

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

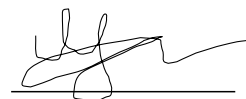
На правах рукопису  
УДК 004.852

До захисту допущено  
В.о. зав. кафедри ШІ  
О.І. ЧУМАЧЕНКО

«\_\_\_» \_\_\_\_\_ 2022 р.

**Магістерська дисертація**  
на здобуття ступеня магістра за спеціальністю 122 «Комп'ютерні науки»  
на тему: «Моделювання та прогнозування курсу криптовалют GameFi  
сегменту»

Виконала:  
студентка 2 курсу, групи КІ-11 мп  
Яцько Яна Віталіївна



Керівник: професор кафедри ММСА,  
д.т.н., доц. Терентьев О.М.



Рецензент: с.н.с. відділу  
прикладної інформатики Інституту  
телекомунікацій і глобального  
інформаційного простору  
к.е.н., доц. Тетяна Іванівна Просянкіна-Жарова



Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів  
без відповідних посилань  
Студент (підпис):



Київ  
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ ІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти — другий (магістерський)  
Спеціальність (ОПП) — 122 «Комп'ютерні науки» («Системи і методи штучного інтелекту»)

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ШІ

Олена ЧУМАЧЕНКО

«\_\_\_» \_\_\_\_\_ 2022 р.

### ЗАВДАННЯ

на магістерську дисертацію студентці Яцько Яні Віталіївни

**1. Тема дисертації:** «Моделювання та прогнозування курсу криптовалют GameFi сегменту», науковий керівник дисертації Терентьев Олександр Миколайович, д.т.н., доцент, затверджені наказом по університету від «03» листопада 2022 р. № 4046-с.

**2. Термін подання студентом дисертації:** 15.12.2022 р.

**3. Об'єкт дослідження:** статистичні дані, аналітичні та фінансові звіти, дані публікацій соцмережі Twitter "Twitter User Data".

**4. Предмет дослідження:** методи аналізу та прогнозування часових рядів, гібридні рекурентні нейронні мережі.

**5. Перелік завдань, які потрібно розробити:**

1) огляд літератури, аналіз готових аналітичних платформ та тенденцій на ринку криптовалют.

2) обґрунтування актуальності обраної теми.

3) огляд існуючих методів для прогнозування курсу криптовалют.

4) додавання даних з мережі Twitter для підвищення точності моделей.

5) прогнозування курсу криптовалют, валідація моделей.

6) вибір найкращої моделі та апробація розробленої програми на реальних даних, розробка аналітичного застосунку.

**6. Орієнтовний перелік графічного (ілюстративного) матеріалу:**

1) Динаміка курсу криптовалют GameFi сегменту.

- 2) Структура даних, взятих з порталу Binance за допомогою Binance-API.
- 3) Архітектура моделей, використаних для прогнозування.
- 4) Архітектура власної запропонованої рекурентної мережі.
- 5) Метрики якості в задачі прогнозування
- 6) Модель класифікації текстових даних
- 7) Отримані результати (таблиця та графіки)

### 7. Орієнтовний перелік публікацій:

Участь у міжнародній науково-практичній конференції та публікація тез доповіді у збірнику матеріалів конференції.

**8. Дата видачі завдання:** 1 жовтня 2022 р.

### Календарний план

| № з/п | Назва етапів виконання магістерської дисертації  | Термін виконання етапів МДР | Примітка |
|-------|--|-----------------------------|----------|
| 1     | Інструктаж з техніки безпеки   | 01.09.2022-02.09.2022       | виконано |
| 2     | Затвердження теми магістерської дипломної роботи (МДР)<br>Ознайомлення зі структурою МДР згідно з Положення про випускню атестацію студентів КПІ ім. Ігоря Сікорського | 03.09.2022                  | виконано |
| 3     | Ознайомлення з ДСТУ 3008-2015 та стандарти Єдиної системи програмної документації (ЄСПД)   | 04.09.2022-09.09.2022       | виконано |
| 4     | Ознайомлення з державним стандартом України ДСТУ 8302:2015   | 10.09.2022-15.09.2022       | виконано |
| 5     | Проведення дослідження за темою МДР під керівництвом керівника   | 16.09.2022-10.10.2022       | виконано |
| 6     | Завершення роботи над першим варіантом основної частини МДР  | 11.10.2022-25.10.2022       | виконано |
| 7     | Продовження роботи над експериментальною частиною МДР та програмним забезпеченням.   | 26.10.2022-15.11.2022       | виконано |
| 8     | Оформлення звіту з переддипломної практики. Підготовка матеріалів до захисту   | 15.11.2022-15.12.2022       | виконано |

Студент

Науковий керівник дисертації




Яна ЯЦЬКО

Олександр ТЕРЕНТЬЄВ

## АНОТАЦІЯ

Дипломна робота: 117 с., 28 табл., 23 рис., 3 додатки, 64 джерела.

### GAMEFI, КРИПТОВАЛЮТА, LSTM, RNN, TWITTER

Актуальність теми: останнім часом у світі стрімко розвивається напрямок торгівлі на криптобіржах. Так як ринок є волатильним, потрібно мати інструмент що допоможе аналізувати та приймати рішення з покупки/продажу. Модель прогнозування курсу криптовалют GameFi сегменту допомагає трейдеру планувати покупки/продажі з ціллю отримати прибуток. Рішення допоможе користувачеві проаналізувати криптовалюту в даному сегменті та, спираючись на прогноз, прийняти рішення щодо покупки/продажі криптовалют.

Мета даної роботи полягає у дослідженні існуючих рішень в області аналізу динаміки курсу криптовалют, побудови моделей машинного навчання, збагачення даних та підвищення якості прогонзування курсу криптовалют за допомогою бази Twitter та розробці зручного інструменту для аналізу та планування торгів.

Об'єктом дослідження є статистичні дані, аналітичні та фінансові звіти, дані публікацій соцмережі Twitter "Twitter User Data".

Предметом дослідження є методи аналізу табличних даних, регресійні моделі, гібридні рекурентні нейронні мережі.

Отриманні результати – створена програма для візуалізації динамічних характеристик криптовалют, скрипт прогнозування курсу криптовалют, що має високу точність прогнозування (оцінюючи метрикою RMSE).

Програмний продукт реалізований мовою програмування Python3.8 у середовищі розробки DataSpell 2022. Для аналізу та побудови моделей було використано середовище Jupyter Notebook.

Отримані результати: розроблено простий інтерфейс для аналізу статистичних і динамічних характеристик монет, що включає в себе прогнози на майбутнє (на обрану кількість кроків).

## ABSTRACT

Thesis: 117 pp., 28 tables, 23 figures, 3 appendixes, 64 sources.

GAMEFI, CRYPTOCURRENCY, LSTM, RNN, TWITTER

Relevance of the topic: Recently, the direction of trading on crypto exchanges has been developing rapidly in the world. Since the market is volatile, you need to have a tool that will help you analyze and make buy/sell decisions. The GameFi segment cryptocurrency rate forecasting model helps the trader to plan buys/sells in order to make a profit. The solution will help the user to analyze cryptocurrencies in a given segment and, based on the forecast, make a decision to buy/sell cryptocurrencies.

The purpose of this work is to investigate insinuating solutions in the field of analysis of the dynamics of the cryptocurrency exchange rate, building machine learning models, data enrichment and improving the quality of cryptocurrency exchange rate forecasting with the help of the Twitter database, and the development of a convenient tool for analysis and planning of trades.

The object of the research is statistical data, analytical and financial reports, data of publications of the Twitter social network "Twitter User Data".

The subject of research are methods of tabular data analysis, regression models, hybrid recurrent neural networks.

The results obtained are a program for visualizing the dynamic characteristics of cryptocurrencies, a script for predicting the exchange rate of cryptocurrencies, which has high forecasting accuracy (estimated by the RMSE metric).

The software product was implemented using the Python programming language in the DataSpell 2022 development environment. The Jupyter Notebook environment was used to analyze and build models.

Results obtained: a simple interface was developed for the analysis of statistical and dynamic characteristics of coins, which includes forecasts for the future (for the selected number of steps).

## ЗМІСТ

|   |    |
|---|----|
| СПИСОК СКОРОЧЕНЬ .....  | 8  |
| ВСТУП .....   | 9  |
| РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ .....   | 10 |
| 1.1 Основні поняття GameFi маркету .....  | 10 |
| 1.2 Історія крипто-маркету та NFT-токенів.....  | 12 |
| 1.3 Продукти та функції DeFi .....  | 15 |
| 1.4 Існуючі рішення в області аналізу та надання прогнозів курсу<br>криптовалют .....   | 16 |
| 1.5 Постановка проблеми магістерської дипломної роботи.....   | 18 |
| 1.6 Висновки до першого розділу .....   | 19 |
| РОЗДІЛ 2 МЕТОДИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПРОГНОЗУВАННЯ<br>КУРСУ КРИПТОВАЛЮТ.....  | 20 |
| 2.1 Задача прогнозування.....   | 20 |
| 2.2 Специфіка даних в задачах прогнозування.....  | 21 |
| 2.3 Методи аналізу даних, особливості обробки .....   | 23 |
| 2.4.1 Методи прогнозування.....   | 25 |
| 2.4.2 Оцінки якості моделі прогнозування .....  | 31 |
| 2.4 Висновки до другого розділу.....  | 41 |
| РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ<br>ДАНИХ З ПОРТАЛУ BINANCE ТА ПОБУДОВИ АНАЛІТИЧНИХ<br>МОДЕЛЕЙ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ..... | 43 |
| 3.1 Вступ .....   | 43 |
| 3.2 Концептуальна блок-схема комп'ютерної програми та опис<br>модулів   | 43 |
| 3.3 Інструкція з експлуатації програми та приклад використання ....   | 45 |
| 3.3.1 Опис даних для дослідження .....  | 49 |

|   |     |
|---|-----|
| 3.3.2 Аналіз даних .....  | 51  |
| 3.3.3 Обробка даних .....   | 52  |
| 3.3.4 Побудова моделей прогнозування .....  | 52  |
| 3.4 Висновки до третього розділу .....  | 59  |
| РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....  | 60  |
| 4.1 План розробки стартапу та масштабування його на ринок.....  | 60  |
| 4.2 Опис ідеї стартап-проекту .....   | 62  |
| 4.3 Технологічний аудит ідеї проекту.....   | 63  |
| 4.4 Аналіз ринкових можливостей запуску стартап-проекту .....   | 67  |
| 4.5 Розроблення ринкової стратегії стартап-проекту.....   | 75  |
| 4.6 Розроблення маркетингової програми стартап-проекту.....   | 78  |
| 4.7 Висновки до розділу 4 .....   | 80  |
| ВИСНОВКИ .....  | 81  |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....   | 82  |
| ДОДАТОК А. Лістинг програмного модулю .....   | 89  |
| ДОДАТОК Б. Архітектура RNN, запропонована в програмі [66] .....   | 116 |
| ДОДАТОК В. Архітектура мережі для обробки текстових даних Twitter та визначення класу (позитивний/негативний твіт)..... | 117 |

