

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ

Нейромережний застосунок прогнозування курсу
криптовалют на прикладі Bitcoin

Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Аналітика даних»
Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи Анд- 41

Тищенко В.О.
(прізвище та ініціали)

Керівник Іларіонов О.Є.
(прізвище та ініціали)

К.Т.Н., ДОЦЕНТ
(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту рішенням
кафедри *інтелектуальних технологій*

Протокол №13 від 05.06.2023 р. зав. кафедри _____ доц. Іларіонов О.Є.

Київ – 2023

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.

“ ___ ” _____ 2023 р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**
Тищенку Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) «Нейромережний застосунок прогнозування курсу криптовалют на прикладі Bitcoin», затверджена протоколом засідання кафедри від «11» листопада 2022 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) _____ травня 2023 року
3. Вихідні дані до проекту (роботи) Локальний тренд Біткоїна
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити): Дослідження та аналіз предметної області, що пов'язана із методами прогнозування курсу криптовалют, програмна реалізація застосунку, тестування розробленого застосунку.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
Мета дослідження дипломного проекту (1 слайд), аналіз предметної області (1 слайд), методи прогнозування (2 слайди), архітектура застосунку (1 слайд), приклад роботи застосунку (1 слайд), аналіз результатів (1 слайд), висновки (1 слайд).

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 15 лютого 2023 року

Керівник _____ / Іларіонов О.Є. /
 (підпис) (ПІБ)

Завдання прийняв до виконання _____ / Тищенко В.О. /
 (підпис) (ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
	Обговорення з керівником постановки завдання, підбір літератури, огляд існуючих рішень	15.02.2023 – 01.03.2023	
	Аналіз постановки задачі, формалізація задачі, аналіз літературних джерел, написання розділу 1	02.03.2023 – 20.03.2023	
	Розробка нейромережного застосунку прогнозування курсу криптовалют, написання розділу 2	21.03.2023 – 11.04.2023	
	Програмна реалізація нейромережного застосунку прогнозування криптовалют, написання розділу 3	12.04.2023 – 15.05.2023	
	Робота над оформленням пояснювальної записки та презентаційних матеріалів	16.05.2023 – 29.05.2023	

Студент-дипломник _____ / Тищенко В.О. /
 (підпис) (ПІБ)

Керівник випускної кваліфікаційної роботи _____ / Іларіонов О.Є. /
 (підпис) (ПІБ)

АНОТАЦІЯ

Тищенко Владислав Олександрович виконав випускню кваліфікаційну роботу освітнього рівня бакалавр на тему «Нейромережний застосунок прогнозування курсу криптовалют на прикладі Bitcoin» зі спеціальності 122 – «Комп'ютерні науки», за освітньою програмою «Аналітика даних».

У випускній кваліфікаційній роботі досліджено питання прогнозування курсу криптовалют, створено нейромережний застосунок із використанням моделі LSTM.

Ключові слова: Блокчейн, криптовалюта, Біткоїн, рекурентна нейронна мережа, LSTM.

ABSTRACT

Tyshchenko Vladyslav Oleksandrovych completed the final qualification work of the bachelor's degree on the topic "Neural network application for predicting the rate of cryptocurrency on the example of Bitcoin" in the specialty 122 - "Computer Science", educational program "Data Science".

In the final qualification work, the issue of predicting the rate of cryptocurrencies was investigated, a neural network application was created using the LSTM model.

Keywords: Blockchain, cryptocurrency, Bitcoin, recurrent neural network, LSTM.

ЗМІСТ

Вступ	7
Розділ 1. Аналітичний огляд методів прогнозування курсу криптовалют	9
1.1. Основні поняття блокчейну та криптовалюти	9
1.2. Дослідницький аналіз даних	12
1.3. Огляд методів прогнозування часових рядів	14
1.4. Рекурентні нейронні мережі.....	17
1.5. Постановка задачі на розробку застосунку	21
1.6. Опис вимог	22
Висновки за розділом 1	22
Розділ 2. Розробка нейромережного застосунку прогнозування курсу криптовалют	23
2.1. Архітектура нейромережного застосунку	23
2.2. Побудова моделі	28
2.3. Оцінка моделі.....	31
2.4. Аналіз результатів моделі	32
Висновки за розділом 2	35
Розділ 3. Програмна реалізація нейромережного застосунку прогнозування криптовалют	36
3.1. Вибір середовища та інструментів для розробки застосунку	36
3.2. Фізична структура застосунку	38
3.3. Програмна реалізація інтерфейсу користувача.....	39
3.4. Тестування та аналіз результату роботи нейромережного застосунку прогнозування курсу криптовалют	43

3.5. Інструктивний матеріал користувача для роботи з нейромережним застосунком прогнозування курсу криптовалют.....	49
Висновки за розділом 3	50
Висновки.....	51
Список використаних джерел.....	52
Додатки.....	54
Додаток А.....	54
Додаток Б	55
Додаток В.....	56
Додаток Г	57
Додаток Д.....	58
Додаток Е.....	59
Додаток Є.....	60
Додаток Ж.....	61
Додаток З	62

ВСТУП

На даний момент прогнозування курсу інструментів фінансового ринку відбувається безліччю різних методів. До цих методів можна віднести як фундаментальний аналіз, який дозволяє будувати судження про подальші зміни на ринку за допомогою вивчення поля новин, так і технічний, в якому основним інструментарієм служать статистичні та математичні засоби. Багато професійних трейдерів для передбачення курсу використовують свої навички в пошуку максимально схожих патернів (моделі повторюваних ситуацій на ринку), користуючись при цьому різними індикаторами. До того ж, хорошим індикатором може бути і курс іншого інструменту над ринком. Таке зустрічається за високого рівня залежності (кореляції) курсів. Крім того, деякі учасники ринку використовують сучасні технології, які допомагають при ухваленні інвестиційних рішень. До таких технологій можна віднести штучні нейронні мережі, які при правильному проектуванні та навчанні здатні стати чудовим інструментом для прогнозування.

Актуальність досліджуваної теми полягає, в першу чергу, в інтересі великих груп людей до криптовалютних ринків. Для прийняття успішних торгових рішень у цій сфері потрібно використовувати різні методи та засоби аналізу часових рядів, а також їх прогнозування.

Ключовим інструментом, що забезпечує успішну діяльність на криптовалютному ринку, вважається використання різних математичних методів, таких як, наприклад, статистичні методи або засоби аналізу та прогнозування.

Метою дипломної роботи є розробка та реалізація нейромережного алгоритму прогнозування курсу Bitcoin з метою покращення точності прогнозів та надання цінних інструментів для трейдерів та інвесторів.

Предмет дослідження: Нейромережні моделі та їх застосування в прогнозуванні фінансових ринків, зокрема курсу криптовалюти Bitcoin.

Об'єкт дослідження: Процес прогнозування курсу криптовалюти Bitcoin з використанням нейромережних моделей та аналізу вхідних даних, таких як цінові показники, обсяги торгів та інші фактори, що можуть впливати на ціну криптовалюти.

Для досягнення поставленої мети дипломної роботи необхідно виконати наступні завдання:

- Зібрати вхідні дані про курс Bitcoin та відповідні фактори, що впливають на його зміну.
- Вивчити теоретичні аспекти нейромереж та їх застосування в прогнозуванні фінансових ринків.
- Розробити та натренувати нейромережну модель для прогнозування курсу Bitcoin на основі зібраних даних.
- Провести експериментальне порівняння розробленої моделі з існуючими методами прогнозування криптовалютного ринку.
- Оцінити точність та ефективність розробленої моделі за допомогою відповідних метрик та порівняти її з результатами інших методів прогнозування.
- Розробити інтерфейс застосунку для надання прогнозів курсу Bitcoin та відображення додаткових корисних аналітичних даних для користувачів.

В результаті дипломної роботи очікується отримання нейромережного застосунку на основі моделі LSTM, який здатний забезпечувати прогнози курсу криптовалюти Bitcoin на , що можуть бути корисними для інвесторів, трейдерів та інших зацікавлених осіб.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

1.1. Основні поняття блокчейну та криптовалюти

Блокчейн — це технологія децентралізованого зберігання та розподіленого внесення записів про транзакції, що базується на криптографічних методах захисту інформації, що дозволяє виключити посередника [1]. З англійської блокчейн перекладається, як «ланцюжок блоків», тобто безперервна послідовність блоків ланцюжку, побудована за певними правилами і яка містить інформацію.

Зазвичай копії всіх ланцюжків блоків незалежно один від одного зберігаються на величезній кількості різних комп'ютерів. Вперше цей термін з'явився як назва повністю розподіленої реплікованої бази даних, реалізованої в системі криптовалюти Біткоїн, внаслідок чого технологію блокчейн зазвичай відносять до транзакцій різних цифрових монет, проте вона може бути поширена серед будь-яких взаємопов'язаних інформаційних блоків.

Дані хронологічно послідовні, оскільки не можна видаляти або змінювати ланцюжок без консенсуса з боку мережі. В результаті можна використовувати технологію блокчейн для створення незмінного або безстрокового реєстру для відстеження замовлень, платежів, рахунків та інших транзакцій [1]. Система має вбудовані механізми, які запобігають несанкціонованому введенню транзакцій та створюють узгодженість у загальному поданні цих транзакцій (рисунок 1.1).

Головні особливості блокчейн технології:

- Прозорість. В блокчейні зберігаються дані про всі проведені операції за всю історію створення системи.
- Стабільність. Неможливо видалити або замінити інформацію, а тільки здійснити нову угоду.
- Незалежність. Копії всіх ланцюжків блоків незалежно один від одного зберігаються на величезній кількості різних комп'ютерів.

Криптовалюта — це різновид цифрових валют, створення яких ґрунтується на методах криптографії. Для запису транзакцій та випуску нових одиниць використовується децентралізована система.

Це система з рівноправними учасниками, що дозволяє будь-якому користувачеві, що перебуває в будь-якому місці, надсилати та отримувати платежі. Криптовалютні платежі існують виключно в цифровому вигляді в онлайн базі даних (блокчейні), що описує конкретні транзакції. При переказі коштів у криптовалюті, транзакції записуються до реєстру.

Криптовалютні ринки децентралізовані, вони не підтримуються або видаються центральним органом влади, оскільки вони працюють через мережу комп'ютерів [2]. Криптовалюта зберігається в електронних гаманцях.

Технологія блокчейну є основою сучасних криптовалют. В його основі лежить постійно оновлюючу публічну 'книгу' записів, для того щоб реєструвати всі транзакції. Технологія блокчейну дозволяє покупцю та продавцю взаємодіяти між собою напряму.

