

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем
122 «Комп'ютерні науки»
Освітня програма «Прикладне програмування»

Кваліфікаційна робота бакалавра

на тему: «Веб-сервіс із прогнозування котирувань на ринку криптовалют»

Виконала _____

Ветряченко Тетяна Сергіївна

Керівник Пирог Микола Володимирович

Попередній захист:

Завідувач кафедри

_____ Плескач В.Л.

Київ – 2024

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№з/п	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	01.11.2023	Виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	15.11.2023	Виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	27.11.2023	Виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.12.2023	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.12.2023	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	31.12.2023	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2024	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	29.03.2024	Виконано
9.	Подання роботи у першому варіанті	29.04.2024	Виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	02.05.2024	Виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	15.05.2024	Виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	27.05.2024	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	13.06.2024	Виконано
14.	Захист кваліфікаційної роботи бакалавра	17.06.2024	Виконано

Здобувач вищої освіти _____

Керівник _____

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Складові частини кваліфікаційної роботи бакалавра	Обсяг, арк.
Титульний аркуш	1
Календарний план кваліфікаційної роботи бакалавра	1
Відомість кваліфікаційної роботи бакалавра	1
Анотація	2
Анотація (іноземною мовою – англійською)	2
Зміст	1
Перелік скорочень, умовних позначень та термінів	1
Вступ	2
Розділ 1	20
Розділ 2	10
Розділ 3	14
Висновки	2
Перелік джерел посилання	4

				ДП ХХХХ 00.000.00			
		ПІБ	Підпис	Дата			
Розробник	Ветряченко Т.С.				Відомість кваліфікаційної роботи	Лист	Листів
Керівник	Пирог М.В.						
Н/контроль							
Зав.кафедри	Плескач В.Л.						

АНОТАЦІЯ

Кваліфікаційна робота: 78 с., 12 рис., 2 табл., 35 джерел, 2 дод.

Актуальність роботи зумовлена зростаючим інтересом до криптовалют та необхідністю точного прогнозування їхніх котирувань для інвесторів.

Метою роботи є реалізація веб-сервісу для отримання графічних даних прогнозування криптовалют із застосуванням методів машинного навчання.

Об'єктом роботи є процес прогнозування котирувань на ринку криптовалют з використанням веб-сервісів.

Предметом роботи є інструментарій методів та алгоритмів аналізу даних і машинного навчання, що застосовуються для прогнозування змін котирувань криптовалют, а також їх інтеграція у веб-сервіс для автоматизації процесу.

Завдання роботи включають:

- дослідження методів прогнозування котирувань криптовалют;
- огляд наявних веб-сервісів;
- аналіз програмно-технологічного стека побудови зручної платформи для візуалізації прогнозування котирувань криптовалют;
- проектування архітектури веб-сервісу та реалізація його backend і frontend частин.

Основними результатами роботи є створення функціонального веб-сервісу для прогнозування котирувань криптовалют з використанням моделі LSTM. Реалізовано повний цикл розробки веб-сервісу, від аналізу методів прогнозування до створення архітектури та реалізації backend і frontend частин. Проведено порівняльний аналіз різних підходів до прогнозування, а також оцінку точності моделей. Веб-сервіс надає можливість прогнозувати котирування криптовалют, використовуючи історичні дані, та візуалізувати результати прогнозів.

Новизна роботи полягає у поєднанні традиційних методів технічного та фундаментального аналізу з сучасними методами машинного навчання для

створення ефективної моделі прогнозування котирувань криптовалют. Запропонований підхід включає також нормалізацію даних та використання різних статистичних показників для покращення якості прогнозів. Ключове значення має побудова зручного інструменту візуалізації прогнозів моделі.

Практичне значення роботи полягає у можливості використання розробленого веб-сервісу для інвесторів та аналітиків, що займаються криптовалютами. Веб-сервіс може стати корисним інструментом для наочного розуміння результатів роботи LSTM моделі. Розроблена система може бути також використана як основа для подальших досліджень та вдосконалення методів прогнозування котирувань фінансових інструментів.

Ключові слова: прогнозування котирувань, криптовалюти, машинне навчання, LSTM, веб-сервіс.

ABSTRACT

Qualification work: 78 p., 12 figs., 2 tables, 35 sources, 2 appendixes.

The relevance of the work is due to the growing interest in cryptocurrencies and the need for accurate forecasting of their quotes for investors.

The purpose of this work is to implement a web service for obtaining graphical data on cryptocurrency forecasting using machine learning methods.

The object of the work is the process of forecasting quotes on the cryptocurrency market using web services.

The subject of this work is a toolkit of data analysis and machine learning methods and algorithms used to predict changes in cryptocurrency quotes, as well as their integration into a web service to automate this process.

The tasks of the work include:

- research of methods for forecasting cryptocurrency quotes;
- review of existing web services;
- analysis of the software and technology stack for building a convenient platform for visualizing cryptocurrency quotes forecasting;
- designing the architecture of the web service and implementing its backend and frontend parts.

The main results of the work include the creation of a functional web service for forecasting cryptocurrency quotes using the LSTM model. A full cycle of web service development has been implemented, from the analysis of forecasting methods to the creation of architecture and the implementation of backend and frontend parts. A comparative analysis of different forecasting approaches was conducted, as well as an assessment of the accuracy of the models. The web service allows forecasting cryptocurrency quotes using historical data and visualizing the results of forecasts.

The novelty of the work is the combination of traditional methods of technical and fundamental analysis with modern machine learning methods to create an effective

model for forecasting cryptocurrency quotes. The key is to build a convenient tool for visualizing the model's forecasts.

The practical significance of the work lies in the possibility of using the developed web service for investors and analysts dealing with cryptocurrencies. The web service can be a useful tool for visualizing the results of the LSTM model. The developed system can also be used as a basis for further research and improvement of methods for forecasting financial instrument quotes.

Keywords: forecasting quotes, cryptocurrencies, machine learning, LSTM, web service.

Зміст

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ	9
ВСТУП.....	10
РОЗДІЛ 1 ОГЛЯД СУЧАСНОГО СТАНУ РИНКУ КРИПТОВАЛЮТ ТА ВЕБ-СЕРВІСІВ ДЛЯ ПРОГНОЗУВАННЯ	12
1.1 Дослідження поняття криптовалюта та особливості криптовалютного ринку	12
1.2 Методи та підходи прогнозування котирувань криптовалют	18
1.3 Веб-сервіси для прогнозування котирувань криптовалют	26
РОЗДІЛ 2 ТЕХНОЛОГІЧНИЙ СТЕК ТА МОДЕЛІ ДЛЯ РОЗРОБКИ.....	32
2.1 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-СЕРВІСУ	32
2.2. LSTM як моделі для прогнозування криптовалют	35
РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ КОТИРУВАНЬ КРИПТОВАЛЮТ	42
3.1 АРХІТЕКТУРА ВЕБ-СЕРВІСУ	42
3.2 РЕАЛІЗАЦІЯ ВАСКЕНД-ЧАСТИНИ ВЕБ-СЕРВІСУ	46
3.3 ПРОГРАМУВАННЯ FRONTEND-ЧАСТИНИ ВЕБ-СЕРВІСУ	52
ВИСНОВОК	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТОК А	62
ДОДАТОК Б.....	67

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

англ. – англійська

BTC – Bitcoin

ETH – Ether

BNB – Binance Coin

USDT – Tether

PoW – Proof-of-Work

PoS – Proof-of-Stake

BEP-2 – Binance Chain Evolution Proposal 2

BEP-20 – Binance Smart Chain Evolution Proposal 20

EVM – Ethereum Virtual Machine

RST – Reinforcement Sensitivity Theory

PCA – Principal Component Analysis

К-середніх – K-means clustering

RNN – Recurrent Neural Network

LSTM – Long Short-Term Memory

BPTT – Backpropagation Through Time

RMSE – Root Mean Squared Error

MSE – Mean Squared Error

MAE – Mean Absolute Error

OOB – Out-of-Bag

CSS – Cascading Style Sheets

UI – User Interface

HTML – Hypertext Markup Language

API – Application Programming Interface

ВСТУП

Актуальність теми зумовлена зростаючим інтересом до криптовалют та необхідністю точного прогнозування їхніх котирувань для інвесторів.

Метою роботи є реалізація веб-сервісу для отримання графічних даних прогнозування криптовалют із застосуванням методів машинного навчання.

Завдання дослідження включає:

- дослідження методів прогнозування котирувань криптовалют;
- огляд існуючих веб-сервісів;
- аналіз програмно-технологічного стеку побудови зручної платформи для візуалізації прогнозування котирувань криптовалют;
- проектування архітектури веб-сервісу та реалізація його backend і frontend частин.

Об'єктом дослідження є процес прогнозування котирувань криптовалют існуючими методами машинного навчання.

Предметом дослідження є методи машинного навчання для прогнозування котирувань криптовалют та програмно-технічні рішення для побудови веб-сервісів.

Методами дослідження є огляд наукових статей, книг та інших джерел, що стосуються методів прогнозування котирувань криптовалют, аналіз існуючих веб-сервісів, що пропонують подібні функції, вибір та застосування методів прогнозування, реалізація моделі LSTM для прогнозування, нормалізація даних та застосування статистичних показників для покращення якості прогнозів, розробка архітектури веб-сервісу, включаючи його backend та frontend частин, реалізація та тестування веб-сервісу, тестування веб-сервісу на реальних даних криптовалют, оцінка точності прогнозів та порівняння з іншими методами, розробка інтерфейсу для візуалізації прогнозів котирувань криптовалют, створення зручного інструменту для користувачів для аналізу прогнозів,

порівняння точності та ефективності запропонованої моделі LSTM з іншими методами прогнозування, аналіз переваг та недоліків кожного методу.

Структура роботи:

Дипломна робота складається зі вступу, трьох розділів, розподілених на підрозділи та висновку.

РОЗДІЛ 1

ОГЛЯД СУЧАСНОГО СТАНУ РИНКУ КРИПТОВАЛЮТ ТА ВЕБ-СЕРВІСІВ ДЛЯ ПРОГНОЗУВАННЯ

1.1 Дослідження поняття криптовалюта та особливості криптовалютного ринку

1.1.1 Визначення, історія, загальні характеристики

Криптовалюта є децентралізованою конвертованою валютою, заснованою на математичних принципах, яка захищена криптографічними методами. Вона використовує криптографію для створення розподіленої, децентралізованої та безпечної інформаційної економіки. Стан і транзакції криптовалюти фіксуються та перевіряються за допомогою технології блокчейн, що дозволяє створювати децентралізовані та незмінні записи транзакцій.

Термін «криптовалюта» отримав поширення у 2011 році після публікації статті Енді Грінберга «Crypto currency» («Криптовалюта») у журналі Forbes.

Офіційна історія криптовалюти починається з 2008 року, коли Сатоші Накамото опублікував свою роботу "Bitcoin: A Peer-to-Peer Electronic Cash System" [6] у Cryptography Mailing List Email. Варто зазначити, що концепція децентралізованої цифрової валюти була вперше описана Ніком Жабо ще в 1998 році. Однак, Жабо не реалізував свою ідею bit gold, яка за деякими ключовими характеристиками та принципом функціонування нагадує біткоїн, зокрема через застосування методу "proof-of-work" та інших подібних механізмів.

Характерними рисами криптовалюти є децентралізованість, що сприяє P2P-транзакціям напряму між людьми. Але замість фізичних гаманців і банківських рахунків, люди отримують доступ до своєї криптовалюти через унікальні криптогаманці чи криптобіржі. Також до основних рис можна додати конвертованість, конфіденційність та віртуальність.

Блокчейн – це децентралізований цифровий реєстр, який безпечно записує транзакції в мережі комп'ютерів. Він використовує криптографію для зв'язування блоків даних у ланцюжок, що ускладнює підробку інформації [3].

Транзакції в блокчейні є публічними, а адреси (публічні ключі) – псевдонімічними, хоча і не повністю анонімними. Іншими словами, хоч транзакції видно в блокчейні, користувачів, які стоять за ними, важко ідентифікувати. Криптовалюти досягають цього завдяки використанню криптографічних методів, таких як хеш-функції та цифрові підписи.

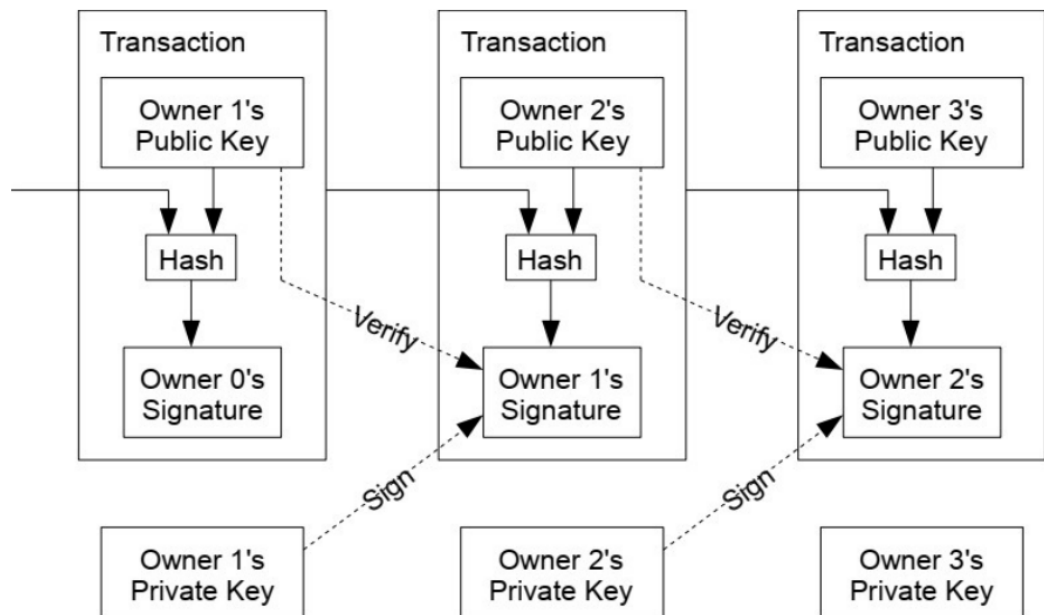


Рисунок 1.1 – Транзакції з електронними монетами

Криптовалюта ґрунтується на децентралізованій мережі комп'ютерів, це, по суті, розподілений цифровий реєстр, де транзакції записуються на безліч спеціалізованих комп'ютерів у цій мережі, цей процес можна побачити на рисунку 1.1. Кожен з цих комп'ютерів, які називаються вузлами, зберігає копію реєстру. Завдяки алгоритму консенсусу блокчейни синхронізуються, що гарантує відхилення фальшивих або несумісних копій. Ця децентралізована структура робить мережу більш безпечною, адже вона не має єдиної точки відмови, як-от банківське сховище, яку могли б зламати зловмисники.