## СПИСОК СКОРОЧЕНЬ

AIC – Akaike information criterion  
DAO – Decentralized Autonomous Organizations  
DeFi – Decentralized finance  
DT – Древа прийняття рішень  
DYOR – Do Your Own Research  
GRU – Gated recurrent units  
LR – Лінійна регресія  
LSTM – Long short-term memory  
MCC – Matthews correlation coefficient  
MFA – Measuring Forecast Accuracy  
MOP – Market Operating Price  
MPA – Merchant Agreement  
MLP – Багатошаровий перцептрон  
NB – Наївний Байєс  
NFT – Non-fungible tokens  
TVL – Total value locked  
P2E – Play to earn  
RF – Випадковий ліс  
SAT – Scholastic Aptitude Test  
SVM – Метод опорних векторів  
RNN – Recurrent neural network



## ВСТУП

Фінансова індустрія визнає трансформаційний вплив технології блокчейн для отримання нових доходів, підвищення ефективності процесів, покращення досвіду кінцевих користувачів і зниження ризиків у бізнес-операціях. Одним із напрямків, які активно розвиваються, є GameFi — комбінація слів game та finance, блокчейн-ігри типу «грай, щоб заробити», які пропонують гравцям економічні стимули. Екосистема GameFi використовує криптовалюти, незамінні токени (NFT) і технологію блокчейн для створення віртуального ігрового середовища [1].

Зазвичай гравці заробляють винагороди в грі, виконуючи завдання, борючись з іншими гравцями та просуваючись на різних рівнях гри. Вони також можуть передавати свої активи за межі гри для торгівлі на криптобіржах і ринках NFT [2].

Більшість блокчейн-ігор дозволяють гравцям переносити ігрові предмети з віртуального світу гри на відміну від традиційних відеоігор. Це дозволяє гравцям торгувати своїми предметами на ринках NFT, а криптовалютними доходами – на криптовалютних біржах.

Торги криптовалютою здебільшого проводяться зі спекулятивних причин, хоча вони також використовуються як інструмент інвестування, зберігання грошей та з інших причин. Новини криптовалютної торгівлі сильно залежать від нової урядової політики, важливих останніх подій, впровадження нових технологій і процедур, а також цитат експертів і впливових осіб у галузі блокчейн і фінтех.

Саме тому аналітичний застосунок з прогнозами дасть змогу простим користувачам проаналізувати криптовалюти, якими вони зацікавлені, побачити поточні тренди та прогнози на майбутнє курсу цих монет та прийняти рішення щодо покупки/продажу для отримання прибутку.

## РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Основні поняття GameFi маркету

Технологія блокчейн дозволяє володіти цифровими активами, тобто гравці монетизують свої внутрішньоігрові активи використовуючи різні способи.

Подібно до відеоігор, гравці можуть мати аватари, домашніх тварин, будинки, зброю, інструменти та ін, але в GameFi ці активи можна випускати або створювати як NFT у блокчейні. Це дає можливість гравцям мати повний контроль над власними активами з достовірністю та можливістю перевірки права власності [3-5].

Прикладом ігор метавсесвіту, зосереджених на концепції власності на землю, є Decentraland і The Sandbox. Вони дозволяють гравцям монетизувати свою віртуальну землю [6]. У The Sandbox гравці мають можливість купувати цифрові об'єкти нерухомості та розвивати їх щоб отримати прибуток. Наприклад, вони можуть стягувати плату з інших гравців за відвідування їхньої землі, отримувати символічні винагороди, розміщуючи вміст і події, або здавати в оренду свою налаштовану землю іншим гравцям.

У GameFi винагорода може надходити в різних формах, наприклад у криптовалютах або внутрішньоігрових активах, таких як віртуальна земля, аватари, зброя та костюми. Кожен проект GameFi матиме різну модель і економіку гри. У більшості випадків внутрішньоігрові активи — це NFT, якими можливо торгувати на ринках NFT. Однак в інших випадках внутрішньоігрові активи потрібно перетворити на NFT, перш ніж гравці зможуть ними торгувати або продавати [16].

Як правило, ігрові активи надають переваги гравцям, надаючи можливість отримувати більше винагород. Певні ігри також містять аватари та косметику, які є суто візуальними та не впливають на ігровий процес і заробіток.

Гравці отримують нагороди, виконуючи завдання, борючись з іншими гравцями або будуючи монетизовані споруди на своїй ділянці землі. Деякі ігри дають можливість гравцям отримувати пасивний дохід, тобто не приймати безпосередню участь у грі, або через ставки, або шляхом позичання своїх ігрових активів іншим гравцям.

Модель Play to Earn (P2E) – основа проектів GameFi. Це суттєво відрізняється від моделі оплати за гру, прийнятої в традиційних відеоіграх. Плата за гру вимагає від геймерів інвестування, перш ніж вони зможуть почати грати. Наприклад, для таких відеоігор, як Call of Duty, гравці потребують придбання ліцензій або регулярних підписок [7].

У більшості випадків традиційні відеоігри не принесуть жодних фінансових прибутків гравцям, а їхні внутрішньоігрові активи контролюються та зберігаються ігровою компанією. Навпаки, P2E-ігри можуть надати гравцям повний контроль над своїми внутрішньоігровими активами, а також пропонувати їм можливість заробляти гроші [8].

Однак треба зауважити, що все залежить від моделі та дизайну гри, прийнятих проектами GameFi. Технологія блокчейн може (і повинна) надати гравцям повний контроль і розпорядження своїми ігровими активами.

Важливим фактом є те, що P2E-ігри можуть бути безкоштовними та приносити гравцям фінансову винагороду, але деякі проекти GameFi вимагають придбання NFT або криптоактивів, перш ніж грати. Таким чином, DYOR завжди важливо оцінювати ризики. Якщо гра P2E потребує великих інвестицій, а винагорода невелика, то велика ймовірність втрати своїх початкових інвестицій [3].

Однією з найпопулярніших ігор типу «грай і заробляй» є Axie Infinity. Це NFT-гра на основі Ethereum, що почала набирати популярність з 2018 року. Виконуючи щоденні квести та борючись з іншими гравцями геймери можуть використовувати своїх улюбленців NFT (Axies), щоб заробляти токени SLP та надалі використовувати їх [14].

Крім купівлі та продажу Axies, гравці можуть давати в оренду своїх Axies іншим, що дозволяє власникам заробляти, не граючи. Ця модель позики має назву стипендія. Це дозволяє вченим використовувати запозичені Axies, щоб грати та отримувати нагороди [5].

Іншими словами, власники Ахіе можуть отримувати пасивний дохід, а вчені можуть грати в гру без жодних інвестицій. Потім отримані нагороди розподіляються між ученими та власниками Axies.

## 1.2 Історія крипто-маркету та NFT-токенів

Ідея створення цифрової валюти не нова. До появи криптовалюти було зроблено багато спроб її створення. Основною проблемою, з якою стикалися більшість із них, була проблема подвійних витрат. Цифровий актив якимось чином має бути придатним для використання лише один раз, щоб запобігти його копіюванню та ефективній підробці [18].

Понад 10 років до появи криптовалют цю концепцію представив комп'ютерний інженер Вей Дай. У 1998 році він опублікував статтю, де обговорював «В-money» [9]. Він обговорював ідею цифрової валюти, яку можна було б надіслати за групою цифрових псевдонімів, які неможливо відстежити. Того ж року нова спроба під назвою Bit Gold була розроблена піонером блокчейну Ніком Сабо. Bit Gold так само розглядав можливість створення децентралізованої цифрової валюти. Ідея Сабо була викликана неефективністю в традиційній фінансовій системі, як-от вимога металу для створення монет і зменшення кількості довіри, необхідної для створення транзакцій. Хоча обидва ніколи не були офіційно запуснені, вони були частиною натхнення для біткойна.

Сатоші Накамото опублікував книгу з назвою «Bitcoin: однорангова електронна готівкова система», в якій описується функціональність мережі

блокчейну Bitcoin. Цей день в історії біткойна проклав шлях для наступних подій [10].

Чотири місяці потому Сатоші Накамото, чия справжня особистість донині залишається таємницею, видобув перший блок мережі Bitcoin, фактично пілотуючи технологію блокчейн. Перший видобутий блок також відомий як блок Genesis.

Перша зареєстрована покупка товарів була здійснена за допомогою біткойнів, коли Ласло Ганец купив дві піци за 10 000 BTC. Цей день досі відзначається як День біткойн-піци [20].

Після народження біткойна як першої криптовалюти потрібно було знайти рішення для торгівлі ними. У березні 2010 року з'явилася перша біржа криптовалют під назвою bitcoinmarket.com (нині не існує). У липні того ж року також був запущений Mt.Gox.

З 2011 по 2013 рік біткойну вдалося досягти паритету з долларом США в лютому. Протягом цього року з'явилося кілька конкуруючих криптовалют: станом на травень 2013 року ринок криптовалют містив 10 цифрових активів, включаючи лайткойн. Ще один великий криптоактив приєднався в серпні під назвою XRP (Ripple) [11].

У міру того, як вартість біткойна зростала, пішли перші хаки. У червні 2011 року Mt.Gox вперше зламали: було вкрадено 2000 BTC, вартість яких на той момент становила близько 30 000 доларів.

У 2013 році Mt.Gox стала найбільшою криптовалютною біржею, на піку якої оброблялося 70% усіх транзакцій з біткойнами [21].

У 2014 році Mt.Gox став першою великою біржею криптовалют, з якої було вкрадено 850 000 BTC. Це найбільша крадіжка BTC в історії біткойна, яка на той час оцінювалася в 460 000 000 доларів США (поточна вартість близько 9,5 мільярдів доларів) [22].

Після цієї ситуації ціна біткойна впала на 50% і не відновилася до початкової вартості до кінця 2016 року. З тих пір хаки на біржах криптовалюти залишаються постійними, хоча рідко такого калібру, як Mt.Gox.

30 липня 2015 року була запущена мережа Ethereum. Наразі другий за ринковою капіталізацією криптоактив, він приніс смарт-контракти та, зрештою, децентралізував фінанси у світі криптовалют. Це дозволяє блокчейну Ethereum запускати всю екосистему на своєму блокчейні, а також розміщувати власну рідну валюту: ефір (ETH). Найменша одиниця ефіру також відома як Вей (0,000,000,000,000,000,001 ETH) [25].