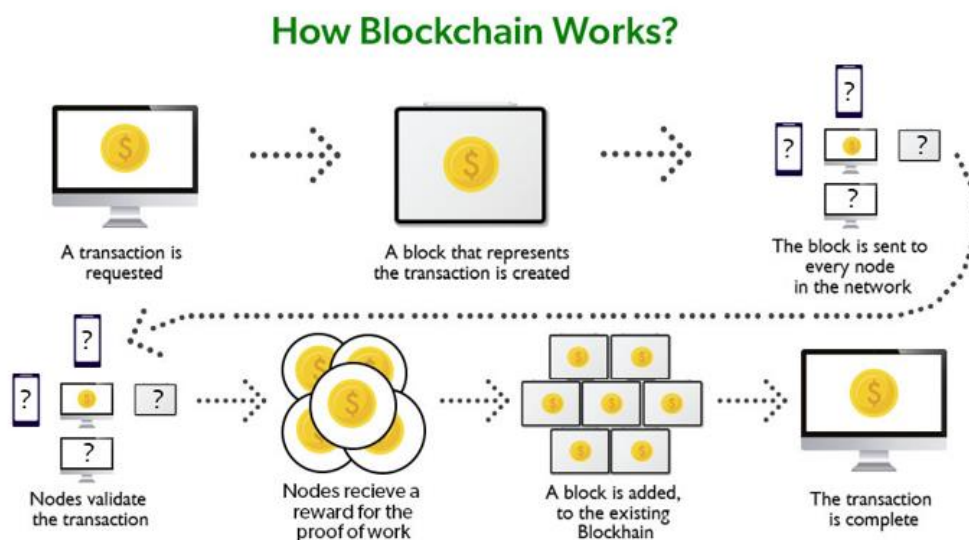


Рисунок 1.1 – Схема роботи блокчейну

Біткоїн — найперша криптовалюта та перша технологія, яка, по суті, запровадила поняття приватної власності у цифровому просторі. Проект було запущено у 2009 р.. Вперше принцип роботи Біткоїна його загадковий творець (або навіть загадкові творці) під псевдонімом Сатоші Накамото описав ще 2008

року у програмній статті про «однорангову електронну платіжну систему». У статті Сатоші, справжня особа якого невідома досі, технічною мовою описав те, як працює Біткоїн. Документ став апогеєм розвитку ідеї системи конфіденційних цифрових платежів, яка не залежала б від держав та інших інститутів.

Мережа Біткоїн автоматично випускає щойно викарбувані Біткоїни для майнерів, коли вони знаходять і додають нові блоки в блокчейн. Загальна пропозиція Біткоїнів має обмеження в 21 мільйон монет, тобто коли кількість монет в обігу досягне 21 мільйонів, протокол припинить карбувати нові монети. У певному сенсі майнінг Біткоїнів одночасно виконує функцію перевірки транзакцій і процесу випуску Біткоїнів.

Мережа Біткоїн використовує стратегію розподілу монет, відому як «халвінг Біткоїнів», яка забезпечує зменшення кількості Біткоїнів, що розподіляються майнерам з часом. Поступово зменшуючи пропозицію нових Біткоїнів, що надходять в обіг, ідея полягає в тому, що це допоможе підтримати ціну активу.

Біткоїн — реальна альтернатива фіатним грошам, яка поступово трансформує світову фінансову систему [2]. Головні переваги криптовалюти Біткоїну:

- Біткоїн не має дозволу. Для використання традиційних валют і форм грошей потрібен дозвіл (від банків, фінансових установ, урядів). Біткоїн не потребує ні від кого дозволу, він безкоштовний і відкритий для використання в усьому світі.
- Децентралізація. Мережа розподілена по всьому світу серед багатьох тисяч вузлів (комп'ютерів) і мільйонів користувачів, де вам не потрібно покладатися на надійних третіх сторін.
- Захищеність. Ніхто не може конфіскувати ваш Біткоїн, оскільки він вам належить, він не розміщений у жодному центральному банку чи компанії.

- Стійкість. За допомогою обчислювального алгоритму під назвою proof of-work ніхто не зможе заблокувати або цензурувати ваші транзакції.
- 24/7/365. Мережа працює цілодобово та без вихідних.
- Без лімітів. Власники Біткоїну можуть обмінювати будь-які суми криптовалют і без обмежень.
- Низькі комісії та швидкі транзакції

1.2. Дослідницький аналіз даних

В основу датасету був взятий набір даних про історичні дані Біткоїна на криптовалютній біржі Binance. Було обрано дані з проміжком у часі в день, оскільки цікавить довгострокове прогнозування. Вся зібрана інформація поділяється на такі підтипи:

- Date – Дата
- Open – Ціна відкриття
- Close – Ціна закриття
- High – Максимальне значення ціни протягом дня
- Low – Мінімальне значення ціни протягом дня
- Volume – Об'єм торгів протягом дня
- Change % – Зміна ціни закриття з попереднім днем

Усі значення вказані у доларах США.

Перед початком роботи з набором, слід перевірити його на наявність нульових значень. Також слід відформатувати стовпець дати у формат дати для зручності використання. Тепер можна перейти до дослідницького аналізу.

Як і в будь-якому іншому проекті, пов'язаному з наукою про дані або аналітикою, EDA (Exploratory Data Analysis) часто виконується для вивчення набору даних і для того, щоб аналітик краще його зрозумів.

Подивимось на дати початку та дати завершення набору даних (рисунок 1.2).

Starting Date 2018-01-01 00:00:00
Ending Date 2022-10-31 00:00:00

Рисунок 1.2 – Дати початку та дати завершення набору даних

Переглянемо кількість рядків (днів) у датасеті (рисунок 1.3).

```
Total number of days present in the dataset: 1765  
Total number of fields present in the dataset: 7
```

Рисунок 1.3 – Кількість рядків та стовпців набору даних

За допомогою графічного аналізу легко встановити статистичні закономірності і на їх основі прогнозувати в майбутньому. Тому спершу переглянемо ціну закриття Біткоїну за деякий час (рисунок 1.4).



Рисунок 1.4 – Ціна Біткоїну за деякий час

На графіку можемо побачити що ціна закриття Біткоїну здобула свого максимуму в листопаді 2021 року, після чого пішла на спад. Тепер переглянемо волатильність Біткоїну, враховуючи майже усі стовпці нашого набору даних (рисунок 1.5).



Рисунок 1.5 – Волатильність Біткоїну за деякий час

Подивимося середньорічну ціну закриття та об'єми торгів (рисунок 1.6).

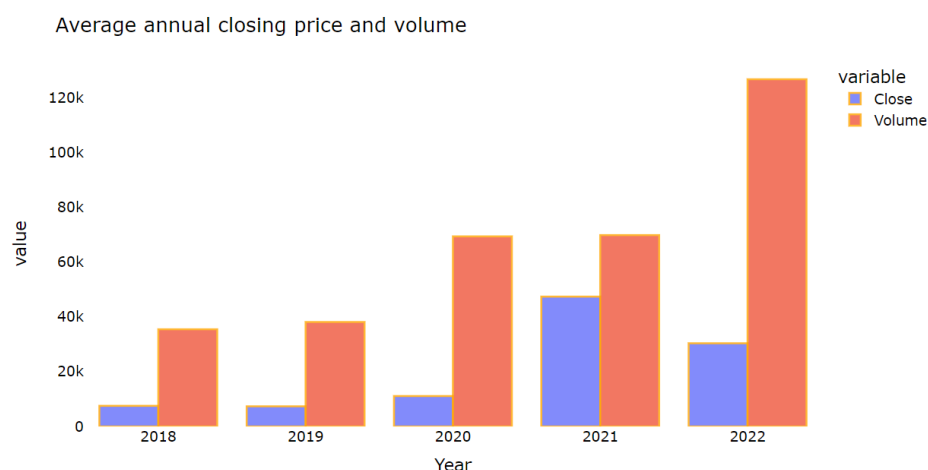


Рисунок 1.6 – Середньорічна ціна закриття та об'єм торгів

Переглянувши гістограму бачимо, що ціна Біткоїна та обсяги торгів з кожним роком значно збільшуються, що означає, що попит на криптовалюту зростає. Також бачимо що в 2022 році середня ціна бікоїну впала.

1.3. Огляд методів прогнозування часових рядів

Оскільки розглядається сукупність значень, взятих через рівні проміжки часу, то це є часовим рядом. Тобто буде виконуватись прогнозування часових рядів.

Модель часового ряду – математична модель прогнозування, побудова якої ґрунтується на виявленні залежності майбутнього значення від минулого всередині самого процесу та на підставі цієї залежності проведення прогнозування значень показників досліджуваного часового ряду [4]. Ці моделі мають універсальний вигляд для різних предметних областей.

Зважаючи на широку кількість предметних областей, провести точну класифікацію моделей предметної області неможливо. Однак, можна класифікувати моделі часових рядів за типом залежності майбутнього значення від минулого на статистичні та структурні моделі прогнозування.

1.3.1. Статистичні моделі прогнозування

У статистичних моделях прогнозування функціональна залежність між майбутніми і фактичними значеннями часового ряду, і навіть зовнішніми чинниками, якщо такі враховуються, задана аналітично, тобто формулою [4]. До статистичних моделей прогнозування належать такі групи:

1. Регресійні моделі. Регресійна модель – це метод вивчення статистичного взаємозв'язку між однією залежною кількісною залежною змінною від однієї або кількох незалежних кількісних змінних.
2. Авторегресійні моделі. Авторегресійна модель – це модель часового ряду, в якій його поточне значення лінійно залежить від попередніх значень цього ряду. Лінійна залежність означає, що поточне значення дорівнює виваженій сумі кількох попередніх значення низки.
3. Моделі експоненційного згладжування. В основу експоненційного згладжування закладено ідею постійного перегляду прогнозних значень у міру надходження фактичних. Модель привласнює експоненційно спадні ваги спостереженням у міру їхнього старіння.

Таким чином, останні доступні спостереження мають більший вплив на прогнозне значення, ніж старші спостереження.

Основними недоліками статистичних методів є середньозважений показник, що не враховує сезонні та інші нециклічні (випадкові) коливання обсягів продажів, тенденції можуть не зберегтися у майбутньому, коливання можуть мати не циклічний, а випадковий характер.

Цей факт визначає розробку методів, заснованих на суміщенні методів штучного інтелекту та інших методів аналізу часових рядів задля досягнення найбільшої точності прогнозу.

1.3.2. Структурні моделі прогнозування

У структурних моделях прогнозування функціональна залежність між майбутніми та фактичними значеннями часового ряду, а також зовнішніми факторами задана структурно, наприклад у вигляді графа [4]. До структурних моделей прогнозування відносяться такі групи:

1. Модель на нейронних мережах. За допомогою нейронних мереж можливе моделювання нелінійної залежності майбутнього значення часового ряду від його фактичних значень та від значень зовнішніх факторів. Нелінійна залежність визначається структурою мережі та функцією активації.
2. Модель на ланцюгах Маркова. Моделі прогнозування з урахуванням ланцюгів Маркова припускають, що майбутній стан процесу залежить від його поточного стану і не залежить від попередніх. У зв'язку з цим процеси, що моделюються ланцюгами Маркова, повинні належати до процесів з короткою пам'яттю.
3. Модель на класифікаційно-регресійних деревах. Структурні моделі розроблені для моделювання процесів, які впливають як безперервні зовнішні чинники, і категоріальні. Якщо зовнішні чинники, що впливають на процес $Z(t)$, безперервні, використовуються регресійні

дерева, а якщо фактори категоріальні, то класифікаційні дерева. Якщо необхідно враховувати фактори обох типів, то використовуються змішані класифікаційно-регресійні дерева.

1.4. Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) — це тип нейронних мереж, що спеціалізуються на обробці послідовностей. Це алгоритми глибокого навчання які використовують для моделювання послідовних даних, таких як часові ряди або природна мова. Вони включені в такі популярні програми, як Siri, голосовий пошук і Google Translate.

Подібно до нейронних мереж прямого зв'язку та згорткових нейронних мереж, рекурентні нейронні мережі використовують навчальні дані для навчання. Вони відрізняються своєю «пам'яттю», оскільки вони беруть інформацію з попередніх входних даних, щоб впливати на поточні входні та вихідні дані. Хоча традиційні глибокі нейронні мережі припускають, що входи та виходи незалежні один від одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності (рисунок 1.7).

