

		MSE	RMSE	MAE
min	train	0.00004	0.00115	0.00092
	test	0.00006	0.00176	0.00114
ave	train	0.000132	0.003608	0.004226
	test	0.000186	0.003916	0.00242

Таблиця 6.8 – Відносне відхилення результатів експерименту після оптимізації за полярністю. (Таблиця виконана самостійно)

		MSE	RMSE	MAE
min	train	0.16667	0.07448	0.11720
	test	0.16216	0.09105	0.09120
ave	train	0.34921	0.18675	0.33318
	test	0.00004	0.00115	0.00092

З цих даних можемо побачити, що оптимізація за полярністю оцінок TextBlob майже на 8-17 відсотків показало покращила прогноз відносно найкращих показників до оптимізації і майже до 35 відсотків від середніх показників полярності за всіма датасетами.

Також тренування моделі на більших інтервалах часу показує кращі результати, незважаючи на те, що кількість новин збільшувалася на протязі усього періоду (див. рис. 4.10) і їх щільність також збільшувалася у датасетах з 2021, 2023 та 2024 років по наростанню.

Використання середньо-взваженого показника полярності показує більш чутливу оцінку і більш точне прогнозування ніж інші показники настроїв, що використовувались в даному дослідженні.

## 6.8. Аналіз подальшого розвитку дослідження

Попри отримані позитивні результати дослідження треба зазначити, що можливі напрямки розвитку наступні:

- пошук та групування даних нових з різних джерел та аналіз впливу різних джерел на результати прогнозування можуть надати можливість класифікувати джерела та використовувати їх для зваженої оцінки;
- вивчення аналізу сентиментів за заголовком та за текстом окремо;
- продовження експерименту з використанням інших інструментів аналізу сентиментів;
- більш тонка настройка параметрів інструментів аналізу , таких як;
- `pos_tagger` - означає тегер частин мови, який використовується для визначення частин мови слів у тексті;
- `analyzer` - визначає аналізатор, який використовується для визначення сентименту тексту;
- `tokenizer`: - визначає токенізатор, який використовується для розбиття тексту на токени (слова або фрази);
- `np_extractor` - визначає екстрактор фраз, який використовується для виявлення іменованих фраз у тексті;
- `parser` - визначає парсер, який використовується для синтаксичного аналізу тексту. `classifier`: Визначає класифікатор, який використовується для класифікації тексту.

Ці параметри можуть бути використані при створенні об'єкту TextBlob для налаштування його функціональності.

### 6.9. Практична цінність отриманих результатів

Отримані результати дають можливість порівняти ефективність різних методів прогнозування ціни закриття криптовалюти на поточний день за допомогою використання бібліотек аналізу настроїв криптовалют.

Використовуючи ці результати можна розробити сервіс для прогнозування курсу криптовалют на поточний день. Цей сервіс може бути корисний для трейдерів, фінансових аналітиків та інших учасників ринку криптовалют.

Також це дослідження може бути корисне для наукових дослідників, студентів, що вивчають теми прогнозування ціни фінансових активів [49].

Також результати цього дослідження можуть бути використані для подальшого дослідження інструментів аналізу настроїв та прогнозування ціни криптовалют.

## ВИСНОВКИ

У ході науково-дослідної роботи було проведено аналіз предметної області, обґрунтовано методи дослідження, сформульовано етапи його проведення.

Було розглянуто штучні нейронні мережі, рекурентні нейронні мережі та LSTM моделі довго короткочасної пам'яті.

Було поставлено та параметризовано задачу багатокритеріального вибору криптовалюти, яку у результаті було розв'язано за методом лінійної адитивної згортки з ваговими коефіцієнтами і отримано найкращу альтернативу для проведення дослідження ефективності використання інформаційних технологій на основі штучного інтелекту для прогнозування курсу криптовалют – BTC.

Було проведено збір даних для побудування датасетів, для чого було розроблено програмне забезпечення з використанням мови програмування Python. Також було виконано підготовку та нормалізацію даних.

Було розглянуто інструменти для аналізу настроїв новинних стрічок із застосуванням бібліотек TextBlob, Vader, Flair, Spacy та Bert і кожний з них було використано для підготовки датасетів для тренування моделі. Від самого початку було підготовлено 40 датасетів для порівняння.

Було розроблено регресійну модель для тренування з шарами LSTM та Dense. Після тренування моделі було виявлено найкращі датасети для прогнозування ціни BTC та проведено оптимізацію датасетів, в ході якої підготовлено ще 10 датасетів для подальшого аналізу.