Криптовалюти, які не мають власного виділеного блокчейну, але використовують блокчейн іншого криптоактиву, називаються токенами. Ті, які знаходяться в мережі Ethereum, називаються токенами ERC-20 [12].

Перший в історії токен ERC був запущений ще в 2015 році. Це був криптоактив, відомий як Augur. З того дня на блокчейні Ethereum було створено безліч токенів. Наразі існує понад 200 000 токенів ERC, що означає, що існує величезна екосистема криптовалют, що працює на одному блокчейні.

Для сторонніх людей ринок NFT може здатися складним. Використання технології блокчейну для невзаємозамінних токенів (NFT) може здатися дивною концепцією, але за останній рік інтерес до них зріс.

Це призвело до захоплюючої статистики та тенденцій NFT.

На початку 2021 року відбулося різке зростання щотижневих продажів цих цифрових активів онлайн, а також було побито кілька рекордів, і багато людей уважно дивилися на них, розмірковуючи про їхню цінність.

Невзаємозамінні токени (NFT) — це цифрові предмети колекціонування, які містять унікальні дані, живуть у блокчейні та не можуть бути скопійовані чи відтворені.

Це означає, що кожен NFT можна унікально ідентифікувати та може існувати лише один раз у світі.

Збирати NFT ніколи не було так просто з CryptoKitties, Etherbots, Etheremon GO, Gods Unchained тощо.

#### Ключова статистика NFT за 2022 рік [13]

- У 2021 році ринок NFT становив 41 мільярд доларів.
- «Злиття» є найдорожчим NFT, коли-небудь проданим — він пішов за 91,8 мільйона доларів.
- Щотижня в блокчейні продається NFT на 10-20 мільйонів доларів.
- Загальна вартість продажів NFT у 2020 році склала 250 мільйонів доларів.
- Перший NFT був створений у 2014 році.
- Понад 50% продажів NFT становлять менше 200 доларів.
- 250 000 людей щомісяця торгують NFT на OpenSea.

#### 1.3 Продукти та функції DeFi

Деякі проекти GameFi також пропонують продукти та функції DeFi, як от стейкинг, майнінг ліквідності та yield farming. Як правило, гравці можуть ставити свої ігрові жетони, щоб отримати нагороди, розблокувати ексклюзивні предмети або отримати доступ до нових ігрових рівнів.

Запровадження елементів DeFi сприяє децентралізації криптоігор. На відміну від звичних ігрових компаній, які централізували контроль над своїми оновленнями ігор, деякі проекти GameFi дозволяють спільноті брати участь у процесі прийняття рішень. Вони можуть пропонувати та голосувати за майбутні оновлення через децентралізовані автономні організації (DAO) [26].

Наприклад, гравці Decentraland можуть голосувати за політику в грі та організаційну політику, заблокувавши свої токени керування (MANA) у DAO. Чим більше жетонів вони заблокують, тим вище їх право голосу. Це дозволяє

геймерам спілкуватися безпосередньо з розробниками гри та впливати на розробку гри [14].

#### 1.4 Існуючі рішення в області аналізу та надання прогнозів курсу криптовалют

Слід зазначити, що на момент дослідження не існувало платформи, яка спеціалізувалась на аналізі криптовалют виключно GameFi сегменту. Незважаючи на це, існуючі платформи для аналізу часто включають в себе деякі монети цього сегменту. Щодо представлення спрогнозованих тенденцій – є декілька рішень, що мають платну підписку, але ці програми не включають в себе прогноз монет GameFi сегменту. Розглянемо детальніше найпопулярніші та найфункціональніші з них, проаналізуємо плюси та мінуси, та на основі цього зробимо висновок, які функціональні особливості будуть доречні у власному застосунку.

1. Glassnode — одна з найпопулярніших платформ криптовалютної аналітики. Складається з онлайн провідника даних з візуалізацією метрик у ланцюзі. Має бібліотеку розширених мережевих показників для широкого діапазону активів. Glassnode надає змістовну та багатообіцяючу поглиблену інформацію, звіти та аналіз криптовалюти на основі багатьох даних у мережі. Ця аналітична платформа також містить діаграми високої роздільної здатності, які частіше оновлюються, щоб аналітики могли правильно прогнозувати майбутнє.

2. Coin Metrics відома своїм найкращим мережевим продуктом даних про криптоактиви з CM Network Data Pro для кращого аналізу криптовалюти. Це канал даних із глибокими та зведеними мережевими показниками для всіх найпопулярніших криптовалют. Він має деякі функції для кращого аналізу — частота в реальному часі, частота наприкінці дня, понад 300 показників для понад 100 активів із повною історією з моменту створення мережі, легкий доступ через REST і WebSocket API із інструментами візуалізації мережевих



даних. Понад 300 метрик включають пропозицію, економіку, використання, майнер, новаторство та багато іншого, використовуючи дані ринку та мережі CoinMetrics, як-от MVRV. CoinMetrics надає такі продукти, як ATLAS, Network Data Pro, канал ринкових даних, довідкові ставки, а також FARUM.

3. Messari — ще одна платформа аналітики криптовалют, яка надає інвесторам надійні дані та ринкову аналітику, щоб отримати вищу рентабельність інвестицій за допомогою належного аналізу. Застосунок зосереджений на забезпеченні прозорості ринку криптовалют шляхом розробки інструментів криптовалютної аналітики для управління процесами прийняття рішень. Він діє як агрегатор криптоданих із фільтром для сортування даних про монети через кілька налаштованих фільтрів для кращого розуміння.

4. Santiment надає онлайн, соціальну та фундаментальну інформацію про понад 1500 криптовалют, доступних на світовому ринку. Ця криптовалютна аналітична платформа допомагає виявляти формування багатьох можливостей, а також визначати та контролювати дуже нестабільні підприємства. Є кілька корисних функцій для початківців — Sanbase Studio, сповіщення з низькою затримкою, персоналізовані списки спостереження для бажаних криптовалют, соціальні показники та скрінінги з унікальними ринковими показниками та індикаторами. Є також ексклюзивні ринкові звіти, аналітика розподілу власників, сигнали розбіжності ціни та DAA, а також спеціальні шаблони та альфа-версії. Він також пропонує два продукти, відомі як Sanbase і SanAPI для початківців, щоб краще зрозуміти складний ринок.

5. eToro — це інтегрована світова платформа аналітики криптовалютної соціальної торгівлі з більш ніж тисячами варіантів аналізу ринку для новачків. Платформа забезпечує безпечну роботу потенційних інвесторів з новими пропозиціями для різноманітного інвестиційного досвіду в будь-якому місці в будь-який час. Це платформа з кількома активами з 0% комісії та безкоштовною страховкою. Забезпечує доступ до понад 2000 різних активів,

включаючи криптовалюту. Понад 10 мільйонів інвесторів у 140 країнах пропонують варіанти коротко-, середньо- та довгострокових інвестицій.

Нижче, на рис.1.1 наведено табличний огляд того, як розвиватиметься криптовалют GameFi сегменту GAFI у короткостроковій перспективі (протягом наступних 90 днів) відповідно до моделі прогнозування за даними CaptainAltcoin's:

| Next 30 days | Next 60 days | Next 90 days    |
|--------------|--------------|-----------------|
| 2022-09-26   |              | +9.57 % \$ 9.92 |
| 2022-09-27   |              | +0.14 % \$ 9.06 |
| 2022-09-28   |              | +0.76 % \$ 9.12 |
| 2022-09-29   |              | +3.66 % \$ 9.38 |
| 2022-09-30   |              | -1.82 % \$ 8.89 |
| 2022-10-01   |              | +0.90 % \$ 9.13 |
| 2022-10-02   |              | -1.21 % \$ 8.94 |
| 2022-10-03   |              | +2.41 % \$ 9.27 |
| 2022-10-04   |              | +1.47 % \$ 9.18 |
| 2022-10-05   |              | +4.34 % \$ 9.44 |
| 2022-10-06   |              | -2.45 % \$ 8.83 |
| 2022-10-07   |              | -5.25 % \$ 8.58 |
| 2022-10-08   |              | -0.67 % \$ 8.99 |

Рис. 1.1 – Прогноз курсу GameFi протягом 90 днів

### 1.5 Постановка проблеми магістерської дипломної роботи

Як показало дослідження існуючих готових аналітичних платформ та тенденцій на ринку криптовалюти тема створення аналітичної платформи, що спирається на аналіз поточних тенденцій та прогноз майбутніх за допомогою моделей штучного інтелекту є актуальною та потребує глибокого аналізу та нових «розумних» рішень.

Для вирішення поставленої проблеми мають бути розв'язані наступні задачі:

- 1) огляд існуючих методів обробки табличних даних.

2) огляд існуючих методів для прогнозування курсу криптовалют.

4) додавання даних взятих з публікацій в соціальних мережах для перевірки гіпотези покращення точності прогнозування з використанням новин/обговорень в соціальних мережах.

5) прогнозування курсу криптовалют на різних часових проміжках, вибір та обґрунтування фіксованого кроку прогнозування, валідація моделей.

6) вибір найкращої моделі, апробація розробленої програми на реальних даних, розробка аналітичного застосунку з користувацьким інтерфейсом.

## 1.6 Висновки до першого розділу

Фінансова індустрія визнає трансформаційний вплив технології блокчейн для отримання нових доходів, підвищення ефективності процесів, покращення досвіду кінцевих користувачів і зниження ризиків у бізнес-операціях. Одним із напрямків, які активно розвиваються, є GameFi — комбінація слів game та finance, блокчейн-ігри типу «грай, щоб заробити», які заохочують гравців економічними стимулами. Екосистема GameFi використовує криптовалюту, незамінні токени (NFT) і технологію блокчейн для створення віртуального ігрового середовища.

Проаналізувавши методи аналізу та існуючі рішення, а також тенденції ринку та технологій, вирішила зосередитись на новому, перспективному і «не дуже дослідженому» напрямку GameFi. Технологія стрімко розвивається та приваблює геймерів, пропонуючи їм можливість заробляти гроші, розважаючись.

Аналітичний застосунок має включати в себе описову та діагностичну аналітику для прогнозування поточних і майбутніх тенденцій. Точне прогнозування курсу на короткочасну та довгострокову перспективу допоможе «розумно» вкладати кошти та купувати/продавати криптовалюту.