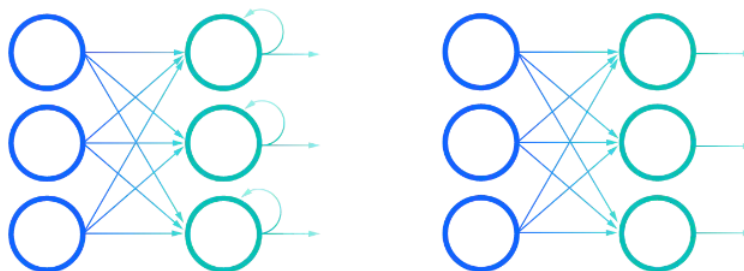


Рисунок 1.7 – Порівняння рекурентних нейронних мереж (ліворуч) і нейронних мереж прямого зв'язку (праворуч)

Якщо схематично, шар RNN використовує цикл for для ітерації за впорядкованою за часом послідовністю, зберігаючи при цьому у внутрішньому стані закодовану інформацію про кроки, які він уже бачив.

Дивлячись на зображення нижче (рисунок 1.8), «згорнуте» зображення RNN представляє всю нейронну мережу. «Розгорнутий» RNN представляє

окремі шари або часові кроки нейронної мережі. Попередні вхідні дані будуть представлені як прихований стан на третьому часовому кроці для прогнозування виходу в послідовності.

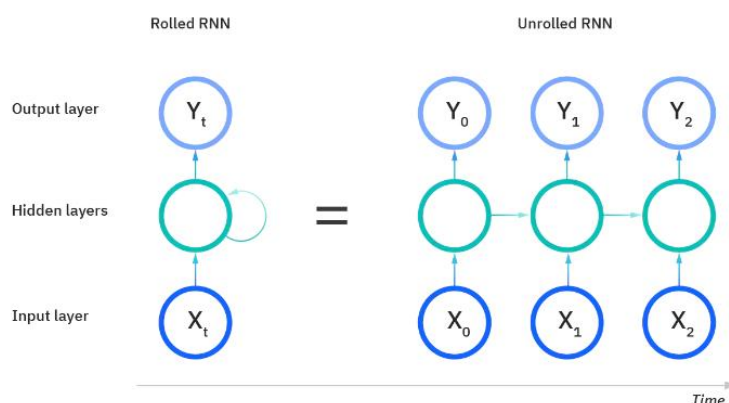


Рисунок 1.8 – Схематичне представлення рекурентних нейронних мереж

Іншою відмінною характеристикою рекурентних мереж є те, що вони спільно використовують параметри на кожному рівні мережі. У той час як мережі прямого зв'язку мають різні ваги для кожного вузла, рекурентні нейронні мережі мають однаковий параметр ваги на кожному рівні мережі. Тим не менш, ці ваги все ще коригуються за допомогою процесів зворотного поширення та градієнтного спуску, щоб полегшити навчання з підкріпленням.

Мережі прямого зв'язку відображають один вхід на один вихід, і хоча вище на наведених діаграмах візуалізовано рекурентні нейронні мережі таким чином, насправді вони не мають цього обмеження.

Натомість їхні вхідні та вихідні дані можуть відрізнятися за довжиною, а різні типи RNN використовуються для різних випадків використання.

Типи RNN зазвичай виражаються за допомогою таких діаграм:

1. Один до одного (рисунок 1.9)



Рисунок 1.9 – RNN один до одного

2. Один до багатьох (рисунок 1.10)

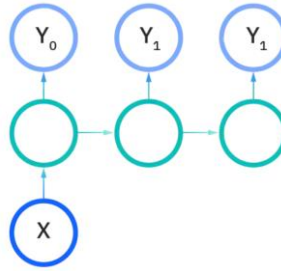


Рисунок 1.10 – RNN один до багатьох

3. Багато до одного (рисунок 1.11)

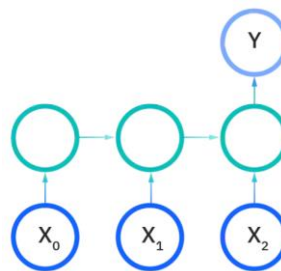


Рисунок 1.11 – RNN багато до одного

4. Багато до багатьох

а. Синхронізований (рисунок 1.12)

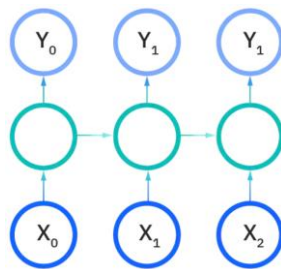


Рисунок 1.12 – RNN багато до багатьох (а)

б. Послідовний (рисунок 1.13)

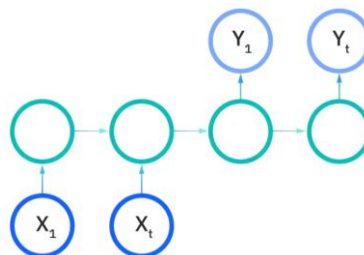


Рисунок 1.13 – RNN багато до багатьох (б)

Три найвідоміші види рекурентних нейронних мереж — це проста RNN, мережа з довгою короткочасною пам'яттю (LSTM) та мережа з рекурентними блоками (GRU).

Перший вид Simple RNN — це «проста» рекурентна нейронна мережа. Простий означає, що вузли мережі не змінюються. RNN має пам'ять для отримання будь-якої інформації з попередніх входних даних для впливу на поточний вхід і вихід [6].

Давайте подивимося на архітектуру блоку RNN (рисунок 1.14). Він приймає дані з попереднього кроку та поточні дані. Тут \tanh — це функція активації, замість \tanh ви також можете використовувати іншу функцію активації.

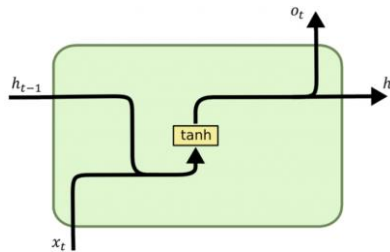


Рисунок 1.14 – Архітектура RNN

LSTM — це особливий вид RNN, здатний працювати над вирішенням проблеми довгострокових залежностей [6]. LSTM мають «клітини» в прихованих шарах нейронної мережі, які мають три ворота: вхідний, вихідний і забутий. Ці ворота контролюють потік інформації, який необхідний для прогнозування виходу в мережі [7].

У звичайній комірці RNN вхідні дані про позначку часу та прихований стан із попереднього кроку часу передаються через рівень активації для отримання нового стану. У той час як у LSTM процес дещо складний, як ви можете бачити в наведеній нижче архітектурі (рисунок 1.15), кожного разу він приймає вхідні дані з трьох різних станів, таких як поточний стан введення, короткочасна пам'ять із попередньої комірки та, нарешті, довгострокова пам'ять.

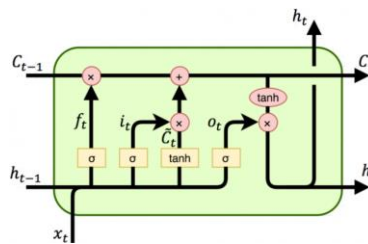


Рисунок 1.15 – Архітектура LSTM

GRU схожий на LSTM, оскільки він також працює для вирішення проблеми короткочасної пам'яті моделей RNN. Замість використання інформації для регулювання «стану комірки» він використовує приховані стани, і замість трьох воріт він має два — ворота скидання та ворота оновлення [6]. Подібно до воріт у LSTM, ворота скидання та оновлення контролюють, скільки та яку інформацію зберігати (рисунок 1.16).

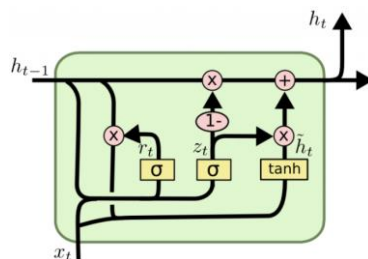


Рисунок 1.16 – Архітектура GRU

1.5. Постановка задачі на розробку застосунку

Після аналізу існуючих моделей прогнозування було обрано модель на нейронних мережах, оскільки можливе моделювання нелінійної залежності майбутнього значення часового ряду від його фактичних значень та від значень зовнішніх факторів. А саме, LSTM, оскільки вона здатна вивчати залежність порядку в задачах передбачення послідовності.

Перед тим як розпочати використовувати LSTM, слід поставити правильно завдання. Буде прогнозуватися курс Біткоїна, але передбачити точну ціну такої монети доволі складно, оскільки вона є доволі волатильною і на її та більшість монет впливають багато різних факторів. Особливо ринок криптовалют схильний до глобальних подій, здатних докорінно змінити його, як приклад

зростаюче прийняття Біткоїну та інших монет як платіжні кошти, інвестиційні активи, використання блокчейн-технологій та додатків у нових галузях бізнесу. Тому нашою задачею буде прогнозування локального тренду.

1.6. Опис вимог

Функціональні вимоги:

- Застосунок приймає дані для аналізу лише у форматі csv.
- Застосунок демонструє розглянуті дані для прогнозування у вигляді графіка.
- Застосунок видає результативний локальний тренд.
- Застосунок демонструє результативний тренд у вигляді графіка.

Нефункціональні вимоги:

- Зрозумілий і зручний у використанні зовнішній інтерфейс.
- Застосунок має оброблювати дані та видавати результати впродовж 3 хвилин.
- Присутня перевірка файлу даних на правильність формату.

Висновки за розділом 1

Отже, перший розділ дипломної роботи включає в себе описання предметної області, ознайомлення з такими поняттями як блокчейн, криптовалюта, Біткоїн і інше.

Було проведено дослідницький аналіз, завдяки котрому було ознайомлено з деталями датасету та підготовлено його до роботи.

Було проведено огляд існуючих моделей та методів прогнозування часових рядів, де добре видно їх недоліки і переваги. Також були розглянуті різновиди нейронних мереж, їх різниця, типи та види.

Зрештою, було поставлено завдання та певні вимоги для розробки нейромережного застосунку прогнозування курсу криптовалют.

РОЗДІЛ 2. РОЗРОБКА НЕЙРОМЕРЕЖНОГО ЗАСТОСУНКУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

2.1. Архітектура нейромережного застосунку

Застосунок буде приймати на вхід дані від користувача, після чого користувач на виході отримає результати прогнозування. На застосунок можуть впливати такі механізми як користувач та сама нейромережа. В якості обмежень, які мають місце, можна вказати правила форматування даних. Нейромережний застосунок можна описати у вигляді діаграми IDEF0 (рисунок 2.1).