Завдяки криптовалюті люди можуть переказувати кошти безпосередньо один одному. У типовій транзакції з криптовалютою відправник ініціює переказ, створюючи цифровий підпис за допомогою свого приватного ключа. Потім транзакція надсилається в мережу, де її перевіряють вузли, підтверджуючи цифровий підпис та те, що у відправника достатньо коштів[5].

Після успішної перевірки транзакція додається до нового блоку, який потім включається до чинного блокчейну, візуалізацію цього процесу можна побачити на рисунку 1.2. Хоча цей процес може здатися складним, користувачам не потрібно про нього турбуватися, адже його виконують майнери.

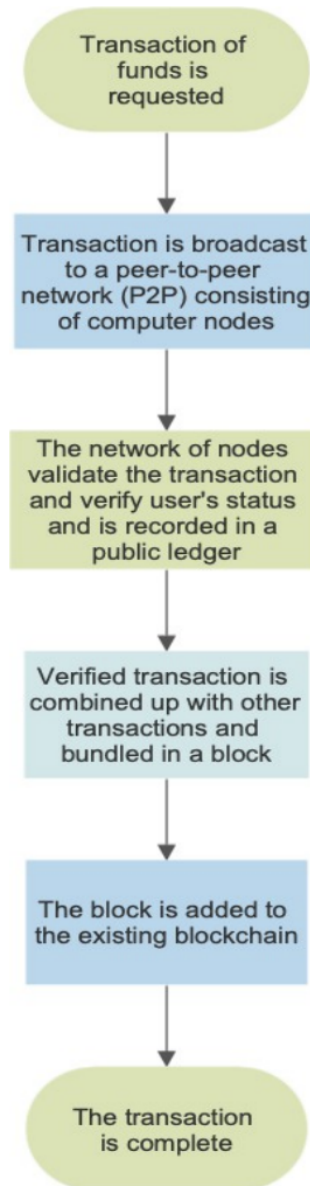


Рисунок 1.2 – Процеси транзакцій, пов'язані з криптовалютою

1.1.2 Класифікація та опис популярних криптовалют

Сьогодні популярні такі види криптовалют: Bitcoin (BTC), Ether (ETH), Binance Coin (BNB), Tether (USDT).

Bitcoin (BTC) – це найвідоміша криптовалюта. Її робота ґрунтується на алгоритмі консенсусу Proof-of-Work (PoW), де спеціальні комп'ютери (майнери) змагаються за право підтверджувати транзакції та забезпечувати функціонування мережі.

Важливою характеристикою BTC є його обмежена кількість – 21 мільйон монет. Ця штучна дефіцитність робить Bitcoin цінним активом та допомагає підтримувати його вартість [4].

Ether (ETH) – це друга за популярністю криптовалюта, створена у 2015. Крім функції обміну цінностями, ETH пропонує платформу для розробки децентралізованих додатків за допомогою смартконтрактів.

На початковому етапі Ethereum використовував алгоритм консенсусу Proof-of-Work (PoW), аналогічно до Bitcoin. Однак з часом проєкт перейшов на більш доцільну та енергоефективну модель Proof-of-Stake (PoS). Ця зміна дозволила користувачам валідувати транзакції та захищати мережу шляхом розміщення своїх ETH на стейкінгу, замість застосування енергомістких вузлів.

BNB (Build N' Build), був запущений у 2017 році криптовалютною біржею Binance як токен стандарту ERC-20 на блокчейні Ethereum. У 2019 році цей токен перейшов на власний блокчейн під назвою BNB Chain, де функціонує як токен стандарту BEP-2.

Пізніше був створений блокчейн Binance Smart Chain (BSC), який нині відомий як BNB Smart Chain. На сьогодні криптовалюта BNB існує одночасно на BNB Chain (BEP-2) та BSC (BEP-20). Важливо зазначити, що BNB Chain включає два блокчейни: BSC, сумісний з Ethereum Virtual Machine (EVM), та BNB Beacon Chain (раніше Binance Chain), який використовується для управління, стейкінгу та голосування.

BNB Chain забезпечує платформу для розробки смартконтрактів і децентралізованих додатків (dApp), пропонуючи нижчі транзакційні комісії та швидший час обробки порівняно з багатьма іншими блокчейнами. BNB має різні варіанти використання, включаючи оплату комісій за транзакції на BNB Chain та за торгівлю на Binance, участь у токенсейлах, стейкінг для верифікації мережі.

Binance також використовує механізм спалювання токенів, який періодично зменшує загальну пропозицію BNB. Це допомагає підтримувати його вартість.

Tether (USDT) – це криптовалюта, прив'язаний до курсу USD. Він був запущений у 2014 році компанією Tether Limited Inc. Такі криптовалюти, прагнуть підтримувати стабільну вартість, прив'язану до стабільного активу, як правило, до фіатних валют. У випадку з USDT, кожен токен підкріплений еквівалентною сумою активів, що зберігаються компанією у інвестиційних інструментах, які дозволяють прив'язатись до долару. Це робить USDT стійким до коливань цін, що характерні для інших криптовалют.

USDT – це популярний вибір для інвесторів, які хочуть отримати переваги криптовалюти без ризику значних коливань ціни.

Розуміння визначення та загальних характеристик криптовалют має вирішальне значення для прогнозування тенденцій на їх ринку. Визнаючи еволюцію ролі криптовалют як валюти та технології, зацікавлені сторони можуть ефективно орієнтуватися в складній ринковій динаміці та факторах, що впливають на ціноутворення, в цьому трансформаційному фінансовому ландшафті.

1.1.3 Особливості та фактори, що впливають на котирування

Інвестування в криптовалюту може принести значний прибуток, але воно також пов'язане з високими ризиками. Найголовнішим з них є волатильність – різкі коливання цін на криптовалюту.

На курс криптовалюти можуть впливати спекулятивні дії учасників ринку, або фінансові кризи, війни та інші події. Також можуть спіткати технічні проблеми, наприклад збої в роботі блокчейнів або бірж.

Наприклад, Bitcoin, який є найпопулярнішою криптовалютою, може значно коливатись у вартості протягом декількох днів.

Ще одним ризиком є децентралізація криптовалютних ринків. Транзакції з криптовалютами відбуваються без посередництва банків, що дає можливість здійснювати їх цілодобово.

Однак децентралізація має й свої недоліки.

Якщо користувач вкаже неправильну адресу одержувача, повернути кошти буде неможливо. У разі шахрайства або втрати приватного ключа відновити доступ до своїх коштів буде практично неможливо.

Важливо усвідомлювати ці ризики перед тим, як інвестувати в криптовалюту.

Ринок криптовалют, однак, пропонує тисячі криптовалют на вибір, тому інвестори можуть інвестувати відповідно до свого рівня толерантності до ризику та очікуваного прибутку. Оскільки інвестори мотивовані відчуттям збільшення винагороди в результаті інвестування в криптовалюту, передбачається, що теорія чутливості до підкріплення (Reinforcement Sensitivity Theory, RST) робить свій внесок в ефект притягування. Особистий інноваційний фактор також бере участь у швартувальному ефекті. На основі емпіричних даних зроблено висновок, що основними причинами інвестування споживачів у волатильний ринок криптовалют є вплив, інноваційність, сприйняття ризику, чутливість до винагороди та знання криптовалютного процесу [6].

Онлайн-біржі керують ринком криптовалют, так само як і біржі фіатних валют. Вони контролюють продаж і купівлю різних криптовалют споживачами. Серед них, зокрема, CME, CBOE, Binance, Coinbase, BitMex та Bitstamp. Існують також біржі, які пропонують власні токени, які є "рідними" для біржі. Існує кілька причин, чому їх включають до складу бірж, зокрема, багатостороння довіра, підвищена ліквідність, знижки на комісійні та управління.

Цифрову валюту купують, продають, переказують і зберігають мільйони споживачів через платформи криптообміну. Криптовалютні біржі дозволяють споживачам створювати гаманці для здійснення транзакцій. Платформи підтримують зручні інтерфейси як онлайн, так і через мобільні додатки (Android, Mac, iOS).

1.1.4 Криптовалюта в Україні

Україна посідає одне з перших місць у світі за кількістю людей, які володіють криптовалютою. Більшість українців не володіє значними обсягами криптовалюти, але багато хто активно цікавиться цією індустрією та робить дрібні інвестиції.

Важливою подією стало прийняття в Україні закону, який легалізував криптовалютний ринок. Це дає українцям можливість:

- законно здійснювати операції з віртуальними активами;
- користуватися офіційними криптовалютними біржами;
- платити за товари та послуги криптовалютою.

Існує декілька законних та безпечних способів купити криптовалюту в Україні. Це може бути покупка через криптовалютні біржі, такі як Binance, яка є найбільшою у світі, також на P2P-платформах, де користувачі укладають угоди безпосередньо, без посередників і через онлайн-обмінники.

1.2 Методи та підходи прогнозування котирувань криптовалют

1.2.1 Технічний аналіз

Технічний аналіз – це метод дослідження фінансових ринків, включаючи ринок криптовалют, який використовує аналіз та інтерпретацію історичних даних про ціни.

Основна ідея полягає в тому, що рухи цін у минулому можуть допомогти передбачити майбутні тенденції та тренди на ринку. На відміну від фундаментального аналізу, технічний аналіз не потребує вивчення економічних показників, новин про компанії чи ринок. Натомість технічні аналітики зосереджуються на ціні та графічних паттернах, які вона утворює.

Переваги використання технічного аналізу:

- допомагає дотримуватися чіткої стратегії та уникати емоційних рішень, які часто призводять до втрат;

- ґрунтується на даних, а не на суб'єктивних думках чи прогнозах;
- може допомогти ідентифікувати потенційні розвороти ринку та скористатися ними.

Важливо зазначити, що технічний аналіз не є бездоганим методом. Він не гарантує точних прогнозів, і завжди існує ризик помилки. ТА не враховує непередбачувані події, які можуть вплинути на ринок. Перед використанням ТА рекомендується вивчити основи технічного аналізу та різні технічні індикатори. Практикуватися на демо-рахунку, перш ніж ризикувати реальними коштами.

1.2.2 Фундаментальний аналіз

Фундаментальний аналіз – це метод оцінки вартості активів, таких як акції або криптовалюта, шляхом вивчення їхніх внутрішніх та зовнішніх характеристик.

Інвестори та трейдери використовують фундаментальний аналіз, щоб визначити, чи є актив переоціненим, недооціненим або справедливо оціненим.

Існують деякі ключові фактори, які враховуються при проведенні фундаментального аналізу. Внутрішні фактори, такі як прибуток, збитки, грошовий потік та співвідношення боргу до власного капіталу. Також сюди можна додати якість менеджменту та команди і конкурентні переваги.

Серед зовнішніх факторів можна виділити економічні умови, такі як відсоткові ставки, інфляція тощо. Також це можуть бути умови галузі та політичні фактори.

Існує два основних підходи до фундаментального аналізу.

Підхід “знизу вгору” починається з аналізу конкретної компанії або активу та просувається до вивчення ширших факторів, які на них впливають.

А підхід “зверху вниз” починається з аналізу загальних макроекономічних умов та перспектив галузі, а потім звужує коло до конкретних компаній або активів.

Метою фундаментального аналізу є визначення очікуваної вартості активу. Якщо очікувана вартість вище за поточну ринкову ціну, актив вважається

недооціненим, і навпаки, якщо очікувана вартість нижча за поточну ринкову ціну, актив вважається переоціненим.

Фундаментальний аналіз може допомогти інвесторам приймати обґрунтовані рішення про те, коли купувати, продавати чи утримувати активи.

Переваги фундаментального аналізу:

- ґрунтується на вивченні реальних фінансових показників та інших факторів;
- доступний, інформацію, необхідну для аналізу, можна знайти у звітах компаній, фінансових новинних джерелах та інших загальнодоступних ресурсах;
- є можливість виявлення недооцінених активів;

Недоліки фундаментального аналізу:

- проведення якісного фундаментального аналізу може бути складним та трудомістким завданням;
- не дає швидких результатів, щоб отримати точні прогнози, потрібний час для вивчення та аналізу інформації;
- не завжди ефективний на криптовалютному ринку.

1.2.3 Прогнозування на основі машинного навчання

Машинне навчання пропонує низку потужних інструментів, які можна використовувати для аналізу історичних даних про ціни, визначення закономірностей і потенційного прогнозування майбутніх рухів. Існують парадигми які охоплюють різні методи, від традиційних статистичних моделей до складних алгоритмів глибокого навчання, кожна з яких спрямована на те, щоб пролити світло на складну динаміку, яка керує цінами криптовалюти.

Кероване навчання (Supervised Learning) – домінуючий підхід, який використовує історичні цінові дані як марковані навчальні множини. Мітки можуть представляти майбутні ціни на певних часових горизонтах (наприклад, щоденна ціна закриття) або спрямовані рухи (вгору/вниз).

До популярних алгоритмів керованого навчання відносяться регресійні методи та нейронні моделі.

Щодо регресійних методів, наприклад, лінійна регресія та її різновиди встановлюють математичний зв'язок між характеристиками (наприклад, обсягом торгів, ринковими настроями) і прогнозами цін. Складніші моделі, такі як регресія опорних векторів (SVR) і Random Forest, можуть відображати нелінійні закономірності.

SVR це алгоритм, заснований на машинах опорних векторів, які зазвичай використовуються для задач класифікації, його візуалізацію продемонстровано на рисунку 1.3. SVR адаптує концепцію SVM для задач регресії, де потрібно передбачити безперервну цільову змінну.

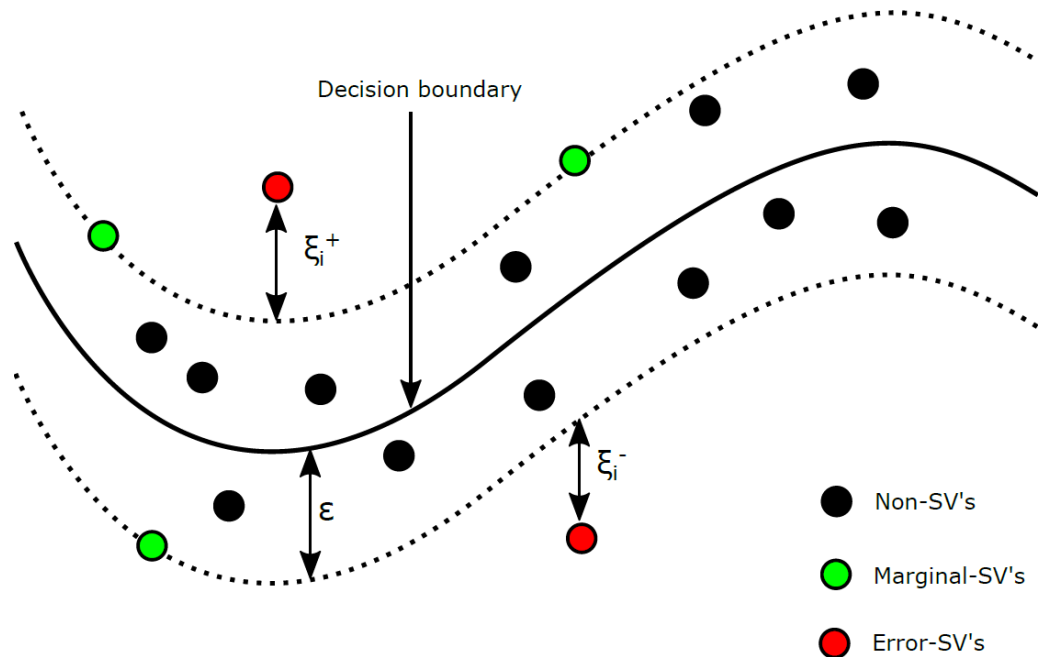


Рисунок 1.3 – Support Vector Regression (SVR)

Простіше кажучи, SVR має на меті знайти функцію, яка найкраще відповідає зв'язку між вхідними змінними та неперервною цільовою змінною, мінімізуючи при цьому похибку прогнозування.