## РОЗДІЛ 2 МЕТОДИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

### 2.1 Задача прогнозування

Прогнозування часових рядів відбувається при роботі з науковими прогнозами на основі історичних даних із мітками часу. Це передбачає побудову моделей за допомогою історичного аналізу та їх використання для спостереження та прийняття майбутніх стратегічних рішень. Важливою відмінністю у прогнозуванні є те, що на момент виконання роботи майбутній результат абсолютно недоступний і може бути оцінений лише шляхом ретельного аналізу та попереднього аналізу, заснованого на доказах [19].

Прогнозування часових рядів – це застосування моделі для передбачування майбутніх значень на основі значень попередньо спостережених. І хоча регресійний аналіз часто застосовується для перевірки теорій чи поточні значення одного чи більше незалежних часових рядів впливають на поточне значення іншого часового ряду, цей тип аналізу часових рядів не називають «аналізом часових рядів», який натомість зосереджується на порівнянні значень одного часового ряду або багатьох залежних часових рядів у різні моменти часу. У деяких галузях прогнозування може стосуватися даних на певний майбутній момент часу, тоді як прогнозування стосується майбутніх даних у цілому. Прогнозування рядів часто використовується в поєднанні з аналізом часових рядів. Аналіз часових рядів включає розробку моделей для розуміння даних, щоб зрозуміти основні причини. Потім прогнозування робить наступний крок щодо того, що робити з цими знаннями та передбачуваними екстраполяціями того, що може статися в майбутньому.

## 2.2 Специфіка даних в задачах прогнозування

Прогнозування має застосування в багатьох галузях, де корисні оцінки майбутніх умов. Залежно від поля точність значно відрізняється. Якщо фактори, пов'язані з тим, що прогнозується, відомі та добре зрозумілі, а також існує значна кількість даних, які можна використати, цілком імовірно, що кінцеве значення буде близьким до прогнозу. Якщо це не так або якщо на реальний результат впливають прогнози, то надійність прогнозів може бути значно нижчою.

Зміна клімату та зростання цін на енергоносії призвели до використання Egain Forecasting для будівель [28]. Це намагається зменшити енергію, необхідну для опалення будівлі, таким чином зменшуючи викиди парникових газів. Прогнозування використовується для планування попиту споживачів у повсякденному бізнесі виробничих і розподільчих компаній [29].

Хоча правдивість прогнозів щодо фактичної прибутковості акцій заперечується через посилення на гіпотезу ефективного ринку, прогнозування широких економічних тенденцій є звичайним. Такий аналіз надають як некомерційні групи, так і приватні комерційні установи.

Прогнозування коливань валютних курсів зазвичай досягається за допомогою поєднання діаграми та фундаментального аналізу. Суттєва відмінність між аналізом діаграм і фундаментальним економічним аналізом полягає в тому, що чартисти вивчають лише цінову дію ринку, тоді як фундаменталісти намагаються шукати причини, що стоять за цією дією. Фінансові установи об'єднують докази, надані їхніми фундаментальними та чартистськими дослідниками, в одну замітку, щоб надати остаточний прогноз щодо відповідної валюти.

Прогнозування використовувалося також для передбачення розвитку конфліктних ситуацій. Прогнозисти проводять дослідження, які

використовують емпіричні результати для вимірювання ефективності певних моделей прогнозування. Проте дослідження показали, що немає великої різниці між точністю прогнозів експертів, обізнаних у конфліктній ситуації, і прогнозів осіб, які знали набагато менше [30].

Прогнозування можна застосовувати в таких сферах:

- Планування попиту клієнтів
- Економічне прогнозування
- Прогнозування землетрусів
- Знову прогнозувати
- Енергетичне прогнозування для інтеграції відновлюваної енергетики
- Фінансування проти ризику дефолту за допомогою кредитних рейтингів і кредитних балів
- Прогнозування землекористування
- Ефективність гравця та команди у спорті
- Політичне прогнозування
- Прогнозування продукції
- Прогнозування продажів
- Технологія прогнозування
- Телекомунікаційне прогнозування
- Транспортне планування та транспортне прогнозування
- Прогнозування погоди, прогнозування повеней і метеорологія

Прогнозування є однією з форм структурованого експертного опитування, яке використовується для прогнозування можливих майбутніх результатів, на яке все частіше покладаються вчені-екологи для інформування щодо планування збереження та управління природними ресурсами [17].

Методи прогнозування можна використовувати для отримання експертних знань або суджень щодо невизначених величин або подій у ймовірнісній формі. Відомо, що методи прогнозування раніше не використовувалися для визначення пріоритетності географічних територій для збереження, оцінки ймовірності призначення МОР або інформації для міжнародних екологічних переговорів. Вище згадане дослідження усуває цю прогалину в літературі та використовує методи прогнозування (передбачення на основі експертів), щоб виявити ймовірності окремої події, тобто ймовірність виникнення, для позначення МРА [31].

### 2.3 Методи аналізу даних, особливості обробки

Часовий ряд — це послідовні дані, які індексуються за часовими мітками. Загальні приклади включають прогнозування фондового ринку, прогнозування температури. У цих прикладах порядок даних важливий і корисний для прогнозування на майбутнє.

Основні операції з підготовки даних:

1) Скользящее окно з операцією Reduce - це стосується виконання будь-якої операції (наприклад, суми, середнього, стандартного відхилення) за допомогою ковзного вікна над усіма даними.

2) Функції з затримкою - загалом це техніка розробки функцій, за якої нові функції генеруються шляхом зсуву заданих даних на  $t-1$ ,  $t-2$ ,  $t-3$  тощо.

3) Автоковаріація - це відноситься до коваріації заданих даних із запізненими версіями самих себе.

4) Часткова автоковаріація - це відноситься до коваріації наданих даних із самими версіями з відставанням, але після видалення коваріації через інші менші затримки.

Приклад. Для обчислення часткової автокореляції між даними та їхньою версією із затримкою  $K$ -ий, ефект коваріації виключається через затримки  $K+1$ -ий,  $K+2$ -ий версії з затримкою  $K$ -ий.

5) Стаціонарність часових рядів - часовий ряд є стаціонарним, якщо середнє значення, стандартне відхилення та автоковаріація залишаються постійними. Це означає, що не повинно бути частих коливань, при використанні рухливого вікна для обчислення середнього/стандартного відхилення/автоковаріації.

6) Тенденції - поступове збільшення/зменшення значень даних з плином часу, починаючи з будь-якого моменту часу.

7) Сезонність - коли тенденція періодично повторюється час від часу.

8) Повторна вибірка - перетворення даних на більш високу чи нижчу частоту.

9) Білий шум - це стосується даних із нульовим середнім, але випадкових/незалежних за своєю природою та має 0 кореляції з іншими значеннями в ряді. Якщо часовий ряд є білим шумом - неможливо робити прогнози.

10) Випадковий процес - процес, який дає різні результати при застосуванні того самого рівняння, тобто випадковий процес.

11) Випадкове блукання - цей процес спрямований на прогнозування майбутніх значень за допомогою рівняння 2.1:

$$Y(t) = Y(t - 1) + e. \quad (2.1)$$

де  $e$  — білий шум, який є випадковим для кожного прогнозу, а отже, і випадковим блуканням.

12) Випадкова ходьба з Drift - це схоже на випадкову блукання з невеликою зміною. Тут значення передбачені за допомогою рівняння 2.2:

$$Y(t) = Y(t - 1) + D + e. \quad (2.2)$$

де  $D$  — постійна дрейфу, а  $e$  — член білого шуму.



13) Адитивні моделі - це моделі, які вважають, що дані були згенеровані за допомогою рівняння 2.3:

$$d = s + t + q . \quad (2.3)$$

де  $d$  – отримані дані,

$s$  – сезонність,

$t$  – тренд,

$q$  – залишки.

де залишки – це значення, що залишилися, коли тренд і сезонність були видалені з будь-якого часового ряду

14) Мультиплікативні моделі - ці моделі враховують, що дані були згенеровані за допомогою рівняння 2.4:

$$d = s \cdot t \cdot q . \quad (2.4)$$

де  $d$  – отримані дані,

$s$  – сезонність,

$t$  – тренд,

$q$  – залишки.

#### 2.4.1 Методи прогнозування

Лінійна регресія (LR) – це статистичний інструмент, який використовується для прогнозування майбутніх значень на основі минулих значень. Він зазвичай використовується як кількісний спосіб визначення базової тенденції та коли ціни надмірно розширені. Лінія тренду лінійної регресії використовує метод найменших квадратів для побудови прямої лінії через ціни, щоб мінімізувати відстань між цінами та результуючою лінією тренду. Цей індикатор лінійної регресії відображає значення лінії тренду для кожної точки даних [33].

Лінійна регресія моделює зв'язок між двома змінними шляхом підлаштування лінійного рівняння під дані спостереження. Одна змінна є пояснювальною, інша – залежною. До прикладу, модельєр може захотіти

зв'язати вагу людей із їхнім зростом за допомогою моделі лінійної регресії. Перш ніж намагатися адаптувати лінійну модель до спостережуваних даних, модельєр повинен спочатку визначити, чи існує зв'язок між цікавими змінними. Це не вказує на те, що одна змінна викликає іншу (наприклад, вищі бали SAT не спричиняють вищих оцінок коледжу), але що між двома змінними існує певний значний зв'язок. Діаграма розсіювання є корисним інструментом для визначення залежності між двома змінними. Якщо здається, що немає жодного зв'язку між запропонованими пояснювальними та залежними змінними (тобто діаграма розсіювання не вказує на зростання або зменшення тенденцій), тоді підгонка моделі лінійної регресії до даних, ймовірно, не забезпечить корисної моделі. Числовим показником залежності між двома змінними є коефіцієнт кореляції, який є варіється від -1 до 1, та вказує на міцність зв'язку спостережуваних даних для двох змінних [34].

Лінія лінійної регресії має рівняння виду, формула 2.5:

$$Y = a + bX. \quad (2.5)$$

де  $X$  — пояснювальна змінна,

$Y$  — залежна змінна.

Нахил лінії дорівнює  $b$ , а  $a$  — точка перетину (значення  $y$ , коли  $x = 0$ ).

Багатошарові перцептрони (MLP) - це штучна нейронна мережа, яка має 3 або більше шарів перцептронів [39]. Ці шари - один вхідний шар, 1 або більше прихованих шарів і один вихідний рівень перцептронів. Дані надходять в одному напрямку, тобто вперед, від вхідних шарів -> прихованих шарів -> вихідного рівня. Зворотне розповсюдження — це техніка, за якої багатошаровий перцептрон отримує зворотний зв'язок щодо помилки своїх результатів, а MLP відповідно коригує свої ваги, щоб робити точніші прогнози в майбутньому. MLP використовується в багатьох методах машинного навчання, таких як класифікація та регресія. Було показано, що вони дають

високоточні результати, зокрема, для проблем класифікації [21]. На рис. 2.1 [43] зображено структуру базового перцептрон.