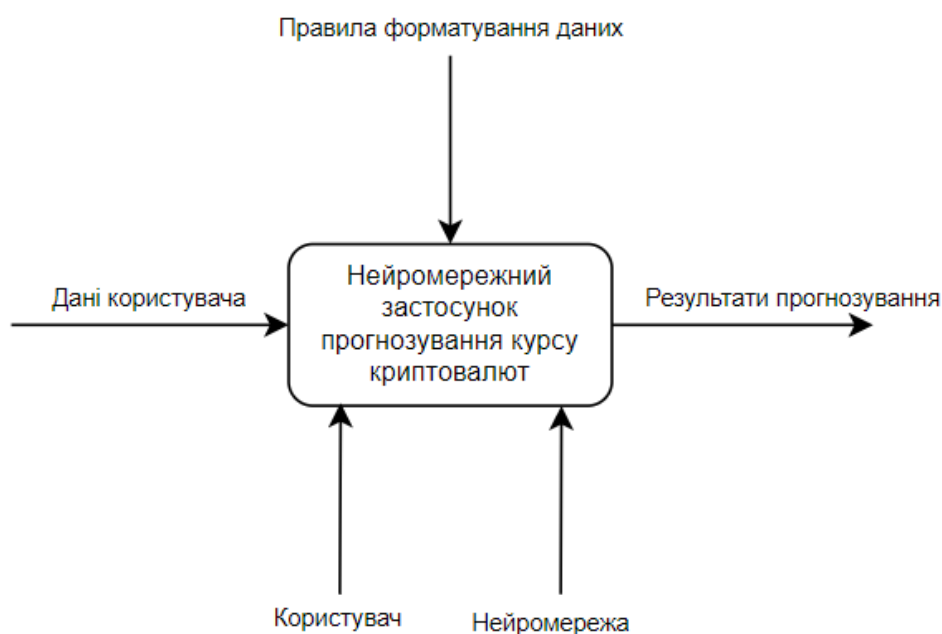


Рисунок 2.1 – Діаграма IDEF0 нейромережного застосунку

Застосунок буде включати в себе наступні основні функції:

1. Отримання даних
2. Побудова моделі
3. Формування результатів

Функція отримання даних буде включати у себе:

- Завантаження файлу користувачем
- Перевірка файлу на правильність формату
- Нормалізація даних

Функція побудови моделі буде включати у себе:

- Обробка даних
- Навчання нейромережі
- Тестування нейромережі

Функція формування результатів буде включати у себе:

- Прогнозування
- Формування тренду

На основі описаного функціонального аналізу можемо побудувати дерево функцій (рисунок 2.2).

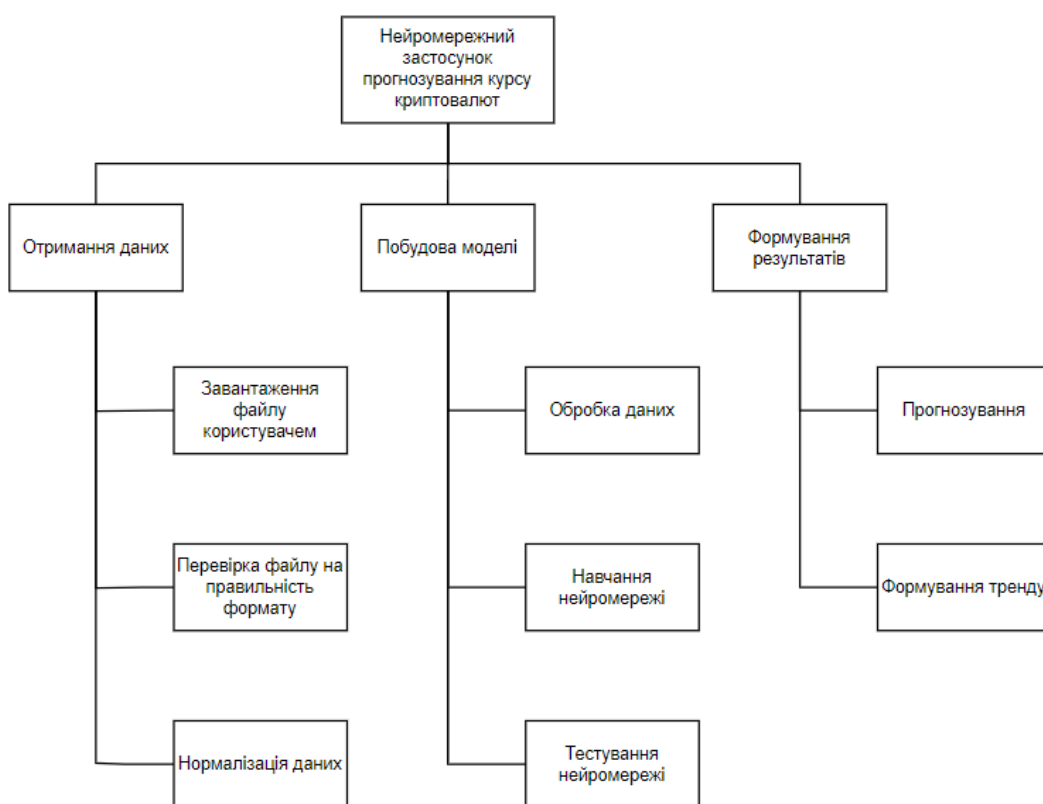


Рисунок 2.2 – Дерево функцій нейромережного застосунку

З складеного дерева функцій можна описати карту процесів.

Для отримання даних нейромережним застосунком, користувачу необхідно завантажити файл заданого формату, після чого цей файл перевіряється на правильність формату і вже потім дані файлу нормалізуються.

Після цього процесу починається побудова моделі, яка включає в себе наступні підфункції: обробка даних, навчання нейромережі та тестування нейромережі.

Із побудованої моделі починається формування результатів, що складається із зв'язаних процесів: прогнозування, оцінка моделі та формування тренду.

Для побудови узагальненої архітектури використаємо StarUML. Застосунок буде мати просту клієнт-серверну архітектуру і відповідні два вузли (рисунок 2.3).

Клієнтський вузол буде складатися з наступних компонентів:

- Отримання файлу користувача («Data Collection»)
- Мережева взаємодія з сервером («Client Network»)
- Інтерфейс користувача («User Interface»)

Серверний вузол буде складатися з наступних компонентів:

- Підготовка даних («Data Preprocessing»)
- Побудова моделі («Model building»)
- Прогнозування («Prediction»)
- Бізнес-логіка сервера («Server Network»)

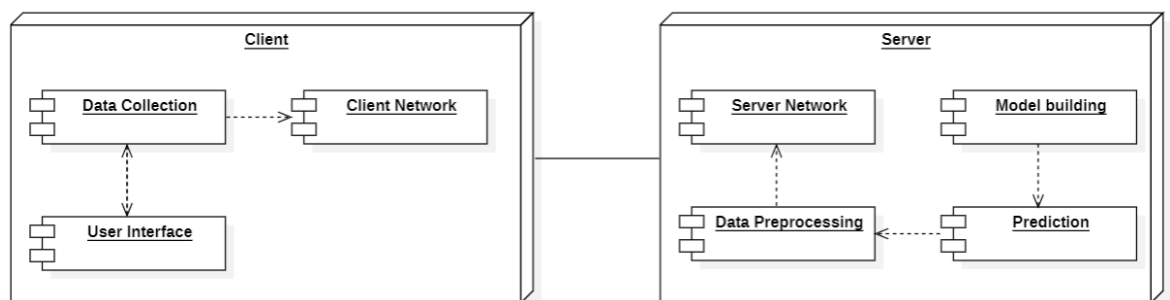


Рисунок 2.3 – Компонентна архітектура застосунку

Застосунок складатиметься з двох частин: клієнтської та серверної. Клієнтська частина відповідатиме за відображення та візуалізацію даних системи, а також надання користувачу основної функціональності застосунку. Користувач зможе локально переглядати інформацію про дані і вибрані дані будуть надсилатись на сервер для подальшої обробки.

На сервері будуть оброблятися дані системи, і за запитом від користувача буде виконуватися функція прогнозування з використанням складеної моделі. Результати цього прогнозу будуть автоматично надсилатись до клієнта. Крім того, сервер буде відповідальний за побудову моделі та виведення графіків з результатами.

Можна проілюструвати нейромережний застосунок за допомогою діаграми прецедентів (рисунок 2.4), яка буде описувати можливості користувача та системи:

1. Завантаження даних. Користувачу надається можливість вибору файлу даних з умовою попереднього форматування даних заданого формату. Якщо формат даних не є правильним, користувач повинен внести зміни і спробувати ще раз.
2. Нормалізація даних. Застосунок приймає та виконує нормалізацію даних.
3. Побудова моделі. Побудова моделі включає такі процеси як обробка даних та навчання моделі. Адмін застосунку може змінювати налаштування моделі.
4. Прогнозування. Проводиться прогнозування і формування тренду.
5. Отримання результатів. Користувач отримує результати прогнозування.

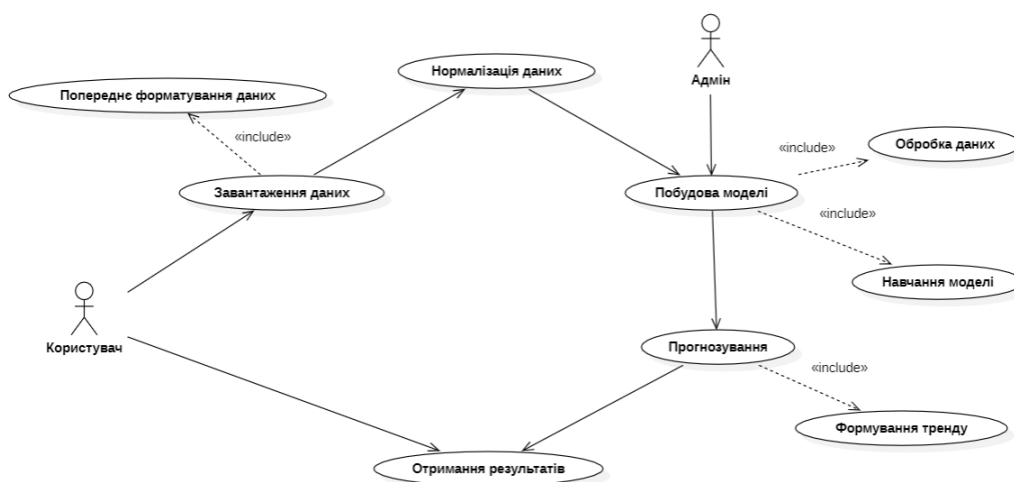


Рисунок 2.4 – Діаграма прецедентів

Для усвідомлення послідовності процесів нейромережного застосунку треба створити діаграму послідовності (рисунок 2.5).

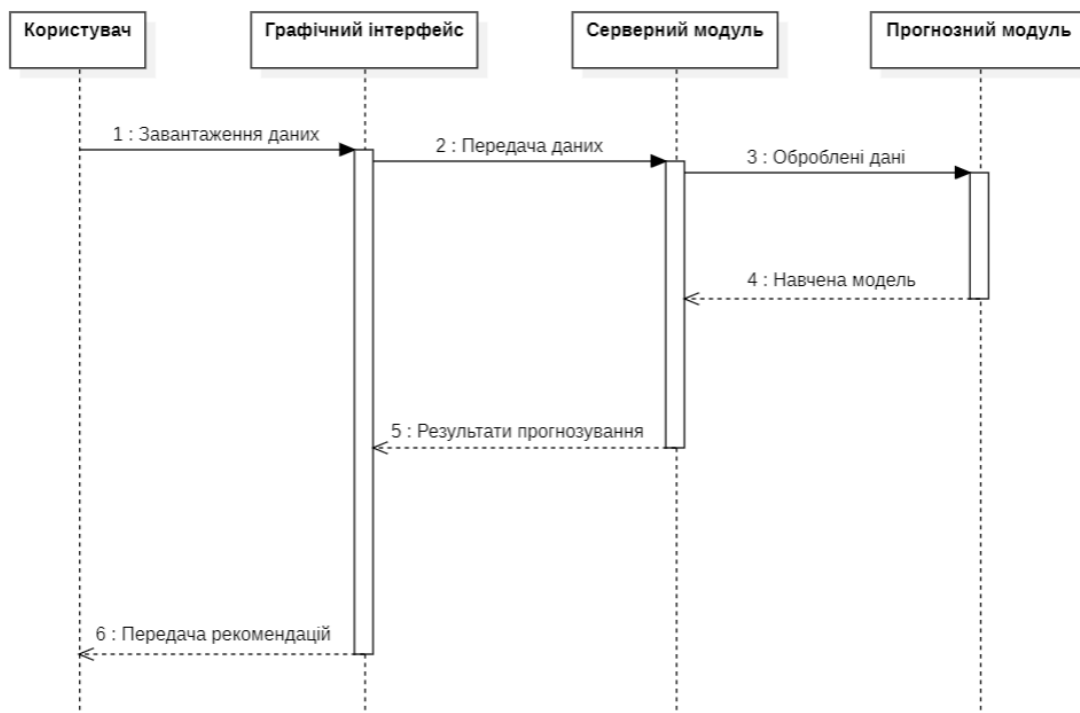


Рисунок 2.5 – Діаграма послідовності

З цієї діаграми можна зрозуміти, що процес починається з завантаження даних користувачем, які відображаються у розробленому графічному інтерфейсі застосунку. Потім дані передаються безпосередньо до серверного модулю, де відбувається обробка. Наступним кроком дані подаються в розроблений прогнозний модуль, після чого результати повертаються на вікно застосунку.