SVR може моделювати складні взаємозв'язки у ваших даних, навіть якщо вони нелінійні. Це робить його хорошим вибором для даних, які мають багато кривих, вигинів або нерівностей. Також визначає точки даних, які є найближчими до лінії ідеальної відповідності (або гіперплощини у вищих

вимірах), і використовує ці точки для визначення допустимої похибки. Ці точки називаються опорними векторами, і вони відіграють вирішальну роль у визначенні поведінки моделі. Використовується SVR в різних сферах, включаючи фінанси, економіку та інженерію.

Random Forest – це потужний алгоритм машинного навчання, який поєднує прогнози кількох дерев рішень, щоб зробити точніший прогноз. Він базується на ідеї ансамблевого навчання (ensemble learning), де декілька моделей навчаються на одних і тих самих даних, але з різними параметрами, а потім їх прогнози комбінуються для отримання більш точного результату. Це схоже на те, що група експертів збирається разом, щоб проголосувати за найкращу відповідь.

Головний процес виконується так, що спочатку робиться підвибірка даних, тобто алгоритм починає роботу з того, що робить вибірки (із заміною) з оригінальних навчальних даних. Це створює завантажувальні репліки, які по суті є копіями вихідних даних з деякими точками, що з'являються кілька разів, а інші взагалі пропущені. Це вносить випадковість у модель.

Далі в кожному вузлі дерева рішень, що будується, випадкова підмножина ознак (предикторів) вибирається із загального пулу. Це вносить додаткову випадковість і запобігає надмірному пристосуванню до якоїсь конкретної ознаки. Потім визначається оптимальний розподіл між цими ознаками для класифікації або прогнозування цільової змінної.

Наступний етап – побудова дерева рішень. Окремі дерева рішень будуються з використанням завантажених даних і випадково вибраних підмножин ознак. Ці дерева ростуть до тих пір, поки не буде досягнуто певного критерію зупинки, наприклад, досягнення максимальної глибини або мінімальної кількості точок даних у вузлі.

Для нового, невидимого екземпляра кожне дерево в лісі голосує за клас або прогнозує значення на основі вивчених правил прийняття рішень. При класифікації більшість голосів визначає передбачуваний клас. У регресії середнє значення прогнозів окремих дерев вважається кінцевим результатом.

Останній етап – це оцінка помилки, під час навчання використовується метод, який називається "дані поза пакетом" (OOB). Це стосується точок даних, не включених до конкретної бутстреп вибірки для дерева. Дані OOB використовуються для оцінки продуктивності кожного дерева, забезпечуючи незміщену оцінку узагальнюваності моделі випадкових лісів.

Головне про метод випадкових лісів:

- ансамблеве навчання допомагає усереднити помилки окремих дерев рішень, що призводить до більш надійної моделі з меншою дисперсією;
- випадкові ліси за своєю суттю можуть впоратися з відсутніми значеннями в даних, оскільки вони розглядають лише випадкову підмножину ознак при кожному розбитті;
- дає уявлення про важливість ознак.

Хоча важливість ознак можна визначити, окремі структури дерев можуть бути складними та менш інтерпретованими порівняно з простішими моделями.

Random Forest (рис. 1.3) пропонує потужний підхід до завдань керованого навчання, поєднуючи в собі сильні сторони декількох дерев рішень. Їх надійність, здатність обробляти відсутні дані та вбудована оцінка важливості ознак роблять їх цінним інструментом у різних програмах машинного навчання.

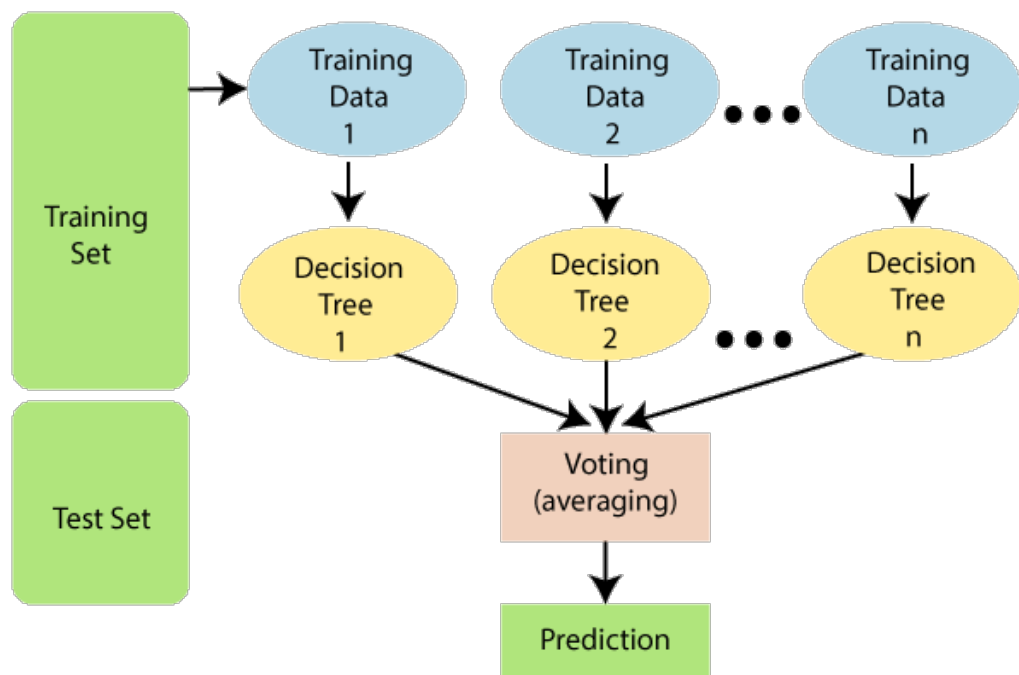


Рисунок 1.3 – Random Forest

Нейронні мережі – архітектури глибокого навчання, зокрема, рекурентні нейронні мережі (RNN) і мережі з довгою короткочасною пам'яттю (LSTM), відмінно справляються з обробкою послідовних даних, таких як цінові дані часових рядів. Ці моделі можуть вивчати складні часові залежності в даних і використовувати їх для прогнозування.

Декілька слів про RNN. Розшифровується як повторювана нейронна мережа. Це тип штучної нейронної мережі, який особливо добре обробляє послідовні дані, такі як: текст, мова або дані часових рядів. На відміну від традиційних нейронних мереж, RNN можуть враховувати минулу інформацію під час обробки нових даних. Ця «пам'ять» дозволяє їм розуміти зв'язки між різними частинами послідовності та робити прогнози на основі цього контексту. RNN призначені для обробки інформації, яка надходить у певному порядку, наприклад, слова в реченні або дані, зібрані з часом. Також мають спеціальну внутрішню структуру, яка дозволяє їм зберігати інформацію з попередніх вхідних даних. RNN обробляють дані крок за кроком, на кожному кроці вони беруть поточний вхід, поєднують його з інформацією, що зберігається в їхній пам'яті, а потім використовують цю об'єднану інформацію, щоб зробити вихід і оновити свою пам'ять.

RNN широко використовуються в різних застосунках, зокрема в машинному перекладі, враховуючи контекст кожного слова в реченні; розпізнаванні мовлення, розуміючи послідовність звуків і їх співвідношення;

в генерації тексту, такі як, чат-боти чи помічники для творчого написання;
і в прогнозуванні фондового ринку.

Прогнозування на основі RNN є ітеративним процесом. Може знадобитися налаштувати архітектуру моделі, параметри навчання та попередню обробку даних на основі ваших конкретних даних і цілей прогнозування. Існують також різні бібліотеки та фреймворки, такі як TensorFlow або PyTorch, які надають інструменти для створення та навчання моделей RNN для прогнозування часових рядів.

Мережі довгострокової короткочасної пам'яті (LSTM) — це тип архітектури рекурентної нейронної мережі (RNN), призначений для подолання обмежень традиційних RNN у захопленні довготривалих залежностей у послідовних даних. Мережі LSTM вперше були представлені Зеппом Хохрайтером і Юргеном Шмідхубером у 1997 році.

На відміну від стандартних RNN, LSTM мають вбудований механізм для обробки довгострокових залежностей. Це робить їх ідеальними для аналізу даних часових рядів, наприклад історичних цін на криптовалюту, де минула інформація може містити підказки про майбутні тенденції.

LSTM використовує комірку пам'яті для зберігання та отримання вхідних даних з попередніх часових кроків, що робить його придатним для моделювання послідовних даних з довготривалими залежностями.

LSTM є чудовим інструментом для прогнозування цін на криптовалюту. LSTM обробляють історичні дані про ціни пункт за пунктом, вони містять спеціальні ворота, які контролюють потік інформації в мережі. Ці ворота дозволяють LSTM вибірково запам'ятовувати або забувати минулі дані, може ідентифікувати закономірності, які зберігаються протягом тривалого часу.

Поширена проблема в RNN, коли градієнти стають занадто малими або великими під час зворотного поширення, що перешкоджає навчанню. LSTM пом'якшують це, дозволяючи їм навчатися на довших послідовностях. Аналізуючи історичні дані, LSTM поступово будує модель, яка фіксує основну динаміку цін на криптовалюту. Потім цю модель можна використовувати для прогнозування майбутніх цін.

Дослідження показали, що LSTM можуть досягати менших похибок прогнозування порівняно з такими методами, як опорні векторні машини (SVM) або поліноміальна регресія.

Навчання без нагляду це підхід, що може бути цінним для розвідувального аналізу, щоб виявити приховані закономірності та потенційну кластеризацію в даних криптовалютного ринку. Такі методи, як аналіз головних компонент (PCA)

і кластеризація за методом К-середніх, можуть допомогти виявити основні взаємозв'язки між різними криптовалютами або сегментами ринку.

Якість і вибір навчальних даних суттєво впливають на продуктивність моделі. Такі фактори, як нормалізація даних, обробка відсутніх значень і включення відповідних характеристик (наприклад, хеш-рейт мережі, настрої в соціальних мережах), мають вирішальне значення для ефективного прогнозування на основі машинного навчання.

Природа "чорного ящика" деяких складних моделей машинного навчання може ускладнити розуміння того, як вони роблять прогнози. Такі методи, як аналіз важливості ознак, можуть допомогти розв'язати цю проблему. Крім того, надмірне припасування, коли модель запам'ятовує навчальні дані, але не може узагальнити їх на невидимі дані, є основним ризиком. Методи регуляризації та набори валідації є важливими для пом'якшення цієї проблеми.

Машинне навчання для прогнозування цін на криптовалюту - сфера, що швидко розвивається. Тривають дослідження, спрямовані на вивчення нових архітектур, залучення нових джерел даних (наприклад, даних в ланцюжку) і розв'язання проблем, притаманних цій галузі. Враховуючи ці аспекти, дослідники можуть розробити більш надійні та зрозумілі моделі машинного навчання, які потенційно можуть допомогти у прийнятті обґрунтованих рішень на криптовалютному ринку.

1.3 Веб-сервіси для прогнозування котирувань криптовалют

1.3.1 Огляд та порівняльний аналіз функціонала

CryptoPredictions – це сайт криптовалютних прогнозів, який надає прогнози цін на понад 8 000 криптовалют. Він пропонує користувачам точні та актуальні прогнози цін на широкий спектр криптовалют. Його мета – допомогти трейдерам, інвесторам та ентузіастам зрозуміти потенційний майбутній рух цін і приймати обґрунтовані рішення на основі цих прогнозів [7].

Сайт збирає великі історичні ринкові дані та застосовує різні статистичні моделі та моделі машинного навчання для прогнозування майбутніх цін. Чим більше історичних даних про криптовалюту, тим вища точність прогнозів. Прогнози для криптовалют з ринковою капіталізацією оновлюються кожні п'ять хвилин, забезпечуючи користувачів найсвіжішими даними. Інші монети оновлюються щодня. Користувачі можуть легко шукати бажані криптовалюти й переглядати детальні графіки та дані прогнозів.

Унікальні функції:

- всі прогнози доступні безкоштовно, без необхідності підписки або реєстрації;
- користувачі можуть конвертувати всі прогнози та ринкові дані з USD в 32 інші валюти, що особливо корисно для міжнародних користувачів;
- сторінка прогнозу кожної криптовалюти містить графік, що візуалізує прогнозовані ціни з плином часу, допомагаючи користувачам краще зрозуміти потенційні тенденції;
- сайт охоплює величезну кількість криптовалют, включаючи такі популярні, як Bitcoin і Ethereum, а також менш відомі монети;
- CryptoPredictions підкреслює важливість проведення особистої перевірки та надає чіткі застереження щодо характеру своїх прогнозів.

Використовуючи ці особливості, CryptoPredictions надає комплексну і доступну платформу для всіх, хто цікавиться прогнозами цін на криптовалюти, дозволяючи користувачам приймати більш обґрунтовані інвестиційні рішення на основі поточних і прогнозованих ринкових даних.

LongForecast – це вебсайт, присвячений наданню довгострокових фінансових прогнозів для різних активів, включаючи криптовалюти. LongForecast пропонує детальні прогнози майбутніх цін на тривалі періоди [8].

LongForecast застосовує різні статистичні та математичні моделі, зосереджуючись на довгострокових тенденціях, а не на короткострокових коливаннях. На відміну від деяких інших сайтів, які часто оновлюють прогнози,

LongForecast зазвичай надає щомісячні прогнози, зосереджуючись на середньо- та довгострокових перспективах. Сайт охоплює широкий спектр активів, включаючи криптовалюти, акції, сировинні товари та обмінні курси.

Унікальні функції:

- LongForecast спеціалізується на наданні щомісячних прогнозів, пропонуючи детальний прогноз на кожен місяць протягом наступних кількох років. Це особливо корисно для довгострокових інвесторів, які хочуть зрозуміти потенційні майбутні тенденції;
- пропонує прогнози для традиційних фінансових ринків, що робить його комплексним ресурсом для диверсифікованих інвесторів;
- для кожного активу LongForecast надає діапазон прогнозованих цін, включаючи очікувані мінімальні, максимальні та середні ціни, а також відсоткову зміну від поточної ціни.