Регресія – це контрольована техніка машинного навчання, яка наближає змінну безперервного значення. Зазвичай він використовується для прогнозування/прогнозування значень на основі значень незалежної змінної. Проблему регресії можна розглядати як наявність одного вихідного нейрона без функції активації.

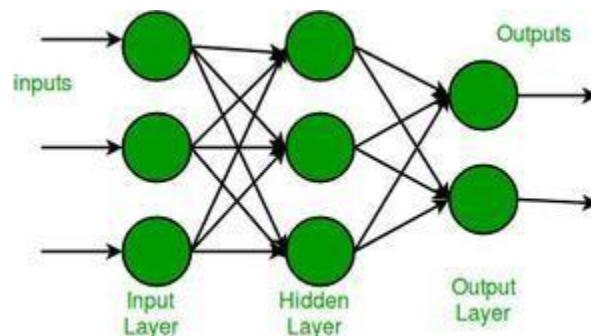


Рис.2.1 – Структура багатошарового перцептрон [43]

В таблиці 2.1 відображено основні відмінності між лінійною регресією та мультишаровим перцептроном:

Таблиця 2.1 – Відмінності між багатошаровим перцептроном та лінійною регресією

| MLP   | LR  |
|---|---|
| Багатошарове сприйняття — це мережа нейронів, які можна використовувати в бінарній/багатокласовій класифікації, а також у задачах регресії. | Модель лінійної регресії визначає лінійний зв'язок між залежною та незалежною змінними.<br>Вихідна функція може бути лінійною або неперервною. Це не має бути пряма лінія. Вихідна функція є лінійною і може бути представлена прямою лінією. |

## Продовження таблиці Таблиці 2.1

|   |   |
|---|---|
| MLP має кілька шарів нейронів із функцією активації та пороговим значенням.   | Модель лінійної регресії не має функції активації чи порогового значення.   |
| MLP зазвичай має кілька входів через 1 або більше вхідних нейронів.   | Проста лінійна регресія потребує лише одного введення – значення незалежної змінної – для прогнозування значення залежної змінної.  |
| MLP здебільшого використовуються для навчання під контролем, але в рідкісних прикладах, таких як самоорганізуюча карта Кохонена, використовувалися в навчанні без контролю для кластеризації. | Регресія – це метод навчання під наглядом.  |
| Сфери застосування MLP включають розпізнавання образів, автономне керування транспортним засобом, рекомендації соціальних медіа тощо.   | Сфери застосування лінійної регресії включають – прогнозування цін на житло, прогнозування продажів для бізнесу, прогнозування врожайності сільськогосподарських культур з урахуванням. опади тощо. |

Акронім моделі ARIMA розшифровується як «інтегроване ковзне середнє з авторегресією», складається з 3 умовних компонентів: AR, I та MA.

Компонент авторегресії — AR(p)

Компонент авторегресії моделі ARIMA представлений AR(p), причому параметр p визначає кількість серій із відставанням, які використовуються, формула 2.6:

$$y_t = c + \sum_{n=1}^p a_n y_{t-n} + c_t, \quad (2.6)$$

AR(0): білий шум

Якщо встановити параметр  $p$  рівним нулю (AR(0)), без членів авторегресії. Цей часовий ряд є просто білим шумом. Кожна точка даних вибирається з розподілу із середнім значенням 0 і дисперсією сигма-квадрат. Це призводить до послідовності випадкових чисел, які неможливо передбачити. Це дуже корисно, оскільки може служити нульовою гіпотезою та захистити наші аналізи від прийняття хибнопозитивних моделей [44].

AR(1): Випадкові блукання та коливання

Якщо для параметра  $p$  встановлено значення 1, враховується попередня позначка часу, скориговану множником, а потім додається білий шум. Якщо множник дорівнює 0, отримується білий шум, а якщо множник дорівнює 1 - випадкове блукання. Якщо множник знаходиться в межах  $0 < \alpha_1 < 1$ , тоді часовий ряд демонструватиме реверсію середнього значення. Це означає, що значення, як правило, коливаються навколо 0 і повертаються до середнього після регресії від нього.

AR(p): терміни вищого порядку

Ще більше збільшення параметра  $p$  означає лише повернення назад і додавання додаткових часових позначок, скоригованих їхніми власними множниками. Можливо повертатися як завгодно далеко назад, але при поверненні в кінці, більш імовірно, що нам слід використовувати додаткові параметри, такі як ковзне середнє (MA(q)) [46].

Ковзне середнє — MA(q)

«Цей компонент — не ковзне середнє, а скоріше відставання в білому шумі» [16]

MA(q) — це модель ковзного середнього, а q — кількість умов помилки прогнозування з відставанням у прогнозі. У моделі MA(1) наш прогноз є постійним членом плюс попередній член білого шуму, помножений на множник, доданий до поточного члена білого шуму. Це лише проста ймовірність + статистика, оскільки коригування прогнозу проходить на основі попередніх умов білого шуму [48].

#### Моделі ARMA та ARIMA

Архітектури ARMA та ARIMA — це просто компоненти AR (авторегресія) і MA (ковзне середнє), разом узяті.

#### ARMA

Модель ARMA є константою плюс сума лагів AR та їх множників, плюс сума лагів MA та їх множників плюс білий шум. Це рівняння лежить в основі всіх наступних моделей і є основою для багатьох моделей прогнозування в різних областях.

#### ARIMA

$$Y_t = \beta_2 + \varpi_1 \varepsilon_{t-1} + \varpi_2 \varepsilon_{t-2} + \dots + \varepsilon_t, \quad (2.7)$$

Модель ARIMA є моделлю ARMA, але з етапом попередньої обробки включено в модель, яка представлена за допомогою I(d). I(d) — порядок різниці, тобто кількість перетворень, необхідних для того, щоб дані стали стаціонарними. Таким чином, модель ARIMA – це просто модель ARMA на різницевих часових рядах. Розрахунок значення Y приведено у формулі 2.7.

#### Моделі SARIMA, ARIMAX, SARIMAX

Модель ARIMA працює добре, але включення в неї сезонності та екзогенних змінних може бути надзвичайно потужним. Оскільки модель ARIMA передбачає, що часовий ряд є стаціонарним, нам потрібно використовувати іншу модель.

#### SARIMA

$$y_t = c + \sum_{n=1}^p a_n y_{t-n} + \sum_{n=1}^q \theta_n \varepsilon_{t-n} + \sum_{n=1}^P \phi_n y_{t-sn} + \sum_{n=1}^Q \eta_n \varepsilon_{t-sn} + \varepsilon_t, \quad (2.8)$$

SARIMA (сезонна ARIMA). Ця модель дуже схожа на модель ARIMA, за винятком того, що є додатковий набір компонентів авторегресії та ковзного середнього, формула 2.8. Додаткові лаги компенсуються частотою сезонності (наприклад, 12 — щомісяця, 24 — щогодини).

Моделі SARIMA дозволяють розрізнити дані за сезонною частотою, а також за несезонними розходженнями. Знати, які параметри найкращі, можна легше за допомогою фреймворків автоматичного пошуку параметрів, таких як `sktime`.

### ARIMAX і SARIMAX

$$d_t = c + \sum_{n=1}^p a_n d_{t-n} + \sum_{n=1}^q \theta_n \epsilon_{t-n} + \sum_{n=1}^P \varphi_n y_{t-sn} + \sum_{n=1}^Q \mu_n \epsilon_{t-sn} + \epsilon_t, \quad (2.9)$$

Вище наведена формула 2.9 моделі SARIMAX. Ця модель враховує екзогенні змінні, іншими словами, використовує зовнішні дані в нашому прогнозі. Деякі реальні приклади таких змінних описують ціну на золото, нафту, температуру повітря, курс валют.

#### 2.4.2 Оцінки якості моделі прогнозування

Середня абсолютна похибка (MAE) і середньоквадратична похибка (RMSE) є двома найпоширенішими показниками, які використовуються для вимірювання точності безперервних змінних.

Середня абсолютна помилка (MAE): MAE оцінює середню величину помилок у векторі прогнозів без урахування знаку, формула 2.10. Це середнє значення абсолютної різниці між прогнозом і фактичним значенням, де кожна різниця має рівну вагу [50].

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (2.10)$$

Якщо абсолютне значення не береться (знаки похибок не видаляються), середня похибка стає середньою похибкою зсуву (MBE) і зазвичай призначена

для вимірювання середнього зміщення моделі. МВЕ може передавати корисну інформацію, але його слід тлумачити обережно, оскільки позитивні та негативні помилки скасовуються.

Середньоквадратична помилка (RMSE): RMSE – це квадратний корінь із середнього квадрата відмінностей між прогнозом і фактичним спостереженням, формула 2.11:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (2.11)$$

Порівняння метрик:

Подібності: і MAE, і RMSE виражають середню помилку прогнозування моделі в одиницях змінної, що цікавить. Ці дві метрики можуть варіюватися від 0 до  $\infty$  і не чутливі до напрямку помилок [53].

Відмінності: витяг квадратного кореня із середніх квадратичних помилок має деякі цікаві наслідки для RMSE. Оскільки помилки зведені в квадрат перед усередненням, RMSE надає відносно високу вагу великим помилкам. Це означає, що RMSE має бути більш точним, за умови, якщо великі помилки є небажаними.

Ще один наслідок формули RMSE, який не часто обговорюється, пов'язаний з розміром вибірки. Використовуючи MAE, можливо встановити нижню та верхню межу RMSE.

$[MAE] \leq [RMSE]$ . Результат RMSE завжди буде більшим або рівним MAE. Якщо всі помилки мають однакову величину, то  $RMSE=MAE$ .

$[RMSE] \leq [MAE * \sqrt{n}]$ , де  $n$  – кількість тестових зразків. Різниця між RMSE і MAE найбільша, коли вся помилка прогнозування походить від однієї тестової вибірки. Тоді квадрат помилки дорівнює  $[MAE^2 * n]$  для цього окремого тестового зразка та 0 для всіх інших зразків. Беручи квадратний корінь, RMSE тоді дорівнює  $[MAE * \sqrt{n}]$  [54].



Зосереджуючись на верхній межі, це означає, що RMSE має тенденцію бути все більшим, ніж MAE, із збільшенням розміру тестової вибірки.

Це може викликати проблеми під час порівняння результатів RMSE, розрахованих на тестових зразках різного розміру, що часто має місце в моделюванні реального світу.