Після тренування моделі на оптимізованих датасетах було виявлено, що найкращі показники показав датасет із найбільшим часовим інтервалом з середньо-зваженою оцінкою полярності, що було виконано бібліотекою TextBlob.

Результати цього дослідження можуть бути використані для розробки сервісу прогнозування цін цифрових активів для подальшого дослідження інструментів аналізу настроїв та прогнозування ціни криптовалют.

Виходячи з вищесказаного можемо стверджувати, що поставлене завдання магістерського дослідження було виконано успішно.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. LEE, David Kuo Chuen; GUO, Li; and WANG, Yu. "Cryptocurrency: A new investment opportunity?". *Journal of Alternative Investments*. 20, (3), 2018, pp. 16-40. DOI: 10.3905/jai.2018.20.3.016
2. R. Raju, M. SaiVignesh and K. I. A. Prasad, "A Study of Current Cryptocurrency Systems". *International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, Chennai, India, 2018, pp. 203-209. DOI: 10.1109/ICCPEIC.2018.8525166.
3. W. Yiyang and Z. Yeze, "Cryptocurrency Price Analysis with Artificial Intelligence". *5th International Conference on Information Management (ICIM)*, Cambridge, UK, 2019, pp. 97-101. DOI: 10.1109/INFOMAN.2019.8714700.
4. S. Oikonomopoulos, K. Tzafilkou, D. Karapiperis and V. Verykios, "Cryptocurrency Price Prediction using Social Media Sentiment Analysis". *13th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Corfu, Greece, 2022, pp. 1-8. DOI: 10.1109/IISA56318.2022.9904351.
5. U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu and R. Brooks, "A brief survey of Cryptocurrency systems". *14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 745-752. DOI: 10.1109/PST.2016.7906988
6. F. Sabry, W. Labda, A. Erbad and Q. Malluhi, "Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities", in *IEEE Access*, vol. 8, pp. 175840-175858, 2020. DOI: 10.1109/ACCESS.2020.3025211
7. Svec, Jiri, Foley, Sean and Aspris, Angelo. "Market structure of cryptocurrencies", *Cryptocurrency and Blockchain Technology*, edited by Shaen Corbet, Andrew Urquhart and Larisa Yarovaya, Berlin, Boston: De Gruyter, 2020, pp. 97-112. DOI: 10/515/9783110660807-006
8. Vladimir Skavysh, Jacob Sharples, Sofia Priazhkina and Salman H. Hasham. "Market structure of cryptoasset exchanges: Introduction, challenges and emerging trends". *Staff Analytical Note/Note analytique du personnel–2024-2*, 2024, p28. DOI: 10.34989/san-2024-2
9. Octavian Nica, Karolina Piotrowska, Klaus Reiner Schenk-Hoppé.

"Cryptocurrencies: Concept and Current Market Structure". Cryptofinance, 2021 pp. 1-28, DOI: 10.1142/9789811239670\_0001.

10. A Guide to Cryptocurrency Fundamental Analysis. URL: <https://academy.binance.com/en/articles/a-guide-to-cryptocurrency-fundamental-analysis> (дата звернення: 07.06.2024).

11. The 7 Key Factors Influencing Cryptocurrency Value. URL: <https://www.makeuseof.com/factorsinfluencing-the-cryptocurrency-value> (дата звернення: 07.06.2024).

12. S. Boshuis, T. Braam, A. Pedroza Marchena and S. Jansen, "The Effect of Generic Strategies on Software Ecosystem Health: The Case of Cryptocurrency Ecosystems". IEEE/ACM 1st International Workshop on Software Health (SoHeal), 2018, pp. 10-17. URL: <https://ieeexplore.ieee.org/document/8445833> (дата звернення: 07.06.2024)

13. Jiangtao Ma, Yaqiong Qiao, Guangwu Hu, Yongzhong Huang, Arun Kumar Sangaiah, Chaoqin Zhang, et al., "De-anonymizing social networks with random forest classifier", IEEE Access, vol. 6, 2017, pp. 10139-10150. DOI: 10.1109/ACCESS.2017.2756904

14. Vynokurova O., Peleshko D., Zhernova P., Perova I., Kovalenko A. (2021) Solving Fraud Detection Tasks Based on Wavelet-Neuro Autoencoder. In: Babichev S., Lytvynenko V., Wójcik W., Vysheymyrskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2021. Advances in Intelligent Systems and Computing, vol 1246. Springer, Cham. pp. 535-546 DOI: 10.1007/978-3-030-54215-3\_34

15. T. Radivilova, L. Kirichenko, D. Ageiev and V. Bulakh, "Classification Methods of Machine Learning to Detect DDoS Attacks". 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2019, pp. 207-210. DOI: 10.1109/IDAACS.2019.8924406.