WalletInvestor – це онлайн-платформа, яка надає прогнози фінансового ринку та прогнози для різних активів, включаючи криптовалюти, акції, валютні пари, сировинні товари тощо [9].

WalletInvestor використовує передові алгоритмічні моделі та методи машинного навчання для аналізу історичних ринкових даних і виявлення тенденцій. Платформа генерує прогнози на основі різних статистичних моделей, враховуючи минулі показники і поточні ринкові умови. Прогнози часто оновлюються, а нові дані регулярно включаються в моделі для забезпечення точності. WalletInvestor надає як короткострокові, так і довгострокові прогнози, обслуговуючи як трейдерів, так і довгострокових інвесторів.

Унікальні особливості:

- сайт охоплює широкий спектр фінансових інструментів, включаючи понад 100 000 активів, таких як криптовалюти, акції, валютні пари, сировинні товари та ETF;
- для кожного активу WalletInvestor надає детальні прогнози, включаючи очікувані діапазони цін, середні ціни та потенційні ринкові тенденції;

- користувачі можуть отримати доступ до прогнозів для різних часових горизонтів, починаючи від щоденних прогнозів і закінчуючи п'ятирічними прогнозами.

Платформа має інтуїтивно зрозумілий і зручний інтерфейс, що дозволяє користувачам легко орієнтуватися і знаходити потрібну інформацію. Візуальні інструменти, такі як діаграми та графіки, допомагають користувачам зрозуміти прогнозовані тенденції та приймати обґрунтовані рішення.

WalletInvestor пропонує різні інструменти та ресурси, включаючи інвестиційні калькулятори, функції управління портфелем і освітній контент, щоб допомогти користувачам поліпшити свої торгові та інвестиційні стратегії. Платформа також надає аналіз ризиків і прогнози прибутковості, даючи користувачам уявлення про потенційні ризики і винагороди, пов'язані з різними інвестиціями.

Порівняння платформ подано в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика веб-сервісів для прогнозування котирувань криптовалют

Сайт	Частота прогнозів	Покриття активів	Користувацький досвід
WalletInvestor	надає оновлення в режимі реального часу	охоплює широкий спектр активів, включаючи криптовалюту, акції, форекс, сировинні товари та ETF	комплексний і зручний інтерфейс з детальними прогнозами, графіками та освітніми ресурсами
LongForecast	зосереджується на щомісячних прогнозах	охоплює криптовалюту, акції, товари та обмінні курси	надає прості щомісячні прогнози з детальними прогнозними даними

Продовження таблиці 1.1

CryptoPredictions	оновлює прогнози для основних криптовалют кожні п'ять хвилин	охоплює понад 8 000 цифрових активів	надає динамічні, часто оновлювані прогнози з конвертером валют і детальними графіками
-------------------	--	--------------------------------------	---

Користувачі можуть взаємодіяти зі спільнотою через форуми та дискусійні дошки, обмінюючись думками та стратегіями. WalletInvestor пропонує клієнтську підтримку, щоб допомогти користувачам з будь-якими проблемами або питаннями, які у них можуть виникнути.

Отже, у висновку можемо сказати, що WalletInvestor – це універсальна і всеосяжна платформа для прогнозування фінансових ринків, орієнтована як на короткострокових трейдерів, так і на довгострокових інвесторів. Завдяки широкому спектру активів, частим оновленням і детальним прогнозам, вона надає цінну інформацію, яка допомагає користувачам приймати обґрунтовані інвестиційні рішення.

Криптовалюти, з моменту їхнього виникнення, продовжують трансформувати фінансовий ландшафт, пропонуючи нові можливості для інвесторів та користувачів у всьому світі. У цьому розділі було розглянуто різні аспекти криптовалют, від їх визначення та історії до технічних характеристик та класифікації основних видів. З'ясувалося, що криптовалюти, засновані на децентралізованих блокчейн-технологіях, надають користувачам безпечний спосіб здійснення транзакцій без посередників, що сприяє розвитку P2P-транзакцій.

Прогнозування котирувань криптовалют є складним завданням, яке вимагає використання різних методів аналізу. Технічний аналіз, ґрунтуючись на історичних даних та графічних паттернах, допомагає трейдерам ідентифікувати потенційні розвороти ринку. Фундаментальний аналіз, зі свого боку, враховує як внутрішні, так і зовнішні фактори, такі як економічні умови та фінансові

показники, що впливають на вартість активів. Машинне навчання, використовуючи потужні алгоритми, пропонує інноваційні методи для прогнозування ринкових тенденцій та цінових рухів. Кероване навчання, зокрема методи, такі як Support Vector Regression та Random Forest, показали високу ефективність у передбаченні майбутніх цін на основі аналізу великих обсягів даних.

Особливу увагу слід приділити веб-сервісам для прогнозування котирувань криптовалют. Платформи, такі як CryptoPredictions, LongForecast та WalletInvestor, надають корисні інструменти та ресурси для аналізу ринку. Вони пропонують різні моделі прогнозування, включаючи короткострокові та довгострокові прогнози, і забезпечують користувачів актуальною інформацією, яка допомагає приймати обґрунтовані інвестиційні рішення. Огляд та порівняльний аналіз цих сервісів підкреслює їхню важливість для трейдерів та інвесторів, які шукають надійні джерела прогнозів та аналізу ринкових даних.

Отже, сучасний стан досліджень у сфері криптовалют демонструє значний прогрес та постійний розвиток. Використання різних методів прогнозування дозволяє отримувати більш точне розуміння складної динаміки ринку. Веб-сервіси для прогнозування криптовалют надають корисні інструменти для аналізу ринку, допомагаючи трейдерам та інвесторам приймати обґрунтовані рішення.

У результаті проведення дослідження в першому розділі представлено всебічний огляд ринку криптовалют, зокрема визначено основні терміни, характеристики криптовалютного ринку та класифіковано популярні криптовалюти. Описано специфіку ринку та чинники, що впливають на вартість криптовалют. Також проаналізовано методи прогнозування вартості, включаючи технічний і фундаментальний аналіз, а також використання машинного навчання. Проведено огляд наявних веб-сервісів для прогнозування криптовалют та їх порівняльний аналіз.

РОЗДІЛ 2

ТЕХНОЛОГІЧНИЙ СТЕК ТА МОДЕЛІ ДЛЯ РОЗРОБКИ

2.1 Вибір технологій для розробки веб-сервісу

2.1.1 Flask фреймворк для розробки API

Flask це легкий веб-фреймворк та є чудовим вибором для створення API завдяки своїй простоті та гнучкості. При розробці веб-сервісу для прогнозування криптовалют мінімалістична структура Flask забезпечує швидке розгортання і легке масштабування, що робить його ідеальним для роботи з динамічною природою криптовалютних даних. У своїй книзі "Біткойн і криптовалютні технології" Нараянан та ін. [11] обговорюють важливість розуміння технології, що лежить в основі криптовалют, таких як біткойн, яка може бути використана при створенні моделей прогнозування для крипторинку. На відміну від складніших фреймворків, простота використання Flask дозволяє розробникам зосередитися на основній функціональності свого веб-сервісу, не занурюючись у непотрібні функції. Крім того, сумісність Flask з такими бібліотеками для обробки даних, як NumPy та Pandas, робить його універсальним інструментом для реалізації алгоритмів машинного навчання для покращення криптовалютних прогнозів. Для порівняння, Flask виділяється серед інших веб-фреймворків для розробки API, пропонуючи простий підхід, який ставить на перше місце ефективність і швидкість розробки [12].

Хоча зберігання невеликих веб-додатків в одному файлі сценарію може бути зручним, такий підхід значно ускладнює масштабування. Зі зростанням складності додатка робота з одним великим вихідним файлом стає все більш проблематичною. Flask не накладає жорстких вимог щодо організації великих проєктів; структура додатка повністю залишається на розсуд розробника.

2.1.2 Технології веб-розробки для веб-застосунків

При розробці веб-сервісу для прогнозування криптовалют використання різних фронтенд-технологій, таких як React, JavaScript, HTML, CSS та Chart.js, відіграє вирішальну роль у покращенні користувацького досвіду та візуалізації даних. Ефективно інтегруючи ці технології, розробники можуть створити зручну для користувача платформу, яка не тільки надає точні прогнози, але й представляє дані у візуально привабливому вигляді.

Як згадується в статті "РОЗРОБКА ТА ВПРОВАДЖЕННЯ ВЕБ-ДОДАТКУ ДЛЯ КОНСУЛЬТАЦІЇ КРИПТОІНВЕСТОРІВ" Анурага Джоші та ін. [13], метою є використання моделей машинного навчання та методів нормалізації для аналізу історичних даних та прогнозування майбутніх цін на криптовалюту з високою точністю. Це підкреслює важливість вибору правильних фронтенд-технологій, щоб забезпечити користувачам доступ до надійної інформації та інтуїтивно зрозумілих інструментів візуалізації даних для прийняття обґрунтованих інвестиційних рішень.

Вибір Chart.js як основного інструменту для візуалізації даних у веб-сервісі для прогнозування криптовалют є обґрунтованим завдяки його гнучкості, продуктивності та інтерактивності. Ця бібліотека дозволяє розробникам створювати інтуїтивно зрозумілі та візуально привабливі графіки, що сприяє покращенню користувацького досвіду та підтримці прийняття обґрунтованих інвестиційних рішень на основі точних даних.

Chart.js легко інтегрується з обраною бібліотекою React, оскільки має відповідний модуль інтеграції.

2.1.3 Tensorflow, Pandas, та, Sklearn для обробки даних та машинного навчання

Tensorflow - це бібліотека з відкритим вихідним кодом, розроблена компанією Google, яка використовує її в основному для застосунків глибокого навчання. Він додатково надає підтримку. TensorFlow спочатку був розроблений для використання в машинному навчанні внутрішньо в компанії Google, він

виявився дуже корисним для розробки додатків глибинного навчання, тому компанія вирішила зробити його з відкритим вихідним кодом.

TensorFlow приймає дані у вигляді багатовимірних масивів великих розмірів. Оскільки механізм роботи Tensorflow має вигляд графів, його набагато легше зрозуміти. Код TensorFlow може виконуватись розподілено на GPU.

ta - це бібліотека Python для технічного аналізу фінансових ринків.

Функціонування фінансових ринків. Містить набір базових технічних індикаторів, що використовуються при технічному аналізі ринку. Використовується для аналізу фінансових даних та розробки технічних індикаторів. Бібліотека складається з понад 150 індикаторів.

Такі як індикатори середніх цін, індикатори руху цін та індикатори обсягу. Завдяки бібліотеці ТА користувачі можуть швидко і легко будувати прогнози. Вона забезпечує підтримку популярних форматів фінансових даних, таких як Pandas DataFrame або масиви NumPy.

Pandas - бібліотека для обробки та аналізу даних на мові Python.

Мова програмування Python. Підтримуються такі формати, як CSV, Excel, SQL та бази даних. Бібліотека дозволяє якісно читати, фільтрувати, трансформувати та агрегувати дані. Основною структурою даних в Pandas є DataFrame, який є табличною структурою. Таблиця, що містить рядки та стовпці даних. DataFrame дозволяє виконувати прості операції.

Scikit-learn, також відома як Sklearn - це бібліотека машинного навчання, призначена для освітнього сектору. Вона надає різноманітні реалізації алгоритмів машинного навчання. Sklearn містить багато модулів, кожен з яких присвячений певному типу навичок. Sklearn - це потужна і корисна бібліотека для машинного навчання, що полегшує створення та оцінку моделей, тим самим допомагаючи в їхньому розвитку та надає можливість швидко та ефективно аналізувати дані.

2.2. LSTM як моделі для прогнозування криптовалют

2.2.1 Переваги та недоліки LSTM-моделі

Завдяки тому, що LSTM має внутрішню структуру з "комірок" та "вентилів" (рисунок 2.1), ця модель особливо корисна у фінансовому середовищі. Три типи вентилів: введення, виведення та забування, є досить точними у прогнозуванні цін та визначенні майбутніх тенденцій, що було показано у роботі Джозефа К. [15].

Вхідний вентиль регулює введення свіжих входів в комірку пам'яті. Вентиль забування вирішує, яку частину попереднього стану пам'яті має бути забута коміркою пам'яті. Вихідний вентиль регулює виведення станів комірки пам'яті, і, нарешті, комірка пам'яті відповідає за збереження даних і передачу їх на наступний часовий крок. Активація тангенса і сигмоїди використовуються для створення LSTM вентилів.

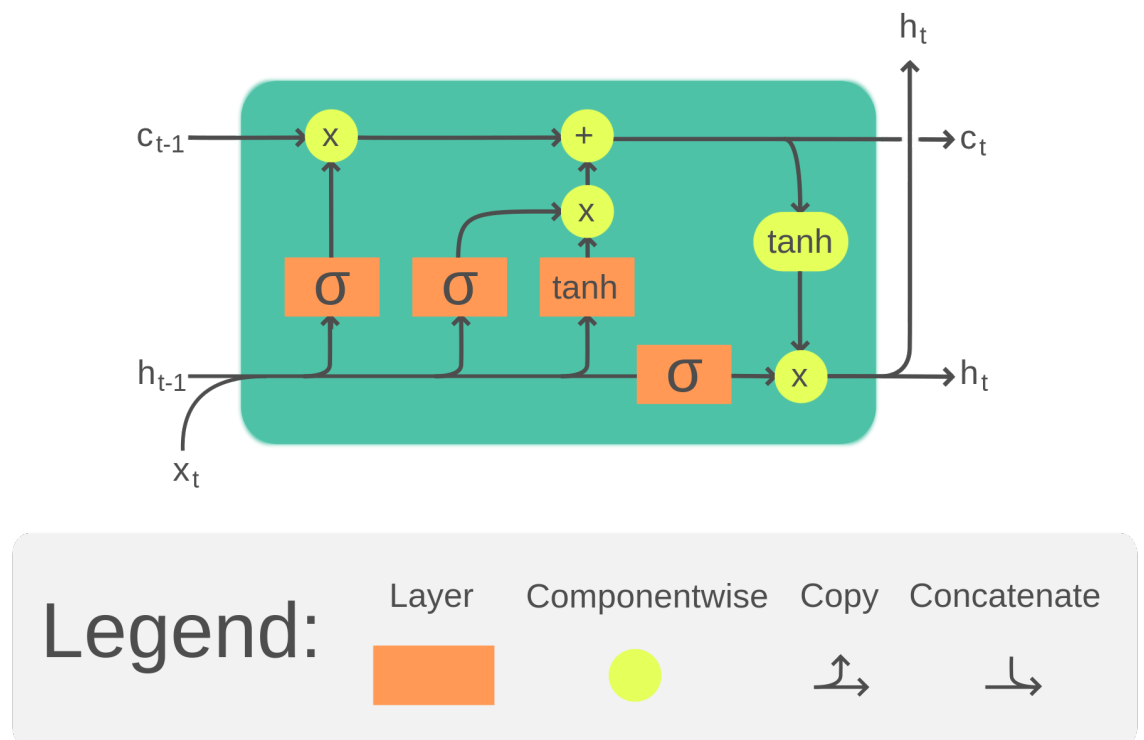


Рисунок 2.1 – Схема моделі довго короткочасної пам'яті

Сигмоїдна функція (Рисунок 2.2) повертає значення від 0 до 1, щоб представити відкритий або закритий стан вентилів. Функція tanh (Рисунок 2.2)

повертає значення від -1 до 1, щоб представити напруженість комірки пам'яті стан комірки.