Для ілюстрації процесів, що включає нейромережний застосунок, була розроблена діаграма діяльності (рисунок 2.6). Характеризуючи процеси застосунку, варто відмітити, що вони мають послідовний характер і взаємозалежні. Після попереднього форматування даних, застосунок готовий для використання та завантаження даних, після чого виконується нормалізація даних. Відбувається обробка даних, які подаються у побудовану модель, вона навчається і після виконується прогнозування та формування тренду. Після усіх пройдених процесів отримуються результати.



Рисунок 2.6 – Діаграма діяльності

2.2. Побудова моделі

Перший крок — це підготовка даних для навчання та тестування. Оскільки задача спрогнозувати локальний тренд, буде взято дані всього за останні 10 місяців, саме за цей період Біткоїн зазнає відносно меншої зміни ціни через глобальних подій за останні кілька років (рисунок 2.7).

Total data for prediction: 273

	Date	Close
0	2022-01-01	47738.6
1	2022-01-02	47288.0
2	2022-01-03	46425.2
3	2022-01-04	45841.1
4	2022-01-05	43421.3
...
268	2022-09-26	19225.4
269	2022-09-27	19080.7
270	2022-09-28	19410.4
271	2022-09-29	19593.2
272	2022-09-30	19422.8

Рисунок 2.7 – Таблиця даних

Подивимося на розглядаємі дані графічно (рисунок 2.8).

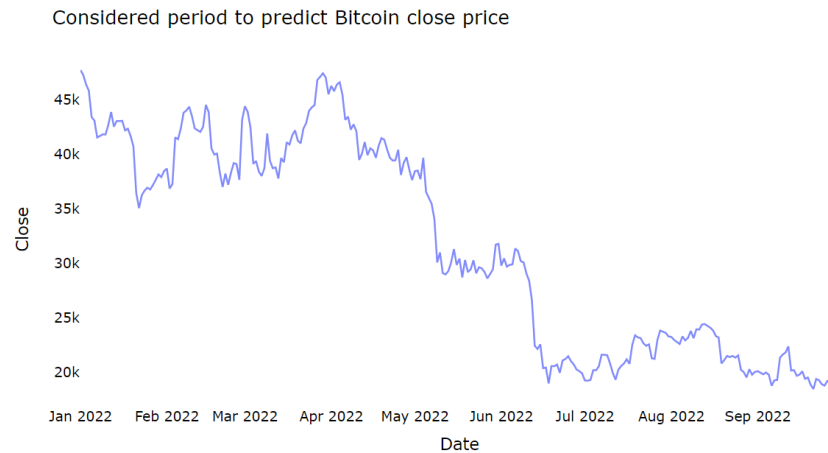


Рисунок 2.8 – Графік розглядаємих даних

Тепер треба нормалізувати набір даних, тобто змінити значення числових стовпців у наборі даних для використання загальної шкали без спотворення відмінностей у діапазонах значень або втрати інформації.

Використаємо MinMax Scaler, це техніка перетворення значень ознаки, що масштабує значення таким чином, що вони знаходяться в діапазоні від 0 до 1. Для кожного значення функція MinMax Scaler віднімає мінімальне значення функції, а потім ділить на діапазон [9]. Діапазон — це різниця між вихідним максимумом і початковим мінімумом. MinMax Scaler зберігає форму вихідного розподілу (додаток А).

Після чого поділимо дані на набір для навчання та набір для тестування. Було обрано 75% для навчального набору та 25% для тестового (додаток Б).

Тепер ми трансформуємо ціну закриття на основі вимог прогнозування аналізу часових рядів, включаючи часовий крок. Для LSTM часовий крок представляє кількість або довжину попередніх точок даних, включаючи поточну точку даних для прогнозування, візьмемо значення 15 (додаток В).

Також перетворимо масив значень у матрицю набору даних. Після чого змінимо форму вхідних даних на зразки, часові кроки та функції, що вимагається для LSTM (додаток Г).

Можемо перейти до фактичного складання моделі (додаток Д).

Тут описано десять прихованих шарів LSTM із функцією `relu` (функція активації випрямленої лінійної одиниці) і щільний одинарний шар.

Функція втрат обчислює величину, яку модель повинна прагнути мінімізувати під час навчання. Для оптимізації використовується алгоритм Адама.

Коли модель, шари, шари активації, функції втрат і оптимізатори визначені, модель починає навчання. Де поля посилаються на таке:

1. `epochs` — кількість циклів для навчання нейронної мережі з усіма даними навчання
2. `batch_size` — кількість зразків, які будуть поширені через мережу
3. `verbose` — деталізація результатів для кожного навчального процесу.

Після, можна порівняти втрати навчання проти втрати перевірки (рисунок 2.9), де:

- Втрата перевірки вказує на те, наскільки добре модель відповідає новим даним.
- Втрата навчання показує, наскільки добре модель відповідає даним навчання.

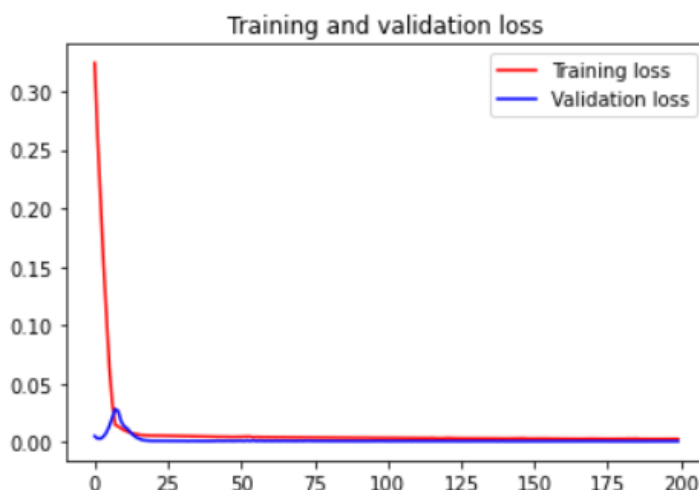


Рисунок 2.9 – Втрати навчання проти втрати перевірки

З кількістю циклів навчання, бачимо, що втрати моделі зменшуються та стають кращими.

Виконаємо прогнозування і перевіримо показники ефективності (додаток Е).

2.3. Оцінка моделі

Важливим кроком у роботі з моделями нейронних мереж є оцінка продуктивності моделі. Це дозволить порівняти з іншими моделями або порівняти, якщо вирішимо налаштувати модель. Тож повернемо дані до початкової форми (додаток Є).

Зазвичай використовуються такі показники, як середня абсолютна помилка (MAE), середня квадратична помилка (MSE) та корінь з середньої квадратичної помилки (RMSE) (додаток Ж).

MAE оцінює абсолютну відстань спостережень (записів набору даних) до прогнозів регресії, беручи середнє значення за всіма спостереженнями. Використовується абсолютне значення відстаней, щоб негативні похибки були враховані належним чином (формула 2.1).

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (2.1)$$

Інший спосіб — звести відстань у квадрат, щоб отримати позитивний результат. Це робиться за допомогою MSE, і вищі помилки (або відстані) мають більшу вагу в метриці, ніж менші, через природу степеневі функції (формула 2.2).

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (2.2)$$

RMSE використовується для повернення помилки MSE до вихідної одиниці шляхом вилучення квадратного кореня з неї, зберігаючи при цьому властивість штрафувати більші помилки (формула 2.3).

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}} \quad (2.3)$$

Також обрахуємо R-квадрат (Також відомий як коефіцієнт детермінації) — це статистичний показник відповідності, який показує, наскільки варіація залежної змінної пояснюється незалежною змінною (змінними) у регресійній моделі. У той час як кореляція пояснює силу зв'язку між незалежною та залежною змінною, R-квадрат пояснює ступінь, в якому дисперсія однієї змінної пояснює дисперсію другої змінної (формула 2.4). Так, якщо R² моделі дорівнює 0.50, то приблизно половина варіації, що спостерігається, може бути пояснена входами в модель (додаток 3).

$$R^2 = 1 - \frac{RSS}{TSS} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (2.4)$$

Значення свідчать про те, що ефективність ціни рухається відносно відповідно до індексу.

Оцінці моделі виглядають досить непогано, слід перейти до самого прогнозування.

2.4. Аналіз результатів моделі

Після перегляду оцінок моделі, можемо спробувати передбачити ціни закриття з нашої моделі LSTM, використовуючи параметри, визначені вище. Накладемо графіки один на одного (рисунок 2.10).



Рисунок 2.10 – Порівняння початкової ціни закриття і прогнозованої ціни закриття

Можемо побачити, що тенденція руху доволі схожа, враховуючи те що маємо дані лише за 10 місяців. Це сигналізує про ефективно складену модель яка прогнозує локальний тренд Біткоїна. Спробуємо спрогнозувати майбутні 30 днів (рисунок 2.11).

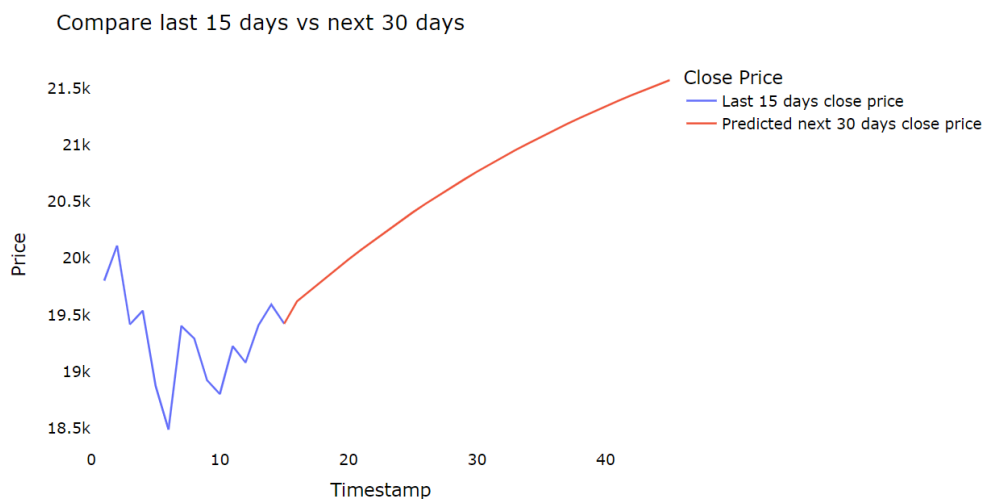


Рисунок 2.11 – Порівняння останніх 15 днів з спрогнозованими

Як бачимо модель спрогнозувала невелике зростання, що означає що тренд висхідний. Тобто інвестору треба купувати актив, щоб заробити, або дочекатися більш впевненого зростання, оскільки модель є приблизною.