RMSE має перевагу штрафувати великі помилки, тому може бути більш доцільним у деяких випадках, наприклад, якщо відхилення на 10 більш ніж удвічі гірше, ніж відхилення на 5, але якщо відхилення на 10 лише вдвічі гірше, ніж якщо вимкнено на 5, тоді MAE є більш доцільним.

Інформаційний критерій Akaike (AIC) — це оцінка похибки прогнозування і відносної якості статистичних моделей для наданого набору даних. Маючи набір моделей для даних, AIC оцінює якість кожної з моделей відносно кожної іншої. Отже, AIC забезпечує засіб вибору моделі [56].

AIC базується на теорії інформації. Статистична модель використовується задля представлення процесу, що створив дані. AIC оцінює відносну кількість інформації, втрачену моделлю: чим менше інформації модель втрачає, тим вища якість цієї моделі, формула 2.12:

$$AIC = -2 \ln(L) + 2 \quad (2.12)$$

де  $L$  - ймовірність і  $k$  - кількість параметрів.

Оцінюючи обсяг інформації, втраченої моделлю, AIC має справу з компромісом між придатністю моделі та простотою моделі. Іншими словами, AIC розглядає як ризик переобладнання, так і ризик недообладнання.

Інформаційний критерій Акайке названий на честь японського статистика Хіротугу Акайке, який його сформулював. Нині він є основою парадигми для основ статистики, а також широко використовується для статистичних висновків.

AIC використовує оцінку максимальної правдоподібності моделі

(логарифм правдоподібності) як міру відповідності. Логарифм правдоподібності є показником того, наскільки ймовірно хтось побачить свої спостережувані дані на основі моделі. Модель з максимальною ймовірністю – це та, яка найкраще «підходить» до даних [58]. Натуральний логарифм правдоподібності використовується для зручності обчислень. Щоб отримати додаткові відомості про логарифм правдоподібності, ось корисне (і трохи дивне) вступне відео для оцінки максимальної правдоподібності та інше відео про те, як оцінка максимальної правдоподібності застосовується до логістичної регресії, щоб отримати більш інтуїтивне відчуття того, що максимізує журнал моделі -імовірність виглядає так.

AIC є низьким для моделей із високим логарифмом правдоподібності (модель краще відповідає даним, чого ми й хочемо), але додає штрафний термін для моделей із вищою складністю параметрів, оскільки більше параметрів означає, що модель, швидше за все, перестане відповідати навчальні дані. На рис. 2.2 відображено графічно вплив логарифмування на штрафна складність параметрів – відповідно навчання моделі:

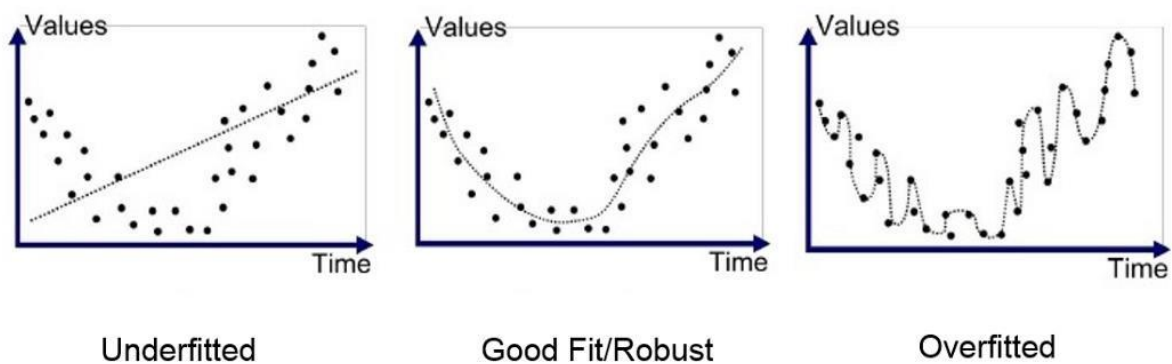


Рис.2.2 – Вплив підбору параметрів на якість моделі [59]

AIC працює, вважаючи наступні припущення виконаними:

1. Використовуються однакові дані в обох моделях
2. Вимірюється однакова змінна результату між моделями
3. Мається вибірка нескінченного розміру

Це останнє припущення пояснюється тим, що АІС збігається до правильної відповіді з нескінченним розміром вибірки. Часто велика вибірка є достатньою для приблизної оцінки, але оскільки використання АІС часто означає, що у вас невеликий розмір вибірки, існує формула, скоригована за розміром вибірки, яка називається АІСс, яка додає коригуючий термін, який збігається з відповіддю АІС для великих вибірок, але дає більш точну відповідь для менших зразків.

Як правило, завжди використовувати АІСс є найбезпечнішим, але АІСс особливо слід використовувати, коли співвідношення ваших точок даних ( $n$ ) : кількість параметрів ( $k$ )  $< 40$  [60].

Коли припущення АІС (або АІСс) виконано, найбільшою перевагою використання АІС/АІСс є те, що ваші моделі не потрібно вкладати, щоб аналіз був дійсним, на відміну від інших одночислових вимірювань відповідності моделі, як ймовірність -тест співвідношення. Вкладена модель — це модель, параметри якої є підмножиною параметрів іншої моделі. Як наслідок, абсолютно різні моделі можна математично порівнювати з АІС.

Припустимо, є розрахований показник АІС для кількох моделей і є ряд балів АІС ( $AIC_1, AIC_2, \dots, AIC_n$ ). Для будь-якого заданого  $AIC_i$  обчислюється ймовірність того, що « $i$ th» модель мінімізує втрату інформації за допомогою наведеної нижче формули, де  $AIC_{min}$  — найнижча оцінка АІС у серії оцінок.

$$P = \exp\left(\frac{AIC_{min} - AIC_i}{2}\right) \quad (2.13)$$

АІС вимірює лише відносну якість моделей. Це означає, що всі протестовані моделі все ще можуть погано підходити. У результаті необхідні інші заходи, щоб показати, що результати вашої моделі відповідають прийнятному абсолютному стандарту (наприклад, обчислення MAPE).

АІС також є відносно простим розрахунком, який був побудований на

інших більш складних з обчислювальної точки зору — але також зазвичай більш точних — узагальнених показниках. Приклади таких включають DIC (критерій інформації про відхилення), WAIC (критерій інформації Ватанабе-Акаїке) і LOO-CV (перехресна перевірка Leave-One-Out, до якої AIC асимптотично підходить із великими вибірками) [58].

Залежно від того, наскільки важлива точність порівняно з обчислювальним навантаженням (і зручність обчислень, враховуючи можливості вашого програмного пакета), можливо вибрати AIC або один із нових, складніших обчислень.

В таблиці 2.2 зібрано основні порівняльні характеристики метрик описаних вище:

Таблиця 2.2 – Метрики для оцінки якості регресійної моделі

| Метрика | Значення                   | Інтерпретація  | Обмеження  |
|---------|----------------------------|--|--|
| MAE     | $0 - \text{inf}$           | Чим нижче, тим менше помилок, тим краще модель           | Використання лише для порівняння на тому самому набору даних |
| RMSE    | $0 - \text{inf}$           | Чим нижче, тим менше штрафних помилок, тим краще модель  | Використання лише для порівняння на тому самому набору даних |
| AIC     | $-\text{inf} - \text{inf}$ | Чим менше, тим менше втрати інформації, тим краще модель | Використання лише для порівняння на тому самому набору даних |

Середня точність прогнозу (MFA - Measuring Forecast Accuracy) також є цікавим показником. Цей показник дуже інтуїтивно зрозумілий і його легко пояснити менеджеру (наша модель має середню точність прогнозування 66%, що також означає, що середня помилка прогнозування наших моделей становить 34%). Це дає гарне наближення того, наскільки добре працює модель прогнозування [61].

RNN — це кероване глибоке навчання, яке використовується для аналізу часових рядів. Рекурентні нейронні мережі представляють один із найдосконаліших алгоритмів, які існують у світі керованого глибокого навчання. RNN схожі на короткочасну пам'ять. Вище згадані моделі можуть запам'ятовувати речі, які щойно відбулися під час попередніх спостережень, і застосовувати ці знання в майбутньому. Для людини короткочасна пам'ять є однією з функцій лобової частки.

Ідея полягає в тому, що ваги мають довгострокову пам'ять, скорочену як LTM. Наприклад, у класичній ШНМ відомі ваги, тому будь-які вхідні дані, які вводяться в ШНМ, вона оброблятиме їх так само, як і вчора. Ваги можуть бути розташовані в скроневій частці людського мозку, оскільки скронева частка відповідає за довгострокову пам'ять LTM [62].

RNN схожа на короткочасну пам'ять, оскільки вона може запам'ятовувати речі, які щойно відбулися під час попередніх кількох спостережень. Наприклад, коли є багато тексту, потрібно оцінити, чи є це коментар негативним чи позитивним, цей RNN називається багато до одного.

Іншим прикладом є Google Translator, який є дуже складним глибоким навчанням. Наприклад, він може адаптувати переклад залежно від статі. Наприклад, англійською можна сказати: я хлопець/дівчинка і люблю читати, але переклад італійською для хлопчика такий: Sono un ragazzo e amo leggere. Переклад для дівчини натомість: Sono una ragazza e amo leggere. Цей RNN називається Багато до багатьох.

Задача градієнтного спуску: знайти глобальний мінімум, щоб отримати оптимальне рішення [63]. У RNN, коли інформація передається мережею, виникає багато помилок, які потрібно оцінити, і важливо пам'ятати, що сині вузли — це не просто нейрони, а повний прихований шар. Тепер кожен окремий нейрон, який бере участь у обчисленні результату, повинен мати оновлену вагу, щоб мінімізувати помилку. Weight Recurring  $W_{rec}$  у часовій петлі потрібно оновлювати багато разів, і вони стають малими.