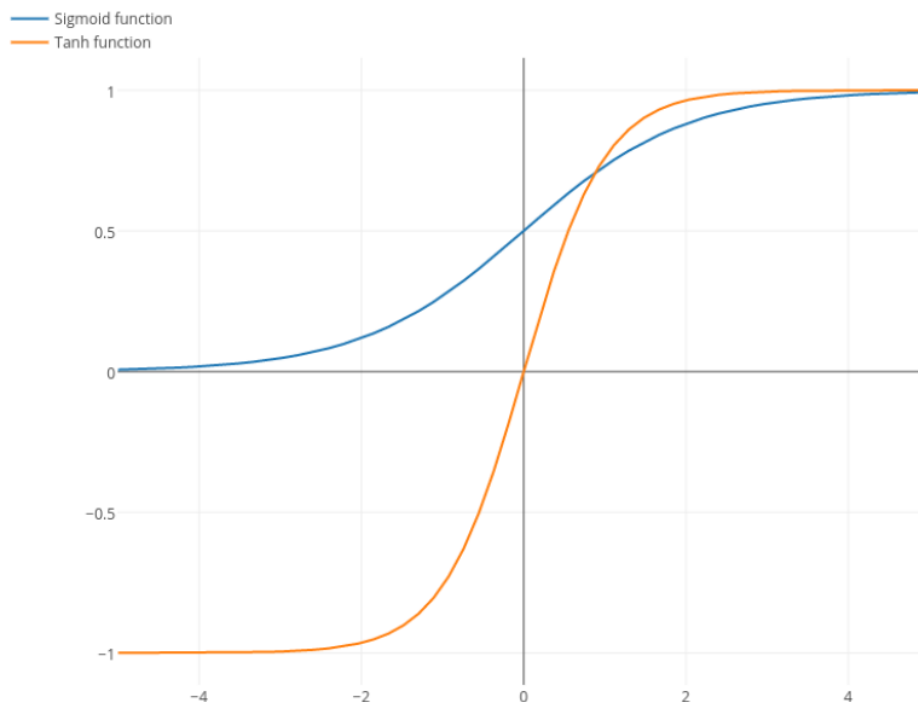


Рисунок 2.2 – Схема активаторної функції моделі LSTM

Для мінімізації функції втрат використовується зворотне розповсюдження в часі (BPTT), оскільки LSTM навчається для визначення найкращих значень її параметрів. Параметри включають вагові матриці та вектори зсуву кожного вентиля разом з вихідним шару.

Відсікання градієнта можна використовувати для обмеження розміру градієнта під час навчання, оскільки LSTM все ще може страждати від зникаючих градієнтів, якщо градієнт занадто малий, і від вибухових градієнтів, якщо градієнт занадто великий. Для того, щоб передбачити майбутні LSTM можна використовувати для виявлення закономірностей в історичних цінових даних. Функція втрат, яка обчислює різницю між прогнозованими та фактичними цінами оптимізується для навчання LSTM.

Одна з переваг LSTM моделі в прогнозуванні котирувань криптовалют – обробка довгих послідовних залежностей.

LSTM можуть вловлювати довгострокові зв'язки в рядах даних, що є корисним для нестабільних і часто безсистемних коливань цін на криптовалюти, які здаються непередбачуваними.

Також LSTM відома стійкістю до зміни довжини послідовності.

На відміну від багатьох інших моделей прогнозування, ефективність LSTM залишається незмінною при різній довжині вхідних даних, що робить його універсальним для використання з різними обсягами історичних даних [14].

Як правило, LSTM забезпечують кращу точність прогнозування порівняно з простішими моделями часових рядів, такими як ARIMA або експоненціальне згладжування, коли мають справу зі складними наборами даних, такими як ціни на криптовалюти.

Крім того, LSTM модель має свої недоліки. По-перше, це обчислювальна інтенсивність. LSTM вимагають значних обчислювальних ресурсів для навчання, особливо зі збільшенням обсягу даних і складності архітектури моделі.

По-друге, завдяки своїй глибокій та складній архітектурі LSTM можуть легко перенавчати навчальні дані, особливо якщо недостатньо навчальних даних або якщо модель не є ефективно регуляризованою.

По-третє, продуктивність LSTM-моделей дуже чутлива до конфігурації їхніх гіперпараметрів, таких як кількість шарів, кількість одиниць у кожному шарі та швидкість навчання. Пошук оптимального набору гіперпараметрів може зайняти багато часу і вимагає тривалих експериментів.

В моїй роботі модель LSTM налаштована для прогнозування цін на криптовалюту, зокрема, на біткоїн (BTC). Конфігурація передбачає послідовний підхід, який використовує історичні цінові дані для прогнозування майбутніх значень. В даній моделі було використано методи нормалізації вхідних даних. Це було необхідно для точного навчання моделі, оскільки параметри ціни криптовалюти досить варіативні та на довгостроковій вибірці з 2017 року можуть призвести до дефектів моделі.

За архітектурою LSTM модель структурована за допомогою Keras, з послідовним компонуванням [15]. Ядром моделі є шар LSTM з розмірністю

вихідного простору 10 і активаційною функцією зрізаний зворотний вузол, який добре підходить для вловлювання нелінійних патернів у русі цін. Далі йде щільний шар, який виводить прогнозовану ціну.

Модель навчається на значній кількості епох (100) і з розміром партії 32, що має на меті оптимізувати процес навчання на великому наборі даних. Дані розділено на навчальний та тестовий набори. Також, 20% даних зарезервовано для тестування, щоб оцінити продуктивність моделі.

Вхідні дані для LSTM переформатовані, щоб включити 15 часових кроків, що означає, що модель буде дивитися на ціни за останні 15 днів, щоб зробити прогноз на наступний день. Такий розмір вікна створює баланс між необхідністю врахувати останні тенденції та перевантаженням моделі занадто великою кількістю інформації та перенавчанням. Після навчання ефективність моделі оцінюється за допомогою декількох метрик, включаючи значення RMSE, MSE, MAE та R-квадрат [16]. Ці показники допомагають зрозуміти точність і надійність прогнозів. Для майбутніх прогнозів скрипт дозволяє передбачити ціни на певну кількість днів за межами наявних даних. Він використовує останні відомі точки даних як базу для ітеративної генерації прогнозів. Ця конфігурація реалізовує надійну основу для управління волатильністю цін на криптовалюту, використовуючи сильні сторони LSTM в послідовному прогнозуванні, щоб потенційно дати уявлення про майбутні рухи ринку.

2.2.2 Порівняння LSTM-моделі з іншими моделями машинного навчання

Як було показано в детальній роботі Sebastião та Godinho [17], модель LSTM працює краще, зокрема завдяки своїй здатності фіксувати довгострокові залежності в даних часових рядів, що має ключове значення для непостійних рухів та моделей, які спостерігаються на ринках криптовалют. Навпаки, такі моделі, як ARIMA, традиційно використовуються для прогнозування часових рядів, але не мають можливості моделювати складні шаблони, які включають нелінійність, типову для наборів фінансових даних, зокрема велику волатильність криптовалют. SVM і випадкові ліси, попри надійність у різних

завданнях прогнозування, за своєю суттю не враховують характер часових рядів даних, але часто використовуються в методах ансамблем, разом з LSTM для підвищення точності прогнозування.

За результатами робіт Sebastião та Godinho [17] модель LSTM показала чудові характеристики з погляду точності. Вона послідовно передбачала ціни з меншою середньоквадратичною помилкою (MSE) порівняно з ARIMA, SVM і випадковими лісами. Здатність LSTM обробляти та запам'ятовувати інформацію протягом довгих послідовностей дозволяє йому перевершувати продуктивність, особливо в нестабільних фазах, коли минула інформація суттєво впливає на майбутні ціни.

Попри необхідність для моделі LSTM обчислювальних потужностей, більших у порівнянні з іншими моделями прогнозування, останні досягнення в паралельних обчисленнях і використанні GPU значно пом'якшили ці проблеми. Навпаки, моделі ARIMA, будучи менш складними, швидше навчаються, але не так добре працюють з великими наборами даних і нелінійними шаблонами. Моделі SVM і Random Forest також продемонстрували достатній час навчання, але відсутність у них можливості послідовної обробки даних часто вимагає додаткових кроків для прогнозування часових рядів, що не завжди показує високу точність. З погляду надійності, LSTM перевершив інші моделі в періоди високої волатильності. Моделі ARIMA, як правило, відставали в продуктивності в такі періоди, тоді як SVM і випадкові ліси, хоча загалом стабільні, менш реагували на раптові зміни ринку, що часто призводило до затримки коригування прогнозів.

Проте одиничне використання LSTM не є найкращим вибором для прогнозування котирувань криптовалют. Чинна література виокремлює низьку ефективність прогнозування окремих моделей у валідаційній вибірці, яка свідчить про те, що збірка моделей дає найкращі результати. Дослідження аналізують прибутковість на основі торгових стратегій, які ініціюють довгі позиції, якщо щонайменше чотири, п'ять або шість окремих моделей збігаються щодо позитивного торгового сигналу на наступний день. Ці стратегії

розглядають лише довгі позиції, враховуючи складність або неможливість коротких продажів на криптовалютних ринках. Це обмеження є суттєвим, особливо з огляду на те, що період тестування характеризувався низхідною тенденцією ринків з середньоденною прибутковістю нижче $-0,20\%$.

2.2.3 Відповідність LSTM-моделі для прогнозування часових рядів

LSTM-мережі пом'якшують проблему зникаючого градієнта, характерну для стандартних рекурентних нейронних мереж (RNN), дозволяючи їм навчатися і запам'ятовувати інформацію протягом тривалих періодів. Ця особливість особливо корисна для моделювання фінансових ринків, де минулі події суттєво впливають на майбутні тенденції.

Кілька емпіричних досліджень [18][19][20][21] продемонстрували надійність LSTM-моделей у прогнозуванні цін на криптовалюту. Наприклад, дослідження показало, що LSTM можуть ефективно моделювати і прогнозувати дуже волатильні цінові рухи біткойна з більшою точністю, ніж традиційні моделі, такі як ARIMA і стандартні RNN [22]. Аналогічно, інше дослідження показало, що інтеграція зовнішніх факторів, таких як тенденції пошуку та активність блокчейну, в LSTM-моделі підвищує їхню прогностичну точність, що свідчить про здатність моделі ефективно включати допоміжну інформацію [23].

Порівняння LSTM з іншими моделями прогнозування підкреслює перевагу першої в обробці нелінійних моделей та зашумлених наборів даних, типових для криптовалютних ринків. Наприклад, моделі GARCH і SVM, хоча і є надійними на стабільних фінансових ринках, не так добре працюють у хаотичних умовах торгівлі криптовалютами [24].

Огляд літератури підкреслює придатність LSTM-моделей для прогнозування цін на криптовалюту завдяки їхній здатності вивчати довгострокові залежності та обробляти нелінійну природу даних. Хоча існують такі проблеми, як попит на обчислювальні ресурси та потенційне перенавчання, постійний розвиток нейромережевих методологій продовжує зміцнювати позиції

LSTM як цінного інструменту в арсеналі методів фінансового прогнозування [25].

Отже у результаті виконання другого розділу роботи було докладно розглянуто технології для розробки веб-сервісів, зокрема Flask для створення API, а також різноманітні фронтенд-технології (React, JavaScript, HTML, CSS, Chart.js) для формування зручного користувацького інтерфейсу. Крім того, обговорюється застосування бібліотек TensorFlow, Pandas та Sklearn для обробки даних та реалізації методів машинного навчання. Особлива увага приділялась моделі LSTM для прогнозування криптовалют, її перевагам та недолікам.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ КОТИРУВАНЬ КРИПТОВАЛЮТ

3.1 Архітектура веб-сервісу

3.1.1 Опис компонентів веб-сервісу та їх взаємодії

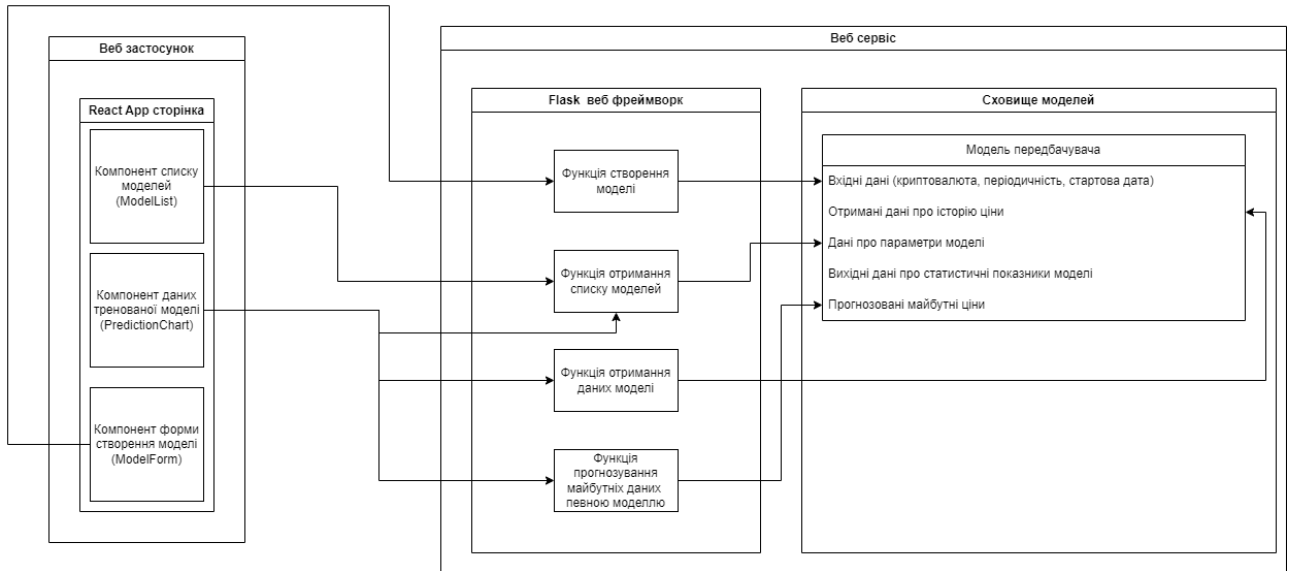


Рисунок 3.1 – Архітектура веб сервісу

Веб-сервіс для прогнозування котирувань на ринку криптовалют складається з трьох основних компонентів: бекенду (backend), фронтенду (frontend) та модулю машинного навчання (ML). Кожен з цих компонентів має свою роль та завдання, та взаємодіє з іншими компонентами для роботи системи.