Тепер спробуємо накласти спрогнозований тренд на весь графік, щоб побачити більш глобальну ситуацію (рисунок 2.12).

	Date	Close	Predict
0	2022-01-01	47738.6	
1	2022-01-02	47288.0	
2	2022-01-03	46425.2	
3	2022-01-04	45841.1	
4	2022-01-05	43421.3	
...
298	2022-10-26		21390.554097
299	2022-10-27		21438.152244
300	2022-10-28		21484.409285
301	2022-10-29		21529.35878
302	2022-10-30		21573.048673

Рисунок 2.12 – Нова таблиця

Таблицю створили, тепер подивимось на графік (рисунок 2.13).



Рисунок 2.13 – Ціна Біткоїна закриття порівняно з наступними 30 днями

Дійсно, тепер наочно і чітко видно спрогнозований тренд, який свідчить про невелике зростання курсу Біткоїна. Але це за умови, що на його ціну не вплинуть якісь значні глобальні події, про які згадувалось вище. В такому випадку тенденція руху може відхилитися вниз, так і навпаки вгору.

Висновки за розділом 2

В результаті виконання другого розділу дипломної роботи було описано архітектуру нейромережного застосунку, використовуючи як словесний опис, так і UML діаграми. Досліджено внутрішні процеси, взаємодію користувача з застосунком, а також послідовність процесів, які відбуваються під час роботи застосунку.

Також було складено та описано модель і графічно порівняно втрати навчання проти втрат перевірки. В кінці кінців, було здійснено прогнозування, перевірено показники ефективності і оцінено модель. Після чого було переглянуто результати прогнозування.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖНОГО ЗАСТОСУНКУ ПРОГНОЗУВАННЯ КРИПТОВАЛЮТ

3.1. Вибір середовища та інструментів для розробки застосунку

Для реалізації нейромережного застосунку було вирішено використовувати середовище розробки Visual Studio Code (VS Code). VS Code є потужним і популярним інтегрованим середовищем розробки з великим спектром функціональних можливостей. Було обрано VS Code з-за його легкості використання, розширюваності та підтримки багатьох мов програмування, включаючи Python. Це надійний і зручний інструмент, який дозволить зосередитись на розробці без зайвих перешкод.

Python. Мова програмування Python була обрана з-за її простоти, читабельності і широкого спектру бібліотек та фреймворків. Python є популярною мовою програмування з багатьма перевагами, але вона також має свої обмеження.

Переваги Python:

- Простота вивчення: Python має чистий і зрозумілий синтаксис, що сприяє швидкому навчанню і розробці.
- Багатофункціональність: Python пропонує широкий спектр бібліотек і модулів для різних завдань, що робить його універсальним і ефективним для різних проектів.
- Велика спільнота: Python має активну спільноту розробників, що забезпечує велику кількість документації, пакетів і підтримку.

Недоліки Python:

- Швидкодія: Порівняно з деякими іншими мовами програмування, Python може бути повільнішим через інтерпретацію коду. Однак, зазвичай це не є критичним фактором для багатьох додатків.

- Обмеження в ресурсах: У випадку великих обчислювальних завдань або потреби в максимальній продуктивності, Python може бути обмежений у порівнянні з мовами, які компілюються.

Основні бібліотеки які будуть використані для реалізації нейромережного застосунку:

- Pandas - це бібліотека для маніпулювання та аналізу даних. Вона надає високопродуктивні структури даних, такі як DataFrames, і функціонал для обробки й аналізу даних.
- NumPy - це потужна бібліотека для наукових обчислень в Python. Вона надає підтримку для роботи з масивами та математичними функціями, що дозволяють ефективно операції над числовими даними.
- TensorFlow - це бібліотека для глибокого навчання та розв'язання складних завдань машинного навчання. Вона має високоефективні інструменти для побудови та тренування нейронних мереж.
- Scikit-learn - це бібліотека машинного навчання, яка надає широкий спектр алгоритмів для класифікації, регресії, кластеризації та інших завдань. Вона також містить інструменти для попередньої обробки даних та оцінки моделей.
- Plotly - це бібліотека для візуалізації даних, яка дозволяє створювати інтерактивні графіки та візуалізації. Вона надає можливості для побудови графіків, діаграм, географічних карт і багато іншого.

Flask. В якості веб-фреймворку було обрано Flask, для розробки застосунку через його простоту, гнучкість та підтримку Python. Flask дозволяє швидко створювати веб-додатки, забезпечуючи базовий функціонал із можливістю розширення за допомогою різних розширень. Враховуючи складність проекту, Flask буде ефективним вибором для реалізації веб-функціональностей.

3.2. Фізична структура застосунку

Застосунок складається із наступних об'єктів (рисунок 3.1):

- Папки
 - «__pycache__» — кеш-папка. Вона використовується для зберігання скомпільованих версій файлів модулів з метою покращення швидкодії виконання програми. Кеш-папка автоматично створюється Python і містить скомпільовані файли з байт-кодом, які можуть бути безпосередньо виконані інтерпретатором при наступних запусках програми.
 - «static» — папка статичних файлів. Розміщення статичних файлів у папці допомагає організувати структуру проекту і забезпечує консистентність шляхів до цих файлів у коді програми. Крім того, фреймворки, такі як Flask, зазвичай мають вбудовану функціональність для обробки статичних файлів, що спрощує включення їх у веб-сторінки.
 - «templates» — папка шаблонів. Використовується для зберігання файлів шаблонів, які визначають структуру і вигляд веб-сторінок.
- Python скрипти
 - «lstm» — прогнозний модуль. Файл містить усі необхідні функції які застосовуються в головному файлі застосунку. Тобто записує та оброблює дані, складає та навчає модель, прогнозує та формує результати.
 - «app» — головний файл додатку або серверний скрипт. Його завданням є налаштування веб-сервера і обробка запитів користувачів.

В даному випадку папка «static» буде мати логотип застосунку та файл CSS, який буде використовуватися для зовнішнього оформлення веб-сторінок. Також в папці буде зберігатися отриманий файл користувача і оброблені файли, які будуть необхідні для роботи серверному модулю.

В папці «templates» буде містити файли HTML, які використовуються для визначення шаблонів веб-сторінок у веб-розробці. Вони містять HTML-код разом з певними місцями для динамічного вставлення даних під час генерації сторінок.

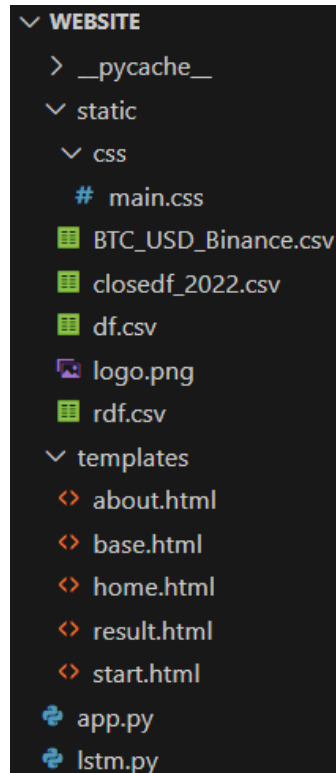


Рисунок 3.1 – Фізична структура застосунку

3.3. Програмна реалізація інтерфейсу користувача

Застосунок має 4 сторінки, а саме «home», «about», «start» та «result». Усі сторінки будуть частково наслідувати елементи сторінки «base», що сприятиме створенню єдиної інтерфейсної концепції для всього застосунку.

Зобразимо інформаційну схему застосунку, яка відображає взаємодію між перерахованими сторінками розробленого застосунку (рисунок 3.2).

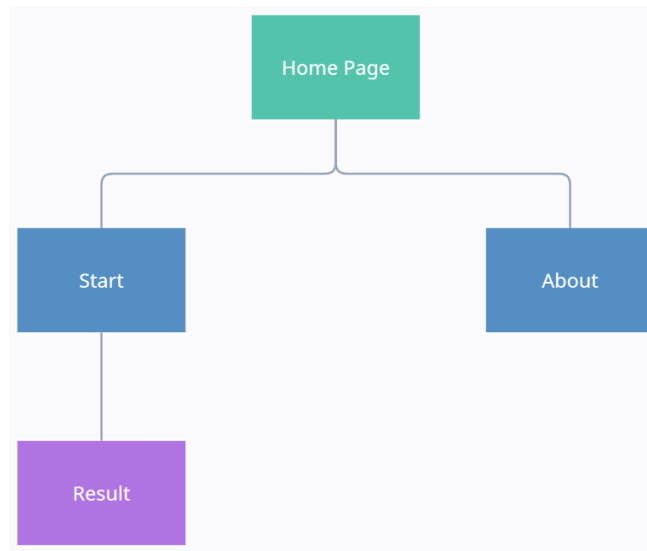


Рисунок 3.2 – Інформаційна схема застосунку

Ця інформаційна схема відображає послідовність переходів між сторінками та взаємодію з користувачем на розробленому застосунку, створюючи зручний та логічний досвід користувачу. Тепер розглянемо створений застосунок.

Сторінка «home» буде служити початковою точкою для користувача, де він зможе почати роботу з прогнозуванням, або перейти на інформаційну сторінку (рисунок 3.3).

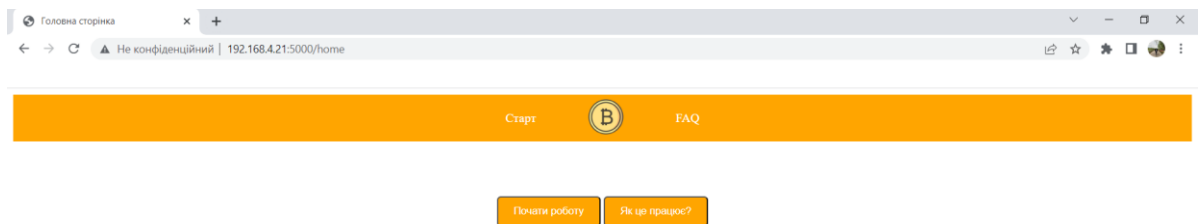


Рисунок 3.3 – Сторінка «home»

На сторінці «about» будуть розміщуватись головні запитання, які повністю характеризують цілі та методи застосунку, а також обмеження щодо формату файлу, який користувач буде завантажувати, і відповідно будуть відповіді до даних запитань (рисунок 3.4).

Серед запитань будуть: «Що робить цей сайт?», «Якого формату треба завантажити файл?», «Яким методом виконується прогнозування?» та «Який може бути результат?»