16. F. A. Cahyadi, A. I. Owen, F. Ricardo and A. A. S. Gunawan, "Blockchain Technology behind Cryptocurrency and Bitcoin for Commercial Transactions," 2021 1st

International Conference on Computer Science and Artificial Intelligence (ICCSAI), 2021, pp. 115-119. DOI: 10.1109/ICCSAI53272.2021.9609790

17. Smelyakov, Kirill, Oleksandr Bizkrovnyi, Natalia Sharonova, Serhii Smelyakov and Anastasiya Chupryna. "Building of Regression Models for Cryptocurrency Price Prediction". International Conference on Computational Linguistics and Intelligent Systems (2022), URL: <https://ceur-ws.org/Vol-3171/paper90.pdf> (дата звернення: 07.06.2024).

18. Patel, Mohil Maheshkumar, et al. "A Deep Learning-Based Cryptocurrency Price Prediction Scheme for Financial Institutions". Journal of Information Security and Applications, Elsevier, Vol. 55, 2020. DOI: 10.1016/j.jisa.2020.102583

19. Pagnottoni, Paolo. "Neural Network Models for Bitcoin Option Pricing." Frontiers in Artificial Intelligence, Frontiers Media S.A., 3 July 2019, URL: <https://www.frontiersin.org/articles/10.3389/frai.2019.00005/full> (дата звернення: 07.06.2024)

20. Sagar, Abhinav. "Cryptocurrency Price Prediction Using Deep Learning." Medium, Towards Data Science, 3 Dec. 2019. URL: <https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep-learning-70cfca50dd3a>. (дата звернення: 07.06.2024)

21. "Stock Price Prediction Using Artificial Neural Network." IJIRSET, International Journal of Innovative Research in Science, Engineering, and Technology, 28 Sept. 2021. DOI: 10.1.1.1066.9415&5

22. A. Aggarwal, I. Gupta, N. Garg and A. Goel, "Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction". Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-5. DOI: 10.1109/IC3.2019.8844928

23. Liu, Mingxi, Guowen Li, Jianping Li, Xiaoqian Zhu, and Yinhong Yao. Forecasting the price of Bitcoin using deep learning. Finance Research. 2021. URL: <https://www.sciencedirect.com/science/article/abs/pii/S1544612320304864> (дата звернення: 07.06.2024)

24. S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using

Machine Learning". 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, UK, 2018, pp. 339-343. DOI: 10.1109/PDP2018.2018.00060.

25. M. Shin, D. Mohaisen and J. Kim, "Bitcoin Price Forecasting via Ensemble-based LSTM Deep Learning Networks". 2021 International Conference on Information Networking (ICOIN), Jeju Island, Korea (South), 2021, pp. 603-608. DOI: 10.1109/ICOIN50884.2021.9333853

26. Chen, Yinghao, Xiaoliang Xie, Tianle Zhang, Jiaxian Bai, and Muzhou Hou. 2020b. A deep residual compensation extreme learning machine and applications. *Journal of Forecasting* 39, 2020, pp. 986–99. DOI: 10.1002/for.2663

27. Akyildirim, E., Cepni, O., Corbet, S. et al. Forecasting mid-price movement of Bitcoin futures using machine learning. *Ann Oper Res* 330, 2023, pp. 553–584. DOI: 10.1007/s10479-021-04205-x

28. N. Jagannath et al., "A Self-Adaptive Deep Learning-Based Algorithm for Predictive Analysis of Bitcoin Price" in *IEEE Access*, vol. 9, 2021. pp. 34054-34066. DOI: 10.1109/ACCESS.2021.3061002

29. M. Rizwan, S. Narejo and M. Javed, "Bitcoin price prediction using Deep Learning Algorithm". 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Karachi, Pakistan, 2019, pp. 1-7. DOI: 10.1109/MACS48846.2019.9024772.

30. T. Phaladisailoed and T. Numnonda, "Machine Learning Models Comparison for Bitcoin Price Prediction". 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia, 2018, pp. 506-511. DOI: 10.1109/ICITEED.2018.8534911.

31. Sebastião, H., Godinho, P. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financ Innov* 7, 2021. DOI: 10.1186/s40854-020-00217-x

32. S. Saadah and A. A. Ahmad Whafa, "Monitoring Financial Stability Based on Prediction of Cryptocurrencies Price Using Intelligent Algorithm". International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2020,

pp. 1-10. DOI: 10.1109/ICoDSA50139.2020.9212968

33. V. Derbentsev, N. Datsenko, V. Babenko, O. Pushko and O. Pursky, "Forecasting Cryptocurrency Prices Using Ensembles-Based Machine Learning Approach". IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 707-712. DOI: 10.1109/PICST51311.2020.9468090.