Бекенд сервісу розроблений на основі Flask, легкого веб-фреймворку для Python. Основні функції бекенду включають обробку запитів від користувачів, управління даними та взаємодію з модулем машинного навчання. Компоненти бекенду реалізовані у файлі **main.py**.

Бекенд використовує бібліотеки **pandas** [35] для обробки даних, **joblib** для збереження та завантаження моделей, **yfinance** для отримання даних про котирування та **threading** для асинхронного навчання моделей.

Фронтенд реалізований за допомогою React, популярної бібліотеки JavaScript для побудови користувацьких інтерфейсів. До складу фронтенду також входять HTML, CSS для створення стилів та Chart.js для візуалізації даних.

Інтерфейс забезпечує взаємодію користувача з веб-сервісом. Основні функції включають форму для створення нових моделей, відображення статусу поточних моделей та графічне представлення прогнозів.

Таблиця 3.1 – Опис ендпоінтів веб-сервісу

Ендпоінт	Опис ендпоінту
/models	Приймає POST-запити, що містять інформацію про тикер криптовалюти, інтервал даних, дату початку, кількість епох навчання та інші параметри. Цей ендпоінт ініціалізує процес навчання нової моделі та зберігає модель до сховища.
/models/<model_name>	Приймає GET-запити для перевірки статусу конкретної моделі (наприклад, "training" або "ready").
/models/<model_name>/data	Повертає дані, що використовуються для навчання моделі зі сховища.
/model/<model_name>/predict	Приймає POST-запити з параметрами для прогнозування. Повертає прогнозовані значення котирувань на задану кількість днів уперед.
/models	Приймає GET-запити, що повертають список всіх моделей зі сховища та їх статуси.

Для відображення історичних даних та прогнозів використовується Chart.js, що дозволяє створювати інтерактивні графіки.

Фронтенд взаємодіє з бекендом через RESTful API, що дозволяє здійснювати асинхронні запити та оновлювати інтерфейс користувача без перезавантаження сторінки.

Модуль машинного навчання

Модуль машинного навчання включає моделі для прогнозування котирувань на основі алгоритму довготривалої короткочасної пам'яті (LSTM). Основні інструменти для побудови та навчання моделей включають TensorFlow, scikit-learn, NumPy та Pandas.

Для підготовки даних використовується pandas, що дозволяє маніпулювати великими наборами даних, а уfinance забезпечує отримання історичних даних про котирування криптовалют.

Моделі LSTM навчаються за допомогою TensorFlow, що забезпечує високоефективне обчислення та можливість налаштування моделей для досягнення оптимальних результатів [27].

Після навчання моделі використовуються для прогнозування майбутніх котирувань на основі вхідних даних. Результати прогнозів передаються до бекенду та відображаються на фронтенді.

Компоненти сервісу взаємодіють один з одним відповідно до рис. 3.1, а саме:

1. Обробка запитів бекенду через RESTful API. Бекенд обробляє запити, викликаючи відповідні функції для обробки даних або навчання моделей.
2. Бекенд взаємодіє з модулем машинного навчання для підготовки даних, навчання моделей та отримання прогнозів.
3. Прогнозовані дані повертаються до фронтенду, де вони візуалізуються за допомогою Chart.js та відображаються користувачу.

Таким чином, веб-сервіс забезпечує інтегровану систему для прогнозування котирувань на ринку криптовалют, використовуючи сучасні технології веб-розробки та машинного навчання.

3.1.2 Дані, що використовуються для прогнозування

Для побудови точних моделей прогнозування котирувань на ринку криптовалют необхідно використовувати якісні та репрезентативні дані. Дані є основою для навчання моделей машинного навчання, і від їх якості залежить

точність прогнозів. У даному підрозділі розглянемо типи даних, джерела їх отримання та методи попередньої обробки залучені у моїй роботі.

Дані, що використовуються для прогнозування котирувань криптовалют, можна розділити на кілька категорій:

1. Історичні котирування, що є основними даними на яких базується прогнозування. Вони включають ціни відкриття, закриття, максимальні та мінімальні ціни за певний період, а також обсяг торгів. Ці дані дозволяють моделі виявляти патерни та тренди.
2. Технічні індикатори які розраховуються на основі історичних котирувань і використовуються для технічного аналізу. Серед них можна виділити середнє ковзне (MA), відносну силу (RSI) та інші. Технічні індикатори допомагають у визначенні тенденцій та можливих змін [28].
3. Дані, що характеризують загальний стан економіки, можуть впливати на ринок криптовалют. Наприклад, індекси інфляції, безробіття, ставки центральних банків тощо.
4. Інформація з соціальних мереж, новини та інші джерела, що можуть впливати на настрої інвесторів [29].

З усіх виділених даних мною у виконаній роботі було використано ціну закриття та середнє ковзне.

Для отримання вищезгаданих типів даних використовується **yfinance** для завантаження даних про котирування. Сервіс `yahoo finance` було обрано, оскільки він дозволяє отримати досить точні історичні дані для великого переліку торговельний пар.

Для розрахунку середнього ковзного використовуються бібліотека для технічного аналізу **ta**.

Попередня обробка даних є критично важливим етапом перед навчанням моделей [30], [31]. Основні кроки включають:

1. Чищення даних, видалення пропусків, виправлення аномалій та некоректних значень. Наприклад, відсутні дані можуть бути заповнені

методами інтерполяції, а екстремальні значення, що можуть впливати на модель, можуть бути відфільтровані.

2. Нормалізація, яка включає масштабування значень до певного діапазону, наприклад $[0, 1]$, що покращує стабільність та швидкість навчання моделей.
3. Розділення даних на тренувальні та тестові набори, що дозволяє оцінити продуктивність моделі та запобігти перенавчанню.

Усі перелічені кроки були використані в моїй роботі для попередньої обробки даних.

Таким чином, використання якісних даних та їх коректна попередня обробка є ключовими факторами для побудови точних та надійних моделей прогнозування котирувань на ринку криптовалют [32].

3.2 Реалізація backend-частини веб-сервісу

3.2.1 Розробка API для взаємодії з веб-сервісом

Розробка API для взаємодії з веб-сервісом є критичним елементом у забезпеченні функціональності та зручності користування системою прогнозування котирувань криптовалют. Основна мета API полягає в наданні програмного інтерфейсу, який дозволяє клієнтам (наприклад, веб- або мобільним додаткам) взаємодіяти з сервісом, виконуючи запити на створення моделей, отримання статусу моделей, завантаження даних та отримання прогнозів. Для реалізації API у даному проєкті використовується фреймворк Flask, який забезпечує легкість і гнучкість у розробці веб-додатків на Python.

Stock Price Predictor API 1.0.0 OAS 3.0

An API for training and predicting stock prices

Servers

http://localhost:5000

default

POST

/models Create a new model

GET

/models List all models

GET

/models/{model_name} Get details of a specific model

POST

/model/{model_name}/predict Make predictions using a trained model

Рисунок 3.2 – Опис кінцевих точок

API реалізовано з використанням архітектурного стилю REST, який є широко поширеним завдяки своїй простоті та масштабованості, візуалізацію API можна побачити на рисунку 3.2. REST API дозволяє здійснювати операції CRUD (Create, Read, Update, Delete) через HTTP-запити, що робить його зручним для інтеграції з різноманітними клієнтськими додатками. Для забезпечення безпеки та сумісності запитів API підтримує механізм Cross-Origin Resource Sharing (CORS), що дозволяє обмінюватися даними між клієнтами та сервером, які розміщені на різних доменах.

POST /models Create a new model

Parameters Try it out

No parameters

Request body required application/json

Example Value | Schema

```
{
  "ticker": "BTC-USD",
  "interval": "1d",
  "start_date": "2024-05-15",
  "epochs": 100,
  "batch_size": 32,
  "test_part": 0.2
}
```

Responses

Code	Description	Links
200	Successfully created model	No links

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "model_status": "ready",
  "model_name": "string"
}
```

Рисунок 3.3 – Опис функції створення нових моделей

Основна функція API для створення нових моделей прогнозування реалізована через ендпоінт (рисунок 3.3), який приймає POST-запити. Користувачі можуть надсилати JSON-запити з параметрами, такими як тикер криптовалюти, інтервал даних, дата початку, кількість епох для навчання, розмір батчу та частка даних для тестування. Після отримання запиту сервер ініціалізує процес створення нової моделі, зберігає її параметри у внутрішній структурі даних та розпочинає навчання моделі у фоновому режимі. Навчання моделі здійснюється асинхронно, що дозволяє користувачам продовжувати роботу з системою без очікування завершення цього процесу.

GET /models/{model_name} Get details of a specific model

Parameters Try it out

Name	Description
model_name * required	
string (path)	model_name

Responses

Code	Description	Links
200	Details of the model	No links
404	Model not found	No links

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "status": "string",
  "model": {
    "epochs": 100,
    "batch_size": 32,
    "interval": "string",
    "start_date": "string",
    "test_part": 0,
    "test_stats": {
      "test_explained_variance": 0,
      "test_mae": 0,
      "test_mgd": 0,
      "test_mpd": 0,
      "test_mse": 0,
      "test_r2": 0,
      "test_rmse": 0
    },
    "ticker": "string",
    "time_step": 0,
    "train_stats": {
      "train_explained_variance": 0,
      "train_mae": 0,
      "train_mgd": 0
    }
  }
}
```

Рисунок 3.4 – Опис функції отримання деталей певної моделі

Для отримання статусу створених моделей API надає GET-запити до відповідного ендпоинта (рисунок 3.4). Кожна модель має унікальний ідентифікатор, який використовується для доступу до її статусу та параметрів. Це дозволяє користувачам перевіряти, чи готова модель до використання, чи вона ще знаходиться в процесі навчання. Крім того, API дозволяє отримувати дані, що використовувалися для навчання моделей. Це реалізовано через GET-запити до спеціального ендпоинта, який повертає історичні дані котирувань у

форматі JSON. Ця функція забезпечує прозорість процесу моделювання та дозволяє користувачам аналізувати вхідні дані.

Окремий ендпоинт API реалізує функцію прогнозування. Користувачі можуть надсилати POST-запити з параметрами, такими як кількість днів для прогнозування. Якщо модель готова, сервер обробляє запит, викликає функцію прогнозування і повертає результат у форматі JSON, що включає прогнозовані значення котирувань на заданий період. У випадку, якщо модель ще не готова або не знайдена, API повертає відповідне повідомлення про помилку.

Для полегшення розробки та тестування API задокументовано за допомогою Swagger, що надає зручний інтерфейс для тестування та перегляду всіх доступних ендпоинтів та їх параметрів. Це допомагає розробникам швидко інтегрувати API у свої додатки, а також забезпечує можливість автоматичного генерування клієнтського коду для різних мов програмування.

3.2.3. Навчання та збереження LSTM-моделі

Процес навчання моделі розпочинається з отримання даних. Для цього використовується функція **fetch_data**, яка завантажує історичні дані про котирування обраної криптовалюти з початкової дати, заданої користувачем, до поточного моменту з вказаним інтервалом. Після завантаження дані очищаються від пропусків та залишаються лише значення закриття котирувань. Дані нормалізуються за допомогою **MinMaxScaler**, щоб значення знаходилися в діапазоні від 0 до 1, що покращує продуктивність моделі.

Після підготовки даних вони розділяються на тренувальний та тестовий набори. Це дозволяє оцінити продуктивність моделі на нових, невідомих даних, що є критично важливим для запобігання перенавчанню. Тренувальний набір використовується для навчання моделі, тоді як тестовий набір дозволяє перевірити, наскільки добре модель здатна узагальнювати нові дані.

Для створення тренувальних та тестових наборів даних використовується функція **create_dataset**, яка формує вхідні послідовності та відповідні цільові значення. Вхідні послідовності формуються з використанням ковзного вікна, що

дозволяє моделі враховувати попередні значення для прогнозування наступного. Ці послідовності потім передаються до LSTM-моделі для навчання.

Архітектура моделі включає один шар LSTM з 10 нейронами, за яким слідує повнозв'язний шар з одним нейроном для прогнозування. Модель компілюється з використанням функції втрат **mean_squared_error** та оптимізатора **adam**, що забезпечує ефективне навчання. Навчання моделі здійснюється протягом заданої кількості епох з певним розміром батчу.

```

Epoch 89/100
49/49 ██████████ 0s 2ms/step - loss: 4.3608e-04 - val_loss: 5.6402e-04
Epoch 90/100
49/49 ██████████ 0s 2ms/step - loss: 4.2415e-04 - val_loss: 9.3653e-04
Epoch 91/100
49/49 ██████████ 0s 3ms/step - loss: 5.0293e-04 - val_loss: 4.6989e-04
Epoch 92/100
49/49 ██████████ 0s 3ms/step - loss: 3.9809e-04 - val_loss: 6.3318e-04
Epoch 93/100
49/49 ██████████ 0s 3ms/step - loss: 4.2420e-04 - val_loss: 5.0852e-04
Epoch 94/100
49/49 ██████████ 0s 3ms/step - loss: 4.4734e-04 - val_loss: 5.9153e-04
Epoch 95/100
49/49 ██████████ 0s 3ms/step - loss: 4.6795e-04 - val_loss: 5.2248e-04
Epoch 96/100
49/49 ██████████ 0s 2ms/step - loss: 4.5937e-04 - val_loss: 4.5782e-04
Epoch 97/100
49/49 ██████████ 0s 3ms/step - loss: 4.0683e-04 - val_loss: 4.4338e-04
Epoch 98/100
49/49 ██████████ 0s 3ms/step - loss: 4.0124e-04 - val_loss: 4.9204e-04
Epoch 99/100
49/49 ██████████ 0s 3ms/step - loss: 4.3335e-04 - val_loss: 4.7289e-04
Epoch 100/100
49/49 ██████████ 0s 4ms/step - loss: 3.6946e-04 - val_loss: 4.9238e-04
49/49 ██████████ 0s 4ms/step
12/12 ██████████ 0s 1ms/step
Train predicted data: (1961, 1)
Test predicted data: (1961, 1)

```

Рисунок 3.5 – Вивід результату навчання моделі у консоль

Після завершення навчання модель проходить оцінку на тренувальних та тестових даних. Основні метрики, такі як корінь середньоквадратичної помилки (RMSE), середньоквадратична помилка (MSE), середня абсолютна помилка (MAE), коефіцієнт детермінації (R2) та інші метрики розраховуються для оцінки якості прогнозів та відображаються на веб-сторінці сервісу для користувача. Ці метрики допомагають визначити, наскільки точно модель здатна передбачати котирування на основі історичних даних (рис. 3.5).