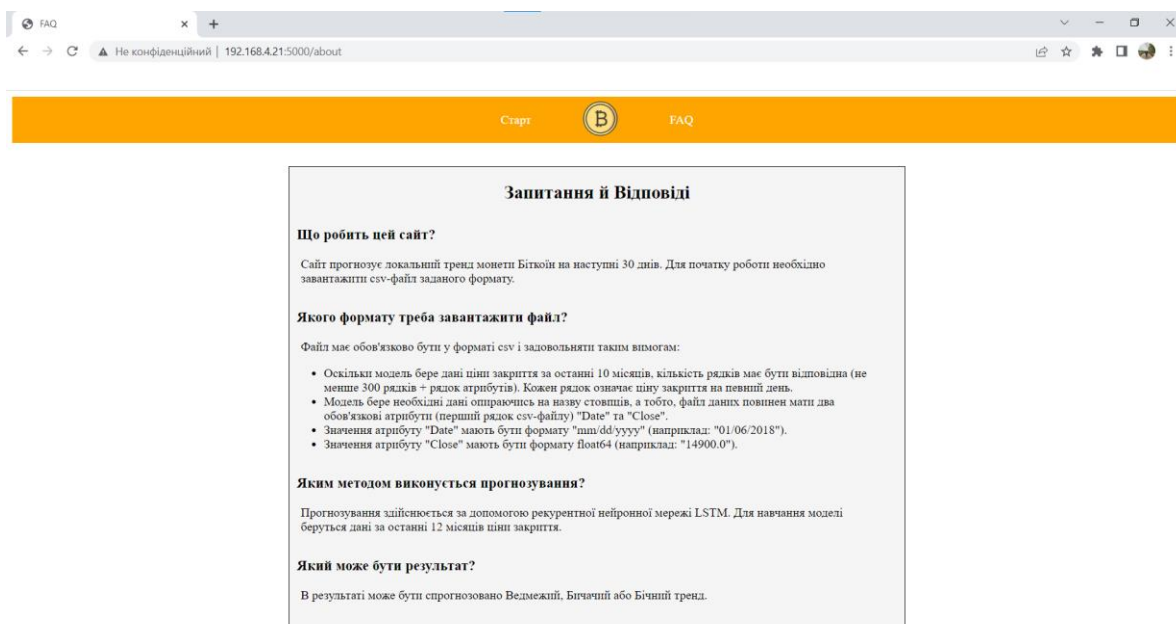


Рисунок 3.4 – Сторінка «about»

Після розуміння користувачем сутності застосунку він може перейти на сторінку «start», де він може розпочати роботу (рисунок 3.5).

На сторінці «start» користувач зможе розпочати взаємодію з застосунком, завантаживши файл з даними. На сторінці можна побачити, що поки користувач не завантажить файл, йому не буде доступна кнопка «Почати». Також потрібно враховувати, що якщо користувач завантажить файл неправильного формату, після натискання кнопки «Почати» на сторінці нічого не з'явиться і користувач повинен внести зміни і спробувати ще раз.

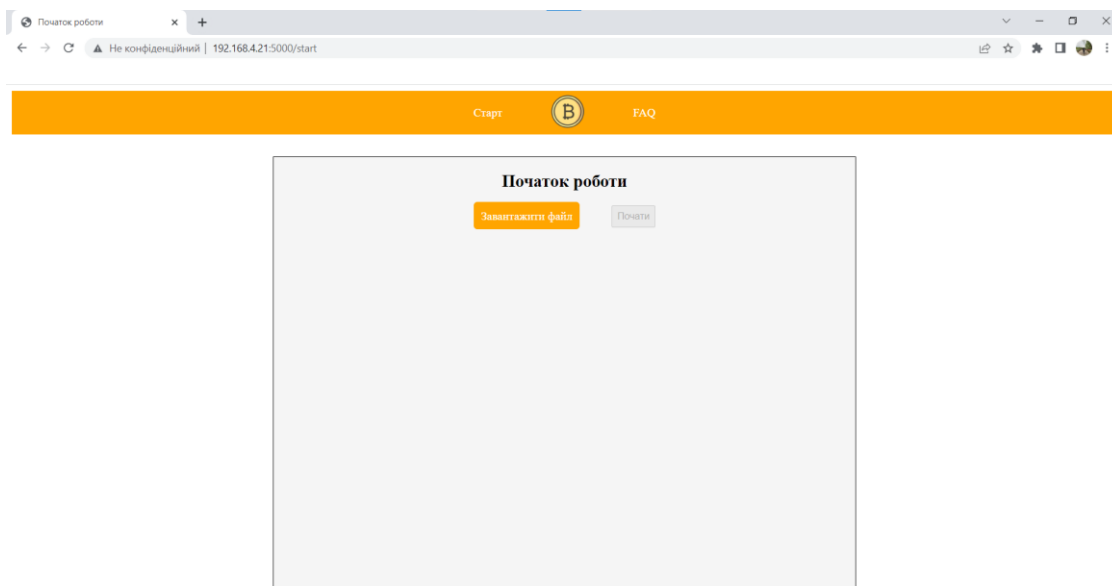


Рисунок 3.5 – Сторінка «start»

Якщо ж користувач завантажить файл правильного формату, через деякий час на сторінці з'являться графік функції завантажених даних та кнопка «Почати прогнозування» (рисунок 3.6). В цей момент серверний модуль отримує дані, та починає їх оброблювати. Після чого зберігає необхідні копії в папку «static» та повертає графік функції даних з якими буде працювати.

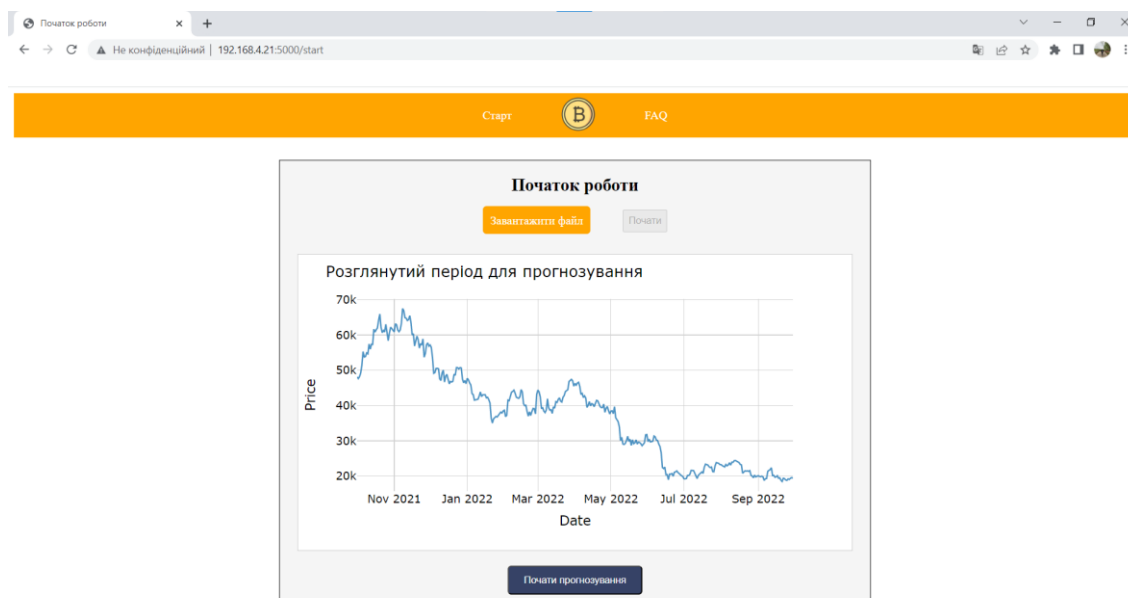


Рисунок 3.6 – Сторінка «start» після завантаження файлу

Після натискання на кнопку «Почати прогнозування» серверний модуль задіє функції прогнозного модулю і почнеться прогнозування. Це займає деякий

час, але враховуючи вимоги нейромережного застосунку, він буде виконувати прогнозування впродовж 3 хвилин.

Після виконання всіх необхідних процесів у застосунку, користувача буде автоматично перенаправлено на сторінку «result», де він зможе ознайомитись з отриманими результатами прогнозування (рисунки 3.7).



Рисунок 3.7 – Сторінка «result»

На сторінці буде знаходитися доповнений графік функції з додатковими 2-ма лініями. При перетинанні кривої лінії «Bull» прогнозований тренд буде Бичачим, при перетинанні лінії «Bear» прогнозований тренд буде Ведмежим. У разі коли крива в кінцевому рахунку залишається між обома лініями прогнозований тренд буде Бічним, в такому разі користувачу буде рекомендовано не робити жодних дій.

3.4. Тестування та аналіз результату роботи нейромережного застосунку прогнозування курсу криптовалют

Оскільки дані в нашому випадку залишаються незмінними, необхідно змоделювати сценарій тестування, що дозволяє оцінити роботу нейромережного застосунку при різних проміжках часу. Було вирішено взяти наступні проміжки часу:

1. 12/2021 – 09/2022 (таблиця 3.1).

Таблиця 3.1 – План 1-го тестування

Мета тесту	Перевірка коректності прогнозування нейромережного застосунку
Вхідні дані	Дані на проміжку 12/2021 – 09/2022
Очікуваний результат	Бічний тренд

Результат: Бічний тренд (рисунок 3.8). Оскільки крива залишається між лініями «Bull» та «Bear» тренд буде бічним. Для наочності порівняємо з цим самим проміжком справжніх даних (рисунок 3.9). З рисунку видно, що там також бічний тренд.



Рисунок 3.8 – Результат 1-го тестування



Рисунок 3.9 – Справжні дані 1-го тестування

2. 10/2021 – 07/2022 (таблиця 3.2).

Таблиця 3.2 – План 2-го тестування

Мета тесту	Перевірка коректності прогнозування нейромережного застосунку
Вхідні дані	Дані на проміжку 10/2021 – 07/2022
Очікуваний результат	Ведмежий тренд

Результат: Бичачий тренд (рисунок 3.10). Крива перетнула лінію «Bull», тому тренд буде бичачим. Для наочності порівняємо з цим самим проміжком справжніх даних (рисунок 3.11). З рисунку видно, що там протилежна ситуація, це пов'язано з тим, що ринок криптовалют на той момент зазнав падіння через вплив глобальних подій. Якби не ці події, ринок не впав би так сильно і, можливо, тренд був би бичачим.



Рисунок 3.10 – Результат 2-го тестування



Рисунок 3.11 – Справжні дані 2-го тестування

3. 05/2021 – 02/2022 (таблиця 3.3).

Таблиця 3.3 – План 3-го тестування

Мета тесту	Перевірка коректності прогнозування нейромережного застосунку
Вхідні дані	Дані на проміжку 05/2021 – 02/2022
Очікуваний результат	Бічний тренд

Результат: Бічний тренд (рисунок 3.12). Оскільки крива залишається між лініями «Bull» та «Bear» тренд буде бічним. Для наочності порівняємо з цим самим проміжком справжніх даних (рисунок 3.13). З рисунку видно, що там також бічний тренд.



Рисунок 3.12 – Результат 3-го тестування



Рисунок 3.13 – Справжні дані 3-го тестування

4. 03/2019 – 12/2019 (таблиця 3.4).

Таблиця 3.4 – План 4-го тестування

Мета тесту	Перевірка коректності прогнозування нейромережного застосунку
Вхідні дані	Дані на проміжку 03/2019 – 12/2019
Очікуваний результат	Бичачий тренд

Результат: Бичачий тренд (рисунок 3.14). Крива перетнула лінію «Bull», тому тренд буде бичачим. Для наочності порівняємо з цим самим проміжком справжніх даних (рисунок 3.15). З рисунку видно, що там також бичачий тренд.