При перемноженні ваг, значення швидко зменшується, а  $W_{rec}$  стає меншим, структуру відображено на рис.2.3 [64]:

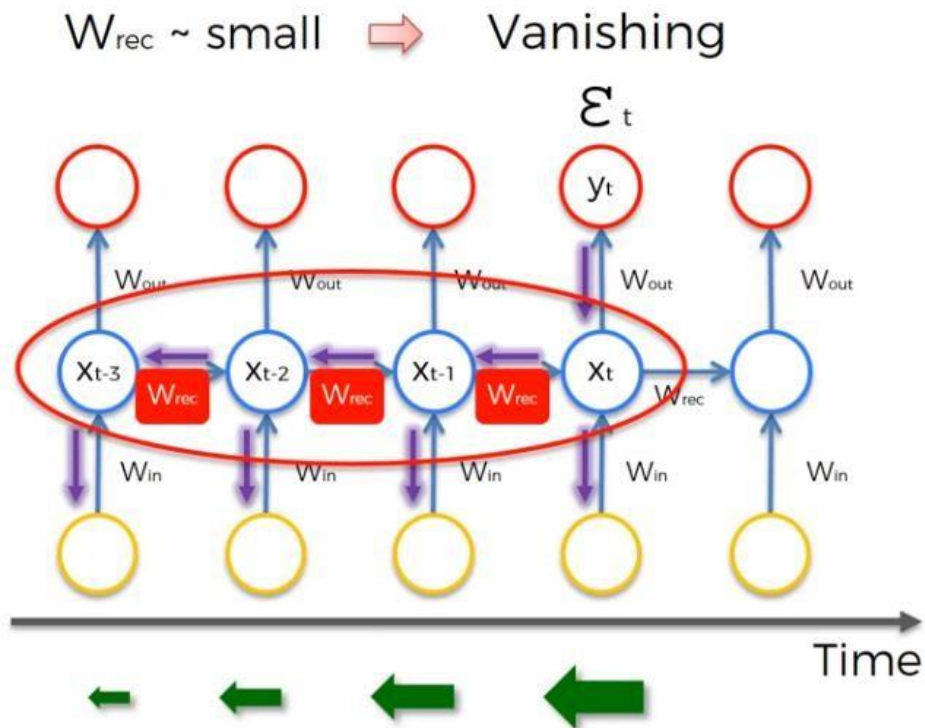


Рис. 2.3 – Передача ваг в мережі [64]

Як описано зеленими стрілками внизу малюнка, коли градієнт повертається через мережу, тим нижчим є градієнт. Наслідком цього є те, що чим нижчий градієнт, тим повільніше відбувається процес оновлення ваг, як ефект доміно. Рішенням є мережа довготривалої короткочасної пам'яті LSTM.

Мережі LSTM з довгою короткочасною пам'яттю

Найшвидше рішення розв'язати зникаючий градієнт – надати значення

Weight Recurring  $W_{rec}$  менше одиниці:  $W_{rec} < 1$  або  $W_{rec} = 1$ .

На рис. 2.4 [65] зображено структуру LSTM:

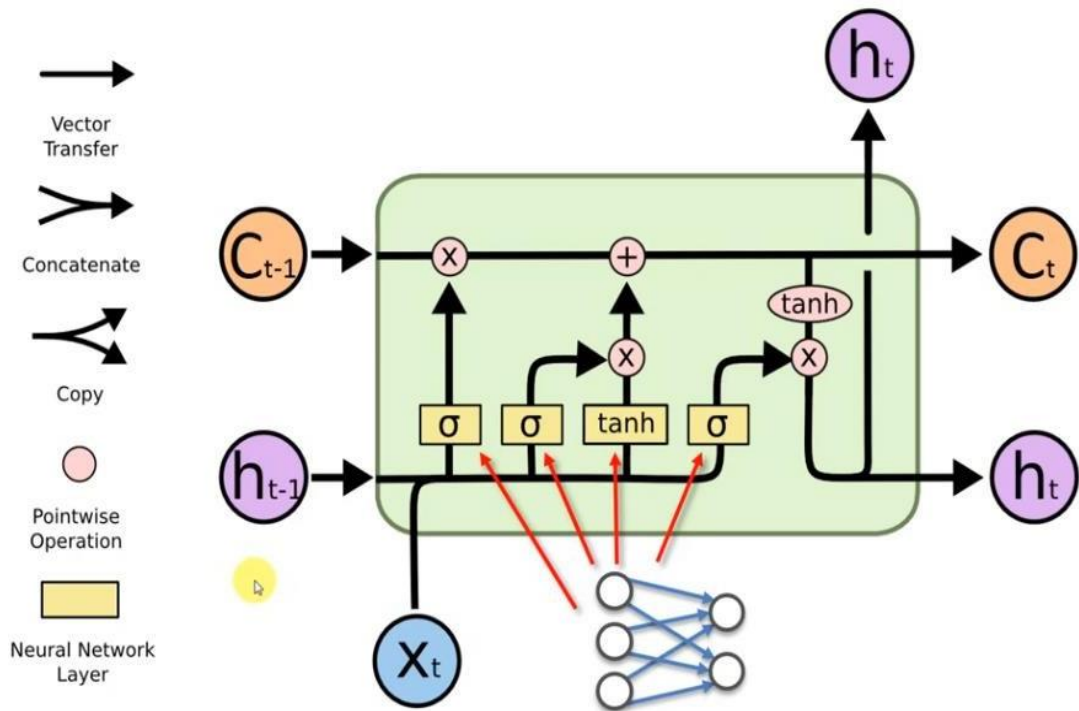


Рис. 2.4 – Архітектура LSTM [65]

$C$  означає пам'ять клітини.  $h$  — вихід, а  $X$  — вхід. Будь-яка чорна лінія на схемі є вектором. Операторами  $\times$  або  $\oplus$ , і, що може відкриватись або закриватись, і коли клапан відкритий, пам'ять передається вільно, і пам'яттю керує сигмоїдна функція, представлена символом сигма на графіку вище. Наприклад, відкриття клапана може допомогти виділити статтю у Google Translator. Іншими словами, це найкраще рішення для коректного перекладу тексту.

Символ  $+$  позначає з'єднання T-Shape і використовується для додавання додаткової пам'яті в потік.

Функція активації гіперболічного тангенса для короткого оператора  $\tanh$  коливається від  $-1$  до  $1$ , враховуючи збільшення та зменшення стану.

Існують також деякі варіанти LSTM, наприклад, так званий Gated Recurring Unit або скорочено GRU. Тут комірка пам'яті  $C$  повністю

ігнорується, і її замінює вихід  $h$ .

Тут новий прихований стан у попередньому прихованому стані об'єднується для формування вектора. Далі відбувається через функцію активації  $\tanh$ , функцію гіперболічної дотичної активації. Це допомагає регулювати потік через мережу та гарантує, що значення залишаються між  $-1$  і  $1$ . Завдяки  $\tanh$  уся інформація зберігається, замість цього, використовуючи функцію активації Sigmoid, можливо втратити інформацію, оскільки малі значення наближаються до нуля, тому вони зникають.

Рекурентні одиниці GRU мають шлюз оновлення та шлюз скидання. Використовуючи ці два вектори, модель уточнює результати, керуючи потоком інформації через модель. Вороти виражаються сигмоїдною функцією:  $0$  = не оновлювати вагу,  $1$  = оновлювати вагу.

#### BI-LSTM (двонаправлена довгострокова пам'ять)

Двонаправлена довго-короткочасна пам'ять (bi-lstm) — це процес створення будь-якою нейронною мережею інформації про послідовність в обох напрямках назад (від майбутнього до минулого) або вперед (від минулого до майбутнього), структура відображена на рис. 2.5

У двонаправленому режимі вхідні дані проходять у двох напрямках, що робить bi-lstm відмінним від звичайного LSTM. За допомогою звичайного LSTM можливо виконати вхідний потік в одному напрямку, або назад або вперед. Однак у двонаправленому режимі вхідний потік може бути виконаний в обох напрямках, щоб зберегти інформацію про майбутнє та минуле.

У реченні «хлопці йдуть до .....» неможливо заповнити порожній простір. Тим не менш, коли у нас є майбутнє речення «хлопці виходять зі школи», легко спрогнозувати минулий порожній простір, подібне, до того що потрібно виконати за допомогою нашої моделі, і двонаправлений LSTM дозволяє нейронній мережі виконати це.



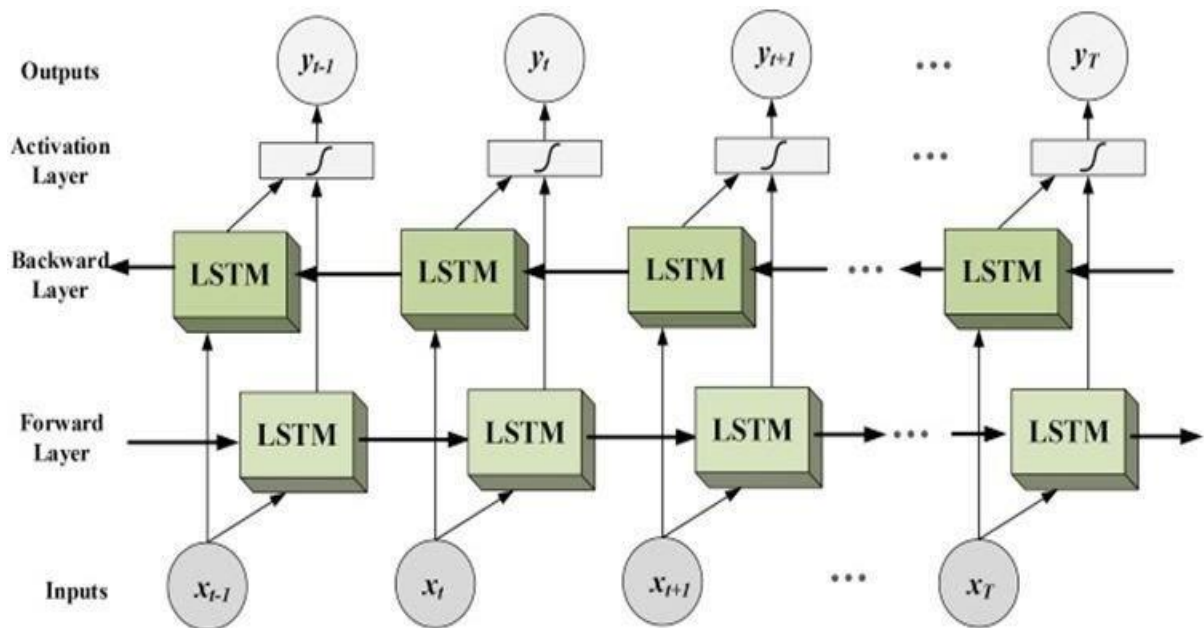


Рис. 2.5 – Архітектура Bi-LSTM моделі [61]

На діаграмі зображено побачити потік інформації від зворотного та прямого рівнів. BI-LSTM зазвичай використовується там, де потрібні послідовні завдання. Цей вид мережі можна використовувати в моделях класифікації тексту, розпізнавання мовлення та прогнозування.

## 2.4 Висновки до другого розділу

Великі дані є розповсюдженими і цінними в практичних задачах класифікації та регресії, які в подальшому вдосконалюють процеси прийняття рішень в компаніях. Зберігання та аналіз великих даних є провідним інструментом для світової економіки. Технології обробки даних дають можливість можливість зручної організації, оптимального зберігання, аналізу та дослідження даних з подальшим покращенням бізнес-процесів в фінансових компаніях [45].