Збереження моделі є наступним важливим етапом. Це дозволяє використовувати навчений алгоритм у майбутньому без необхідності повторного

навчання. Модель зберігається у файлі за допомогою бібліотеки **joblib**, що дозволяє швидко завантажувати модель для подальшого використання. Збережені моделі можуть бути легко інтегровані у веб-сервіс, забезпечуючи швидке та ефективно прогнозування на основі нових даних.

3.3 Програмування frontend-частини веб-сервісу

3.3.1 Розробка інтерфейсу користувача

Розробка інтерфейсу користувача (UI) для веб-сервісу прогнозування котирувань криптовалют є важливим аспектом, що забезпечує зручну та ефективну взаємодію користувачів з системою. Інтерфейс розроблено з використанням технологій React, HTML, CSS та JavaScript, що дозволяє створити інтерактивний і динамічний веб-додаток. Також була використана бібліотека Chart.js, що забезпечує візуалізацію даних у вигляді графіків.



Рисунок 3.6 – Інтерфейс веб-сервісу

Основні компоненти інтерфейсу включають панель керування, основний екран з графіками та розділ зі статистикою моделей, їх відповідно зображено на рисунку 3.6. Панель керування розташована зліва і забезпечує доступ до функцій вибору котирувань, встановлення параметрів моделі та створення нових моделей. Вона містить елементи управління, такі як випадаючі списки для вибору тикера криптовалюти, поля введення для встановлення дати початку даних, кількості епох, розміру батчу та частки тестових даних. Користувачі

можуть заповнити ці поля та натиснути кнопку "Create Model" для ініціації процесу створення моделі.

Основний екран розташований праворуч від панелі керування і включає графік, що відображає історичні дані котирувань та прогнозовані значення. Графік створюється за допомогою бібліотеки Chart.js, яка забезпечує високу якість візуалізації та інтерактивність. Користувачі можуть взаємодіяти з графіком, переглядати детальні дані по окремих точках та масштабувати відображення до останніх 100 значень за допомогою кнопки "Scale to Latest 100". Це дозволяє швидко аналізувати поточні тенденції та прогнозовані зміни котирувань.

У верхній частині основного екрана розташовано назву моделі, що відображає поточний стан моделі (наприклад, "ready" або "training"). Також передбачена кнопка "Refresh", яка дозволяє оновити статус моделі та графік, забезпечуючи актуальність відображуваних даних. Додатково, реалізовано таймер автоматичного оновлення, що дозволяє системі самостійно оновлювати дані через певний інтервал часу (5 секунд за замовчуванням).

Нижня частина основного екрана містить розділ зі статистикою моделей, де відображаються ключові метрики, такі як пояснена дисперсія, середня абсолютна помилка, середньоквадратична помилка, корінь середньоквадратичної помилки та коефіцієнт детермінації (R^2). Ці метрики надають користувачам інформацію про якість моделі та точність прогнозів, що дозволяє оцінити ефективність роботи системи.

Розробка інтерфейсу користувача передбачає використання CSS для створення сучасного та привабливого дизайну. Використано гнучкі контейнери для забезпечення адаптивності інтерфейсу на різних пристроях та екранах різних розмірів. Стили кнопок, форм та інших елементів інтерфейсу розроблено з урахуванням принципів зручності користування та естетичності. Було використано принципи дизайну сайтів Material design, розроблені в компанії Google [34].

JavaScript та React забезпечують динамічність інтерфейсу, дозволяючи реалізувати асинхронні запити до сервера для отримання даних, створення моделей та оновлення графіків. Це дозволяє користувачам отримувати миттєві відповіді на свої дії без необхідності перезавантаження сторінки. Крім того, використання React забезпечує високу продуктивність та масштабованість додатка, що є важливим для обробки великої кількості даних та забезпечення стабільної роботи системи.

3.3.2 Візуалізація прогнозів котирувань.

Візуалізація прогнозів котирувань є ключовим компонентом веб-сервісу, оскільки вона дозволяє користувачам швидко та ефективно аналізувати результати моделей. Веб-сервіс використовує бібліотеку Chart.js для створення інтерактивних графіків, що відображають історичні дані та прогнозовані значення котирувань криптовалют. Ця бібліотека забезпечує високу якість графічного відображення та підтримує різноманітні типи графіків, що дозволяє налаштовувати візуалізацію відповідно до потреб користувача.

Процес візуалізації розпочинається з отримання даних від сервера. Після обробки запиту на прогнозування сервер надсилає клієнту JSON-відповідь, що містить масиви даних з історичними та прогнозованими значеннями котирувань. Ці дані потім обробляються на клієнтській стороні та передаються до Chart.js для побудови графіків. Використання асинхронних запитів дозволяє миттєво оновлювати графіки без перезавантаження сторінки, що забезпечує безперервність роботи користувача з інтерфейсом.

На графіках, створених за допомогою Chart.js, історичні дані відображаються у вигляді лінійного графіка, що показує зміни котирувань у часі. Прогнозовані значення, які генерує модель, відображаються на тому ж графіку, що дозволяє користувачам порівнювати фактичні та прогнозовані дані. Для покращення читабельності графіків використовуються різні кольори для історичних та прогнозованих значень, а також додаткові елементи, такі як маркери точок прогнозів та легенда.

Візуалізація також включає можливість масштабування та навігації по графіку, що дозволяє користувачам детально аналізувати окремі періоди часу або відображати дані за весь доступний інтервал. Інтерактивні можливості Chart.js дозволяють користувачам наводити курсор на окремі точки графіка для отримання точних значень котирувань у певний момент часу.

Отже, в результаті виконання третього розділу роботи детально розглянуто процес створення архітектури веб-сервісу та впровадження його ключових компонентів. Описано основні етапи розробки, які включають аналіз прогнозних методів, проєктування системної архітектури, реалізацію бекенд та фронтенд частин веб-сервісу, а також методи валідації розробленої системи. Проведено порівняльний аналіз різних підходів до прогнозування та здійснено оцінку точності моделей. Результатом є реалізований веб-сервіс, що забезпечує інтегровану систему для прогнозування котирувань на ринку криптовалют, використовуючи сучасні технології веб-розробки та машинного навчання.

ВИСНОВОК

Таким чином, у результаті виконання кваліфікаційної роботи бакалавра розроблено програмну систему веб-сервісу для прогнозування котирувань на ринку криптовалют. Зокрема було досліджено ринок криптовалют, наявні рішення, методи прогнозування криптовалют, методи розробки веб-сервісів, розроблено архітектуру веб-сервісу, розроблено дизайн інтерфейсу користувача, реалізовано backend та frontend частин веб-сервісу.

У першому розділі проведено детальний аналіз ринку криптовалют, зокрема визначення основних понять, характеристик криптовалютного ринку, а також класифікація популярних криптовалют. Описані особливості ринку та фактори, що впливають на котирування криптовалют. Також проведено аналіз методів прогнозування котирувань, включаючи технічний та фундаментальний аналіз, а також машинне навчання. Проведено огляд наявних веб-сервісів для прогнозування криптовалют і їх порівняльний аналіз.

У другому розділі детально розглянуто вибір технологій для розробки веб-сервісу, таких як Flask для розробки API, різні фронтенд-технології (React, JavaScript, HTML, CSS, Chart.js) для створення зручного інтерфейсу користувача. Також описано використання бібліотек TensorFlow, Pandas та Sklearn для обробки даних та реалізації методів машинного навчання. Особлива увага приділяється моделі LSTM для прогнозування криптовалют, її перевагам та недолікам.

У третьому розділі описано процес проєктування архітектури веб-сервісу та реалізацію його основних компонентів. Докладно розглянуто етапи розробки, включаючи аналіз методів прогнозування, проєктування архітектури системи, реалізацію backend та frontend частин веб-сервісу, а також методи валідації розробленої системи. Описано проведення порівняльного аналізу різних підходів до прогнозування та оцінку точності моделей.

На основі проведеного дослідження створено функціональний веб-сервіс для прогнозування котирувань криптовалют з використанням моделі LSTM.

Реалізовано повний цикл розробки веб-сервісу від аналізу методів прогнозування до створення архітектури та реалізації backend і frontend частин. Проведено порівняльний аналіз різних підходів до прогнозування, а також оцінку точності моделей. Веб-сервіс надає можливість прогнозувати котирування криптовалют, використовуючи історичні дані та візуалізувати результати прогнозів. Розроблена система може бути використана як корисний інструмент для інвесторів та аналітиків, що займаються криптовалютами, а також як основа для подальших досліджень та вдосконалення методів прогнозування котирувань фінансових інструментів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Preeti, Bala R., Singh R. P. Financial and non-stationary time series forecasting using LSTM recurrent neural network for short and long horizon. 2019 10th international conference on computing, communication and networking technologies (ICCCNT), м. Kanpur, India, 6–8 лип. 2019 р. 2019. URL: <https://doi.org/10.1109/icccnt45670.2019.8944624>
2. Karpathy A., Johnson J., Li F. F. Visualizing and understanding recurrent networks. Iclr 2016. 2016. URL: <http://vision.stanford.edu/pdf/KarpathyICLR2016.pdf>
3. Bitcoin - Open source P2P money. URL: <https://bitcoin.org/bitcoin.pdf>
4. Khmelenko K. M., Tymchenko L. M. General characteristics of cryptocurrency. Juridical scientific and electronic journal. 2022. № 12. С. 344–347. URL: <https://doi.org/10.32782/2524-0374/2022-12/80>
5. Криптовалютний ринок для новачка. URL: <https://crew-c.com/dlya-novachka/kryptovalyutnyj-rynok/>
6. Ваша подорож починається прямо зараз | Binance Academy. Binance Academy. URL: <https://academy.binance.com/uk/start-here>
7. CryptoPredictions.com - (likely) the BEST Crypto Prediction Site. URL: <https://cryptopredictions.com/>
8. BTC health stock forecast: down to 0.0216 AUD? - BTC stock price prediction, long-term & short-term share revenue prognosis with smart technical analysis. Walletinvestor.com. URL: <https://walletinvestor.com/asx-stock-forecast/btc-stock-prediction>
9. Introduction. EFA Forecast. URL: <https://longforecast.com/>
10. Hamayel M. J., Owda A. Y. A novel cryptocurrency price prediction model using GRU, LSTM and bi-lstm machine learning algorithms. Ai. 2021. Т. 2, № 4. С. 477–496. URL: <https://doi.org/10.3390/ai2040030>

11. Bitcoin and cryptocurrency technologies arvind narayanan, joseph bonneau, edward felten, andrew miller and steven goldfeder. Network security. 2016. T. 2016, № 8. C. 4. URL: [https://doi.org/10.1016/s1353-4858\(16\)30074-5](https://doi.org/10.1016/s1353-4858(16)30074-5)
12. Design decisions in flask – flask documentation (3.1.x). Welcome to Flask – Flask Documentation (3.0.x). URL: <https://flask.palletsprojects.com/en/latest/design/>
13. Anurag J. Design and implementation of a web application to advise crypto investors. International Research Journal of Modernization in Engineering Technology and Science. 2022. №6. C. 2043–2036.
14. Sebastião H., Godinho P. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. Financial innovation. 2021. T. 7, № 1. URL: <https://doi.org/10.1186/s40854-020-00217-x>
15. Joseph, K., Biswas, A., & Kandpal, C. Bitcoin Price Prediction Using LSTM Neural Network. International Journal of Engineering and Advanced Technology (IJEAT). 2019. 8(6S), 238-242. URL: <https://www.ijeat.org/wpcontent/uploads/papers/v8i6S/F30430886S19.pdf>
16. A cryptocurrency price prediction model using deep learning / A. V. та ін. E3S web of conferences. 2023. T. 391. C. 01112. URL: <https://doi.org/10.1051/e3sconf/202339101112>
17. Encean A.-A., Zinca D. Cryptocurrency price prediction using LSTM and GRU networks. 2022 international symposium on electronics and telecommunications (ISETC), м. Timisoara, Romania, 10–11 листоп. 2022 р. 2022. URL: <https://doi.org/10.1109/isetc56213.2022.10010329>
18. Xu D. Price prediction of cryptocurrency based on LSTM model: evidence from ethereum. Highlights in science, engineering and technology. 2023. T. 39. C. 744–748. URL: <https://doi.org/10.54097/hset.v39i.6639>
19. Zheng. LSTM-Based cryptocurrency price prediction. Journal of financial data science. 2019.
20. Lee. Enhancing cryptocurrency forecasts with LSTM networks. International Journal of Electronic Commerce Studies. 2020.

21. Smith, Jones. A comparative study of financial forecasting models. *Cryptocurrency Journal of Predictive Analytics*. 2021.
22. Cryptocurrency price prediction using LSTM and recurrent neural networks / V. Kumar та ін. 2023 2nd international conference on applied artificial intelligence and computing (ICAAIC), м. Salem, India, 4–6 трав. 2023 р. 2023. URL: <https://doi.org/10.1109/icaaic56838.2023.10141048>
23. Hamayel M. J., Owda A. Y. A novel cryptocurrency price prediction model using GRU, LSTM and bi-lstm machine learning algorithms. *Ai*. 2021. Т. 2, № 4. С. 477–496. URL: <https://doi.org/10.3390/ai2040030>
24. Gunarto D. M., Sa'adah S., Utama D. Q. Predicting cryptocurrency price using RNN and LSTM method. *Jurnal sisfokom (sistem informasi dan komputer)*. 2023. Т. 12, № 1. С. 1–8. URL: <https://doi.org/10.32736/sisfokom.v12i1.1554>
25. Seabe P. L., Moutsinga C. R. B., Pindza E. Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: a deep learning approach. *Fractal and fractional*. 2023. Т. 7, № 2. С. 203. URL: <https://doi.org/10.3390/fractalfract7020203>
26. Development of recurrent neural networks for price forecasting at cryptocurrency exchanges / V. Tyshchenko та ін. *Eastern-European journal of enterprise technologies*. 2023. Т. 5, № 4 (125). С. 43–54. URL: <https://doi.org/10.15587/1729-4061.2023.287094>
27. Lahmiri S., Bekiros S. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, solitons & fractals*. 2019. Т. 118. С. 35–40. URL: <https://doi.org/10.1016/j.chaos.2018.11.014>
28. Lin Q. Price prediction of bitcoin using LSTM neural network. *BCP business & management*. 2023. Т. 38. С. 2738–2744. URL: <https://doi.org/10.54691/bcpbm.v38i.4183>
29. Jothidurai D. A. M., Kanchan P. D., Raj R. Cryptocurrency price prediction and forecasting market momentum using machine learning techniques. *International journal for research in applied science and engineering technology*. 2023. Т. 11, № 7. С. 2100–2115. URL: <https://doi.org/10.22214/ijraset.2023.55026>