34. A. Politis, K. Doka and N. Koziris, "Ether Price Prediction Using Advanced Deep Learning Models". IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 2021, pp. 1-3. DOI: 10.1109/ICBC51069.2021.9461061

35. Livieris, Ioannis E., Niki Kiriakidou, Stavros Stavroyiannis, and Panagiotis Pintelas. "An Advanced CNN-LSTM Model for Cryptocurrency Forecasting". Electronics 2021, p. 287. DOI: 10.3390/electronics10030287

36. Zhang, Y., Shirakawa, M., Wang, Y. et al. "Twitter-aided decision making: a review of recent developments". Appl Intell 52, 2022, pp. 13839–13854. DOI: 10.1007/s10489-022-03241-9

37. Lamon, C., Nielsen, E. and Redondo, E., 2017. Cryptocurrency price prediction using news and social media sentiment. SMU Data Sci. Rev, 1(3), pp.1-22. URL: <https://cs229.stanford.edu/proj2017/final-reports/5237280.pdf> (дата звернення: 07.06.2024)

38. Gordiievych, A. , Shubin, I. Forecasting of airfare prices using time series. Information Technologies in Innovation Business Conference, ITIB 2015, 2015, pp. 68-71. DOI: 10.1109/ITIB.2015.7355055.

39. Afanasieva I.V., Naumov A.B., Onyshchenko K.G. "Bitcoin price prediction using the boosting algorithm". The 2nd International scientific and practical conference "Science and society: modern trends in a changing world" (January 22-24, 2024) MDPC Publishing, Vienna, Austria. 2024. 662 p. ISBN 978-3-954754-01-4

40. Наумов А.Б., Смеляков К.С., Чуприна А.С. "Емоційна оцінка новин для прогнозування курсу біткоїну". 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., –

Харків: ХНУРЕ. 2024. – 958 с. DOI 10.30837/IYF.IIS.2024

41. Evita Stenqvist and Jacob Lönnö. 2017. Predicting Bitcoin price fluctuation with Twitter sentiment analysis. URL: <https://api.semanticscholar.org/CorpusID:64544792> (дата звернення: 07.06.2024)

42. Xin Huang, Wenbin Zhang, Yiyi Huang, Xuejiao Tang, Mingli Zhang, Jayachander Surbiryala, Vasileios Iosifidis, Zhen Liu, and Ji Zhang. 2021. “LSTM Based Sentiment Analysis for Cryptocurrency Prediction”. arXiv preprint arXiv:2103.14804, 2021. DOI: 10.48550/arXiv.2103.14804

43. Franco Valencia, Alfonso Gómez-Espinosa, and Benjamín Valdés-Aguirre. “Price movement prediction of cryptocurrencies using sentiment analysis and machine learning”. Entropy 21, 6, 2019, p.589. DOI: 10.3390/e21060589

44. Nagesh Singh Chauhan. Introduction to RNN and LSTM. 2020. URL: <https://www.theaidream.com/post/introduction-to-rnn-and-lstm> (дата звернення: 07.06.2024)

45. Jurafsky, Daniel and Martin, James H. Speech and Language Processing (2Nd Edition) 2009. ISBN 0131873210

46. Josh Brule, Neal Gupta, Assaf Magen, Saeedreza Seddhighin. Guiding Hidden Layer Representations for Improved Rule Extraction from Neural Networks. URL: <https://www.cs.umd.edu/research/projects/16671> (дата звернення: 07.06.2024)

47. Hira Ahmad. Navigating the Realm of Sequential Data Handling: RNNs, LSTMs, GRUs, GANs, and Transformers. URL: <https://blog.gopenai.com/navigating-the-realm-of-sequential-data-handling-rnns-lstms-grus-gans-and-transformers-2d26cefb9416> (дата звернення: 07.06.2024)

48. Dmytro Nazarenko, Irina Afanasieva, Natalia Golian, Vira Golian Investigation of the deep learning approaches to classify emotions in texts / al. Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021). Volume I: Main Conference, Lviv, 22–23 April 2021, pp. 206–224. URL: <https://ceur-ws.org/Vol-2870/paper19.pdf> (дата звернення: 07.06.2024)

49. Ворочек О.Г, Дударь В.В. Исследование интеллектуальных средств анализа и прогнозирования. Журнал «Вестник» Херсонского национального технического университета №4(27), 2007.