Рисунок 3.14 – Результат 4-го тестування



Рисунок 3.15 – Справжні дані 4-го тестування

5. 11/2020 – 08/2021 (таблиця 3.5).

Таблиця 3.5 – План 1-го тестування

Мета тесту	Перевірка коректності прогнозування нейромережного застосунку
Вхідні дані	Дані на проміжку 11/2020 – 08/2021
Очікуваний результат	Бічний тренд

Результат: Бічний тренд (рисунок 3.16). Оскільки крива залишається між лініями «Bull» та «Bear» тренд буде бічним. Для наочності порівняємо з цим самим проміжком справжніх даних (рисунок 3.17). З рисунку видно, що там також бічний тренд.



Рисунок 3.16 – Результат 5-го тестування



Рисунок 3.17 – Справжні дані 5-го тестування

В результаті тестування було спрогнозовано правильно 4 з 5-ти трендів. Це ще раз підкреслює правильність зробленого застосунку, за умови що на ринок не будуть впливати ніякі події.

3.5. Інструктивний матеріал користувача для роботи з нейромережним застосунком прогнозування курсу криптовалют

Використання застосунку було максимально спрощено для потенційного користувача (трейдера чи інвестора), який просто повинен зайти у застосунок, після чого перейти на сторінку «start» та завантажити файл заданого формату, файл має обов'язково бути у форматі csv і задовольняти таким вимогам:

- Оскільки модель бере дані ціни закриття за останні 10 місяців, кількість рядків має бути відповідна (не менше 300 рядків + рядок атрибутів). Кожен рядок означає ціну закриття на певний день.
- Модель бере необхідні дані опираючись на назву стовпців, а тобто, файл даних повинен мати два обов'язкові атрибути (перший рядок csv-файлу) "Date" та "Close".
- Значення атрибуту "Date" мають бути у форматі "mm/dd/yyyy" (наприклад: "01/06/2018").
- Значення атрибуту "Close" мають бути у форматі float64 (наприклад: "14900.0").

Користувач буде мати можливість переглянути дані на графіку і запевнитись у правильності вибору файлу.

Після чого користувачу треба лише натиснути кнопку «Почати прогнозування» і зачекати деякий час (не більше 3 хвилин). Після не тривалого очікування користувач зможе переглянути результати.

У разі виникнення питань щодо застосунку, весь матеріал дублюється для користувача на сторінці «about».

Користувач може використовувати даний застосунок для прогнозування тренду будь-якої криптовалюти, але оскільки нейромережний застосунок

розроблювався для прогнозування тренду саме Біткоїну, точність прогнозування для інших криптовалют буде меншою.

Висновки за розділом 3

В рамках третього розділу було описано використані інструменти для розробки нейромережного застосунку, були розглянуті як і фізична структура застосунку, так і програмна реалізація інтерфейсу користувача, де було показано з яких об'єктів складається застосунок, які сторінки існують та як вони взаємодіють.

Було проведено тестування та аналіз результату роботи нейромережного застосунку, в якому було продемонстровано 5 тестів, 4 з яких показали відмінний результат.

Також було описано інструктивний матеріал користувача для роботи з нейромережним застосунком, в якому було наведено правильний формат файлу, який необхідно завантажити.

ВИСНОВКИ

У результаті виконання дипломної роботи було отримано нейромережний застосунок прогнозування криптовалют на прикладі Біткоїн.

Був проведений аналітичний огляд дипломної роботи, в ході якого були розглянуті різні підходи, рішення та результати, що досягнуто на даний момент часу. Були проаналізовані наявні методи та підходи для досягнення кращого результату.

Також був проведений функціональний аналіз та визначені вимоги до застосунку. На основі цього була розроблена архітектура застосунку, визначені зв'язки між компонентами системи та створена документація для користувача. Розроблено застосунок із використанням мов програмування Python та веб-фреймворка Flask.

Даний застосунок можна буде в подальшому використовувати учасникам криптовалютного ринку для прийняття успішних торгових рішень. З технічної точки зору даний застосунок можна покращити шляхом оптимізації коду та залученню нових методів прогнозування, що в подальшому підвищить точність прогнозування.

Точність нейронної мережі була підвищена і продемонструвала непогані результати, але при врахуванні деяких факторів впливу на ціну, які змінюють тенденцію ринку, точність застосунку може підвищитись.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Michael Crosby, Pradhan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman // BlockChain Technology – 2015 – p. 35.
2. Aleksander Berentsen and Fabian Schär // A Short Introduction to the World of Cryptocurrencies – 2018 – p. 16.
3. Satoshi Nakamoto // Bitcoin: A Peer-to-Peer Electronic Cash System – 2008 – p. 9.
4. Chris Chatfield // Time-Series Forecasting – 2000 – p. 280.
5. Omer Berat Sezer, M. Ugur Gudelek, Ahmet Murat Ozbayoglu // Financial Time Series Forecasting with Deep Learning – 2019 – p. 63.
6. Sakib Ashraf Zargar // Introduction to Sequence Learning Models: RNN, LSTM, GRU – 2021 – p. 14.
7. Christopher Olah // Understanding LSTM Networks – 2015 [Електронний ресурс]. Режим доступу: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/#fnref1>.
8. Ralf C. Staudemeyer, Eric Rothstein Morris // Understanding LSTM: a tutorial into Long Short-Term Memory Recurrent Neural Networks – 2019 – p. 42.
9. Zexin Hu, Yiqi Zhao, Matloob Khushi // A Survey of Forex and Stock Price Prediction Using Deep Learning – 2021 – p. 30.
10. Опис мови програмування Python [Електронний ресурс]. Режим доступу: <https://disted.edu.vn.ua/courses/learn/7649>.
11. Francois Chollet // Deep Learning with Python – 2017 – p. 110.
12. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov // Improving neural networks by preventing co-adaptation of feature detectors – 2012 – p. 18.
13. L. Breiman // Random Forests – 2001 – p. 32.
14. S. Hochreiter and J. Schmidhuber // Long short-term memory – 1997 – p. 1735.

15. Miguel Grinberg // Flask Web Development with Python Tutorial [Электронный ресурс]. Режим доступа: <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>.

16. Wes McKinney // Python for Data Analysis – 2017 – p. 453.

17. Travis E. Oliphant // Guide to NumPy – 2006 – p. 378.

18. Aurélien Géron // Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – 2019.

19. Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram // Plotly for R and Python [Электронный ресурс]. Режим доступа: <https://plotly.com/python/>.

ДОДАТКИ

Додаток А

Лістинг «Нормалізації за допомогою MinMax Scaler»

```
1 # видалення стовпця дати та нормалізація за допомогою MinMax Scaler
2 del closedf['Date']
3 scaler=MinMaxScaler(feature_range=(0,1))
4 closedf=scaler.fit_transform(np.array(closedf).reshape(-1,1))
5 print(closedf.shape)
```

(273, 1)

Додаток Б

Лістинг «Поділ даних на набори для навчання та для тестування»

```
1 # Поділ даних на набір для навчання(75%) та набір для тестування(25%)
2 training_size=int(len(closedf)*0.75)
3 test_size=len(closedf)-training_size
4 train_data,test_data=closedf[0:training_size:],closedf[training_size:len(closedf),:1]
5 print("train_data: ", train_data.shape)
6 print("test_data: ", test_data.shape)
```

```
train_data: (204, 1)
test_data: (69, 1)
```

Додаток В

Лістинг «Трансформування ціни закриття»

```
1 # Тепер ми трансформуємо ціну закриття на основі вимог прогнозування аналізу часових рядів. Візьмемо 15
2
3 # Перетворення масиву значень у матрицю набору даних
4 def create_dataset(dataset, time_step=1):
5     dataX, dataY = [], []
6     for i in range(len(dataset)-time_step-1):
7         a = dataset[i:(i+time_step), 0] ###i=0, 0,1,2,3-----99 100 |
8         dataX.append(a)
9         dataY.append(dataset[i + time_step, 0])
10    return np.array(dataX), np.array(dataY)
11
12 time_step = 15
13 X_train, y_train = create_dataset(train_data, time_step)
14 X_test, y_test = create_dataset(test_data, time_step)
15
16 print("X_train: ", X_train.shape)
17 print("y_train: ", y_train.shape)
18 print("X_test: ", X_test.shape)
19 print("y_test", y_test.shape)
```

```
X_train: (188, 15)
y_train: (188,)
X_test: (53, 15)
y_test (53,)
```

Додаток Г

Лістинг «Зміна форми вхідних даних»

```
1 # Змінимо форму вхідних даних на [зразки, часові кроки, функції], що вимагається для LSTM
2 X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
3 X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
4
5 print("X_train: ", X_train.shape)
6 print("X_test: ", X_test.shape)
```

```
X_train: (188, 15, 1)
X_test: (53, 15, 1)
```

Додаток Д

Лістинг «Складання моделі»

```
1 # Фактичне складання моделі
2 model=Sequential()
3 # Додавання шару LSTM з 10 внутрішніми блоками
4 model.add(LSTM(10,input_shape=(None,1),activation="relu"))
5 # Додавання щільного шару з 1 блоком.
6 model.add(Dense(1))
7 # Функція втрати + оптимізатор
8 model.compile(loss="mean_squared_error",optimizer="adam")
9
10 history = model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=200,batch_size=32,verbose=1)
```

Додаток Е

Лістинг «Прогнозування і перевірка показників ефективності»

```
1 # Зробимо прогноз і перевіримо показники ефективності
2 train_predict=model.predict(X_train)
3 test_predict=model.predict(X_test)
4 train_predict.shape, test_predict.shape

6/6 [=====] - 1s 7ms/step
2/2 [=====] - 0s 9ms/step
```

Додаток Є

Лістинг «Перетворення назад до початкової форми»

```
1 # Перетворення назад до початкової форми
2 train_predict = scaler.inverse_transform(train_predict)
3 test_predict = scaler.inverse_transform(test_predict)
4 original_ytrain = scaler.inverse_transform(y_train.reshape(-1,1))
5 original_ytest = scaler.inverse_transform(y_test.reshape(-1,1))
```

Додаток Ж

Лістинг «Метрики оцінки RMSE, MSE та MAE»

```
1 # Метрики оцінки RMSE, MSE та MAE
2 print("Train data RMSE: ", math.sqrt(mean_squared_error(original_ytrain,train_predict)))
3 print("Train data MSE: ", mean_squared_error(original_ytrain,train_predict))
4 print("Train data MAE: ", mean_absolute_error(original_ytrain,train_predict))
5 print("-----")
6 print("Test data RMSE: ", math.sqrt(mean_squared_error(original_ytest,test_predict)))
7 print("Test data MSE: ", mean_squared_error(original_ytest,test_predict))
8 print("Test data MAE: ", mean_absolute_error(original_ytest,test_predict))
```

```
Train data RMSE: 1532.0298689002475
Train data MSE: 2347115.5192025094
Train data MAE: 1153.4120491190156
```

```
-----
Test data RMSE: 856.0076674767206
Test data MSE: 732749.1267789359
Test data MAE: 654.8602520636794
```

Додаток 3

Лістинг «R квадратний бал для регресії»

```
1 # R квадратний бал для регресії
2 print("Train data R2 score:", r2_score(original_ytrain, train_predict))
3 print("Test data R2 score:", r2_score(original_ytest, test_predict))
```

Train data R2 score: 0.967380259834233

Test data R2 score: 0.7649987232872971