30. Wen N. S., Ling L. S. Evaluation of cryptocurrency price prediction using LSTM and cnns models. JOIV : international journal on informatics visualization. 2023. T. 7, № 3-2. C. 2016. URL: <https://doi.org/10.30630/joiv.7.3-2.2344>
31. Material design. Material Design. URL: <https://m2.material.io/design/introduction>
32. Lyu H. Cryptocurrency price forecasting: a comparative study of machine learning model in short-term trading. 2022 asia conference on algorithms, computing and machine learning (CACML), м. Hangzhou, China, 25–27 без. 2022 p. 2022. URL: <https://doi.org/10.1109/cacml55074.2022.00054>
33. B Mahesh. Machine learning algorithms -A review. International journal of science and research (IJSR). 2019. URL: https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-_A_Review.
34. A. Marosyan. The relative strength index revisited. African journal of business management, T.5, C. 5855, 2011. URL: https://www.researchgate.net/publication/267716142_The_relative_strength_index_revisited
35. The pandas development team. Pandas-dev/pandas: pandas. Zenodo. URL: <https://doi.org/10.5281/zenodo.3509134>

ДОДАТОК А

```
import os
from flask import Flask, request, jsonify
from datetime import datetime, timedelta
import json
import threading
import pandas as pd
import yfinance as yf
import joblib
from base_predictor import StockPricePredictor
from flask_cors import CORS, cross_origin

app = Flask(__name__)
cors = CORS(app)
app.config["CORS_HEADERS"] = "Content-Type"

models = {}

# Custom validation function for date format
def validate_date(date_text):
    try:
        datetime.strptime(date_text, "%Y-%m-%d")
        return True
    except ValueError:
        return False
```

```

def train_model(model_name):
    model = models[model_name]["model"]
    model.train()
    models[model_name]["status"] = "ready"
    # joblib.dump(models, "models.pkl")

# Endpoint to create a model
@app.route("/models", methods=["POST"])
@cross_origin()
def create_model():
    data = request.json
    ticker = data["ticker"]
    interval = data["interval"]
    start_date = data["start_date"]
    epochs = data.get("epochs", 100)
    batch_size = data.get("batch_size", 32)
    test_part = data.get("test_part", 0.2)
    model_name = f'{ticker}_{interval}_{start_date}'
    models[model_name] = {
        "status": "training",
        "ticker": ticker,
        "interval": interval,
        "start_date": start_date,
    }
    models[model_name]["model"] = StockPricePredictor(
        ticker=ticker,
        interval=interval,
        start_date=start_date,
        epochs=epochs,

```

```

        batch_size=batch_size,
        test_part=test_part,
    )

    # Start training in a separate thread
    threading.Thread(target=train_model, args=(model_name,)).start()

    return jsonify({"model_status": "training", "model_name": model_name})

# Endpoint to get model status
@app.route("/models/<model_name>", methods=["GET"])
@cross_origin()
def get_model(model_name):
    if model_name in models:
        return jsonify(models[model_name])
    else:
        return jsonify({"error": "Model not found"}), 404

@app.route("/models/<model_name>/data", methods=["GET"])
@cross_origin()
def get_model_data(model_name):
    if model_name in models:
        maindf = models[model_name]["model"].data
        modified_maindf = pd.DataFrame(
            {
                "date": maindf.index,
                "close": maindf["close"],
            }

```

```

)

modified_maindf.reset_index(drop=True, inplace=True)
modified_maindf["date"] = modified_maindf["date"].apply(str)
response = {
    "columns": modified_maindf.columns.tolist(),
    "values": modified_maindf.values.tolist(),
}
return jsonify({"data": response})
else:
    return jsonify({"error": "Model not found"}), 404

```

Endpoint to make predictions

```
@app.route("/model/<model_name>/predict", methods=["POST"])
```

```
@cross_origin()
```

```
def predict(model_name):
```

```
    data = request.json
```

```
    n_predictions = data["n_predictions"]
```

```
    print(model_name)
```

```
    if model_name in models and models[model_name]["status"] == "ready":
```

```
        model = models[model_name]["model"]
```

```
        predictions = model.predict(n_predictions)
```

```
        date_range = pd.date_range(
```

```
            datetime.today().replace(hour=0, minute=0, second=0, microsecond=0)
```

```
            - timedelta(days=15),
```

```
            periods=len(predictions),
```

```
            freq="D",
```

```

    )
    predictions["date"] = date_range
    predictions["date"] = predictions["date"].apply(str)
    response = {
        "columns": predictions.columns.tolist(),
        "predictions": predictions.values.tolist(),
    }
    return jsonify({"predictions": response})
else:
    return jsonify({"error": "Model not ready or not found"}), 404

```

Endpoint to list all models

```
@app.route("/models", methods=["GET"])
```

```
@cross_origin()
```

```
def list_models():
```

```
    response = {
```

```
        model_name: {
```

```
            "status": model["status"],
```

```
            "model": model["model"].toJSON() if model["status"] == "ready" else
```

None,

```
        }
```

```
        for model_name, model in models.items()
```

```
    }
```

```
    return jsonify(response)
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

ДОДАТОК Б

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
import yfinance as yf
import ta
from ta.volatility import BollingerBands

import os
import pandas as pd
import numpy as np
import math
import datetime as dt

from sklearn.metrics import (
    mean_squared_error,
    mean_absolute_error,
    explained_variance_score,
    r2_score,
)
from sklearn.metrics import mean_poisson_deviance, mean_gamma_deviance,
accuracy_score
from sklearn.preprocessing import MinMaxScaler

import tensorflow as tf

import matplotlib.pyplot as plt
from itertools import cycle
import plotly.graph_objects as go
```

```
import plotly.express as px
from plotly.subplots import make_subplots

class StockPricePredictor:
    def __init__(
        self,
        ticker="BTC-USD",
        start_date="2017-01-01",
        interval="1d",
        epochs=100,
        batch_size=32,
        test_part=0.2,
    ):
        self.ticker = ticker
        self.start_date = start_date
        self.interval = interval
        self.epochs = epochs
        self.batch_size = batch_size
        self.test_part = test_part
        self.model = None
        self.time_step = 15
        self.train_stats = {}
        self.test_stats = {}
        self.plot = None
        self.scaler = MinMaxScaler(feature_range=(0, 1))
        self.data = None

    def toJSON(self):
        return {
            "ticker": self.ticker,
```

```

    "start_date": self.start_date,
    "interval": self.interval,
    "epochs": self.epochs,
    "batch_size": self.batch_size,
    "test_part": self.test_part,
    "time_step": self.time_step,
    "train_stats": self.train_stats,
    "test_stats": self.test_stats,
}

```

```
def fetch_data(self):
```

```

    data = yf.download(self.ticker, start=self.start_date, interval=self.interval)
    data = data.dropna()
    data = data[["Close"]]
    data.columns = ["close"] # Rename the column here for consistency

    # Calculate technical indicators
    # data["rsi"] = ta.momentum.RSIIndicator(data["close"], window=14).rsi()
    # data["macd"] = ta.trend.MACD(data["close"]).macd()
    # data["ema20"] = ta.trend.EMAIndicator(data["close"],
window=20).ema_indicator()
    # indicator_bb = BollingerBands(close=data["close"], window=20,
window_dev=2)

    # data["bb_bbm"] = indicator_bb.bollinger_mavg()
    # data["bb_bbh"] = indicator_bb.bollinger_hband()
    # data["bb_bbl"] = indicator_bb.bollinger_lband()
    data.dropna(inplace=True)
    self.data = data
    return data

```

```

def train(self):
    data = self.fetch_data()
    closedf = data[["close"]]

    closedf.reset_index(drop=False, inplace=True)
    closedf.rename(columns={"Date": "Date", "close": "Close"},
inplace=True)

    close_stock = closedf.copy()
    del closedf["Date"]
    closedf = self.scaler.fit_transform(np.array(closedf).reshape(-1, 1))
    print(closedf.shape)

    training_size = int(len(closedf) * (1 - self.test_part))
    test_size = len(closedf) - training_size
    self.test_size = test_size
    train_data, test_data = (
        closedf[0:training_size, :],
        closedf[training_size : len(closedf), :1],
    )
    self.closedf = closedf
    self.test_data = test_data

def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset) - time_step - 1):
        a = dataset[i : (i + time_step), 0]
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)

```

```
self.time_step = 15
X_train, y_train = create_dataset(train_data, self.time_step)
X_test, y_test = create_dataset(test_data, self.time_step)

print("X_train: ", X_train.shape)
print("y_train: ", y_train.shape)
print("X_test: ", X_test.shape)
print("y_test", y_test.shape)

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

print("X_train: ", X_train.shape)
print("X_test: ", X_test.shape)

self.model = Sequential()
self.model.add(LSTM(10, input_shape=(None, 1), activation="relu"))
self.model.add(Dense(1))
self.model.compile(loss="mean_squared_error", optimizer="adam")

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

```

train_predict = self.model.predict(X_train)
test_predict = self.model.predict(X_test)
train_predict.shape, test_predict.shape

```

```

train_predict = self.scaler.inverse_transform(train_predict)
test_predict = self.scaler.inverse_transform(test_predict)
original_ytrain = self.scaler.inverse_transform(y_train.reshape(-1, 1))
original_ytest = self.scaler.inverse_transform(y_test.reshape(-1, 1))

```

```
try:
```

```

    train_rmse = math.sqrt(mean_squared_error(original_ytrain,
train_predict))

```

```
except Exception:
```

```
    train_rmse = 0
```

```
try:
```

```
    train_mse = mean_squared_error(original_ytrain, train_predict)
```

```
except Exception:
```

```
    train_mse = 0
```

```
try:
```

```
    train_mae = mean_absolute_error(original_ytrain, train_predict)
```

```
except Exception:
```

```
    train_mae = 0
```

```
try:
```

```

    train_explained_variance = explained_variance_score(
        original_ytrain, train_predict
    )

```

```
except Exception:
```

```
train_explained_variance = 0
```

```
try:
```

```
    train_r2 = r2_score(original_ytrain, train_predict)
```

```
except Exception:
```

```
    train_r2 = 0
```

```
try:
```

```
    train_mgd = mean_gamma_deviance(original_ytrain, train_predict)
```

```
except Exception:
```

```
    train_mgd = 0
```

```
try:
```

```
    train_mpd = mean_poisson_deviance(original_ytrain, train_predict)
```

```
except Exception:
```

```
    train_mpd = 0
```

```
self.train_stats = {
```

```
    "train_rmse": train_rmse,
```

```
    "train_mse": train_mse,
```

```
    "train_mae": train_mae,
```

```
    "train_explained_variance": train_explained_variance,
```

```
    "train_r2": train_r2,
```

```
    "train_mgd": train_mgd,
```

```
    "train_mpd": train_mpd,
```

```
}
```

```
try:
```

```
    test_rmse = math.sqrt(mean_squared_error(original_ytest, test_predict))
```

```
except Exception:
```

```
test_rmse = 0
```

```
try:
```

```
    test_mse = mean_squared_error(original_ytest, test_predict)
```

```
except Exception:
```

```
    test_mse = 0
```

```
try:
```

```
    test_mae = mean_absolute_error(original_ytest, test_predict)
```

```
except Exception:
```

```
    test_mae = 0
```

```
try:
```

```
    test_explained_variance = explained_variance_score(
        original_ytest, test_predict
    )
```

```
except Exception:
```

```
    test_explained_variance = 0
```

```
try:
```

```
    test_r2 = r2_score(original_ytest, test_predict)
```

```
except Exception:
```

```
    test_r2 = 0
```

```
try:
```

```
    test_mgd = mean_gamma_deviance(original_ytest, test_predict)
```

```
except Exception:
```

```
    test_mgd = 0
```

```
try:
```

```

        test_mpd = mean_poisson_deviance(original_ytest, test_predict)
except Exception:
    test_mpd = 0

self.test_stats = {
    "test_rmse": test_rmse,
    "test_mse": test_mse,
    "test_mae": test_mae,
    "test_explained_variance": test_explained_variance,
    "test_r2": test_r2,
    "test_mgd": test_mgd,
    "test_mpd": test_mpd,
}

look_back = self.time_step
trainPredictPlot = np.empty_like(closedf)
trainPredictPlot[:, :] = np.nan
    trainPredictPlot[look_back : len(train_predict) + look_back, :] =
train_predict
print("Train predicted data: ", trainPredictPlot.shape)

# shift test predictions for plotting
testPredictPlot = np.empty_like(closedf)
testPredictPlot[:, :] = np.nan
testPredictPlot[
    len(train_predict) + (look_back * 2) + 1 : len(closedf) - 1, :
] = test_predict
print("Test predicted data: ", testPredictPlot.shape)

plotdf = pd.DataFrame(

```

```

    {
        "date": close_stock["Date"],
        "original_close": close_stock["Close"],
        "train_predicted_close": trainPredictPlot.reshape(1, -1)[0].tolist(),
        "test_predicted_close": testPredictPlot.reshape(1, -1)[0].tolist(),
    }
)
self.plot = plotdf

```

```

def predict(self, days_count):
    x_input = self.test_data[len(self.test_data) - self.time_step :].reshape(1, -1)
    temp_input = list(x_input)
    temp_input = temp_input[0].tolist()

    from numpy import array

    lst_output = []
    n_steps = self.time_step
    i = 0
    pred_days = days_count
    while i < pred_days:

        if len(temp_input) > self.time_step:

            x_input = np.array(temp_input[1:])
            x_input = x_input.reshape(1, -1)
            x_input = x_input.reshape((1, n_steps, 1))

            yhat = self.model.predict(x_input, verbose=0)
            temp_input.extend(yhat[0].tolist())

```

```

temp_input = temp_input[1:]

lst_output.extend(yhat.tolist())
i = i + 1

else:

    x_input = x_input.reshape((1, n_steps, 1))
    yhat = self.model.predict(x_input, verbose=0)
    temp_input.extend(yhat[0].tolist())

    lst_output.extend(yhat.tolist())
    i = i + 1
print("Output of predicted next days: ", len(lst_output))
last_days = np.arange(1, self.time_step + 1)

temp_mat = np.empty((len(last_days) + pred_days + 1, 1))
temp_mat[:] = np.nan
temp_mat = temp_mat.reshape(1, -1).tolist()[0]

last_original_days_value = temp_mat
next_predicted_days_value = temp_mat

last_original_days_value[0 : self.time_step + 1] = (
    self.scaler.inverse_transform(
        self.closedf[len(self.closedf) - self.time_step : ]
    )
    .reshape(1, -1)
    .tolist()[0]
)

```

```
next_predicted_days_value[self.time_step + 1 :] = (  
    self.scaler.inverse_transform(np.array(lst_output).reshape(-1, 1))  
    .reshape(1, -1)  
    .tolist()[0]  
)  
new_pred_plot = pd.DataFrame(  
    {  
        "last_original_days_value": last_original_days_value,  
        "next_predicted_days_value": next_predicted_days_value,  
    }  
)  
new_pred_plot.dropna(inplace=True)  
return new_pred_plot
```