Для отримання моделей з високою точністю прогнозування безперервних величин використано такі методи машинного навчання: ARIMA/SARIMA, LR, MLP, RNN, LSTM, та їх модифікаціями.

В розділі подано опис технік обробки даних, методів прогнозування безперервної величини (методи регресії) та оцінок якості моделей. В подальшому планується реалізація описаних методів обробки табличних даних та побудови моделей для вирішення задачі фінансового аналізу з метою прогнозування курсу криптовалют для застосування даного інструменту в торгівлі.

## РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДАНИХ З ПОРТАЛУ BINANCE ТА ПОБУДОВИ АНАЛІТИЧНИХ МОДЕЛЕЙ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

### 3.1 Вступ

У цьому розділі розміщується опис розробленої в рамках магістрської дипломної роботи комп'ютерної програми для візуалізації прогнозу та реальних цін на ринку криптовалют. Програма призначена для використання в власних цілях щодо оцінки курсу криптовалюти та оцінки реального значення та прогнозу в момент часу. Дана програма реалізована мовою програмування python в середовищі розробки DataSpell 2022.

Розроблений програмний продукт дає змогу користувачам з різним рівнем підготовки використовувати моделі штучного інтелекту для оцінки прогнозу курсу криптовалют через деякий момент часу, приймати рішення щодо покупки/продажу, одержувати статистичні характеристики й спрогнозовані значення на основі побудованої моделі.

Інтерфейс користувача є інтуїтивно зрозумілим і призначений для того, щоб розуміти ситуацію з курсом на ринку та передбачувану ціну однією з розроблених моделей. Технічний рівень застосунку – настільний програмний продукт, що може бути використаний одним користувацьким комп'ютером. Система не призначена для мережевої роботи.

### 3.2 Концептуальна блок-схема комп'ютерної програми та опис модулів

У цьому розділі розглядається схема роботи застосунку та опис програмних модулів. На рис. 3.1 наведена структура розробленого застосунку. Є 4 основні блоки: блок інтерфейсу, блок зберігання даних, основний операційний блок та блок виведення результатів.

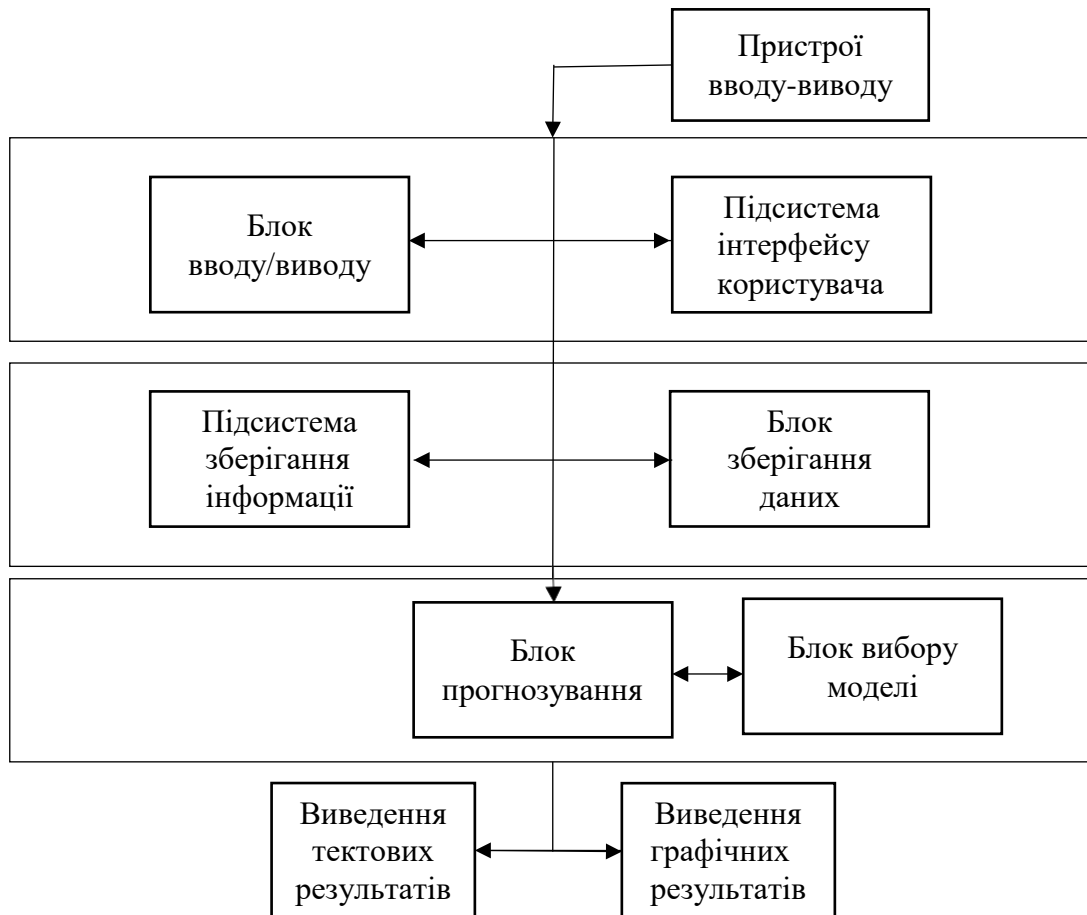


Рис. 3.1 – Структура системи підтримки прийняття рішень

Пристрої вводу-виводу забезпечують можливість користувачу завантажувати дані до програми та фільтрувати криптовалюту. Для цього підсистема вводу-виводу зв'язана з підсистемою інтерфейсу користувача. Підсистема інтерфейсу користувача призначена для здійснення зв'язку між користувачем і внутрішніми елементами системи та забезпечує ввід та вивід інформації. Інтерфейс дає змогу оператору ввести інформацію, дані, фільтрувати параметри і запити в систему та отримати вихідну інформацію в зручному для сприйняття вигляді. Підсистема зберігання інформації складається з бази даних, що призначена для зберігання табличних історизованих даних, з метою їх подальшого аналізу, обробки та використання. Підсистема обробки інформації призначена для перевірки даних, їх нормалізації, розбиття на тренувальний та тестовий набори. Блок прогнозування відповідає за видачу прогнозів на отримані дані. Підсистема

оцінювання якості моделі обчислює статистичні показники побудованої моделі, які характеризують прогнозуючу здатність моделі. Підсистема виведення результатів складається з графіків, таблиць для прийняття рішення користувачем щодо покупки/продажі. Представлення результатів прогнозування та критеріїв оцінювання моделі дозволяє зробити висновки щодо можливості використання отриманої моделі для прогнозування. Для створення програми були використані мова програмування python3.8, а саме модуль для машинного навчання sklearn, tensorflow та середовище розробки DataSpell 2022 для розробки застосунку та Jupyter Notebook від Anaconda для аналізу даних. Були використані як готові бібліотеки, так і написаний власний модуль для повного циклу роботи з даними: отримання, зберігання, обробка, визначення періодів, моделювання, прогнозування та збереження результатів.

Для роботи програмного продукту необхідна наявність персонального комп'ютера з наступними мінімальними характеристиками:

- а) операційна система Windows 7/10;
- б) тактова частота процесора 1 ГГц;
- в) оперативна пам'ять розміром 512 Мбайт;
- г) вільний дисковий простір: 5 Мбайт для розміщення виконавчого файлу, вхідних даних і результатів роботи;
- д) клавіатура та комп'ютерна мишка;
- е) монітор з мінімальною розподільчою здатністю 1024×768.

### 3.3 Інструкція з експлуатації програми та приклад використання

Робота з усіма елементами інтерфейсу є стандартною для програмного забезпечення, яку працює на платформі операційної системи MS Windows та Mac OS. Основний робочий екран ПП має структуру наведену на рис. 3.2.

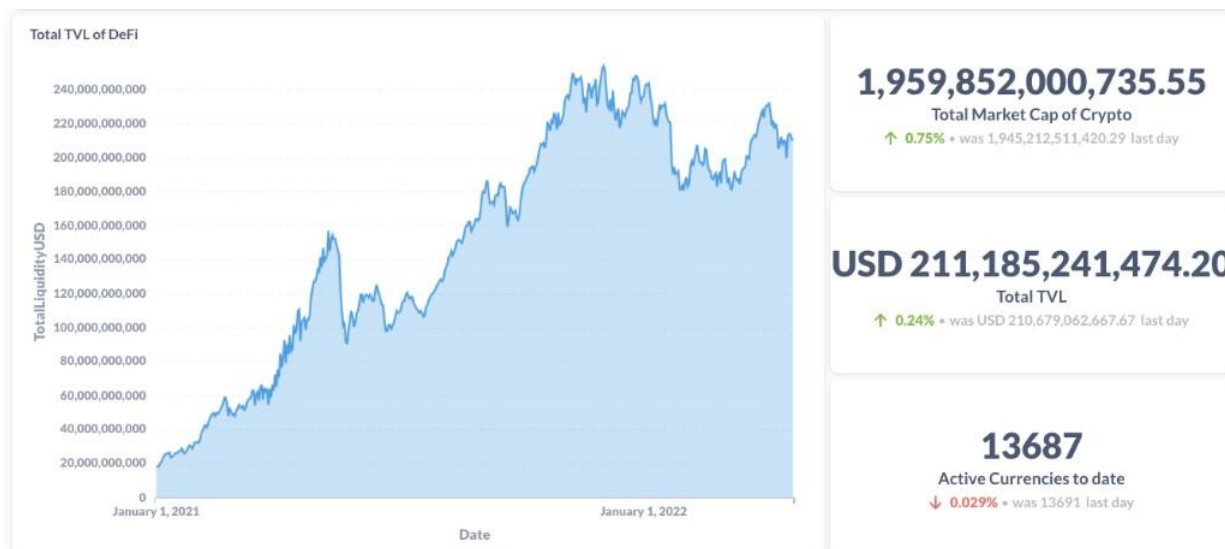


Рис. 3.2 – Головне вікно програми

Головне вікно програми призначене для перегляду основних статистик по монетам та оцінки динамічних показників (аналітичне вікно), де можна обрати цікаву користувачу монету та отримати інформацію.

Головну форму програми можна інтуїтивно розділити на 2 частини:

- Робоча область програми, в якій відображається графік історизації курсу монети до долару.
- Блок статистичних значень.

Натискаючи на фільтр, можна вибрати зі списку монету, яка цікавить.

В залежності від обраного типу криптовалют програма буде показувати агреговані показники. Якщо ввімкнений фільтр типу криптовалют, то вікно матиме вигляд як на рис. 3.3:

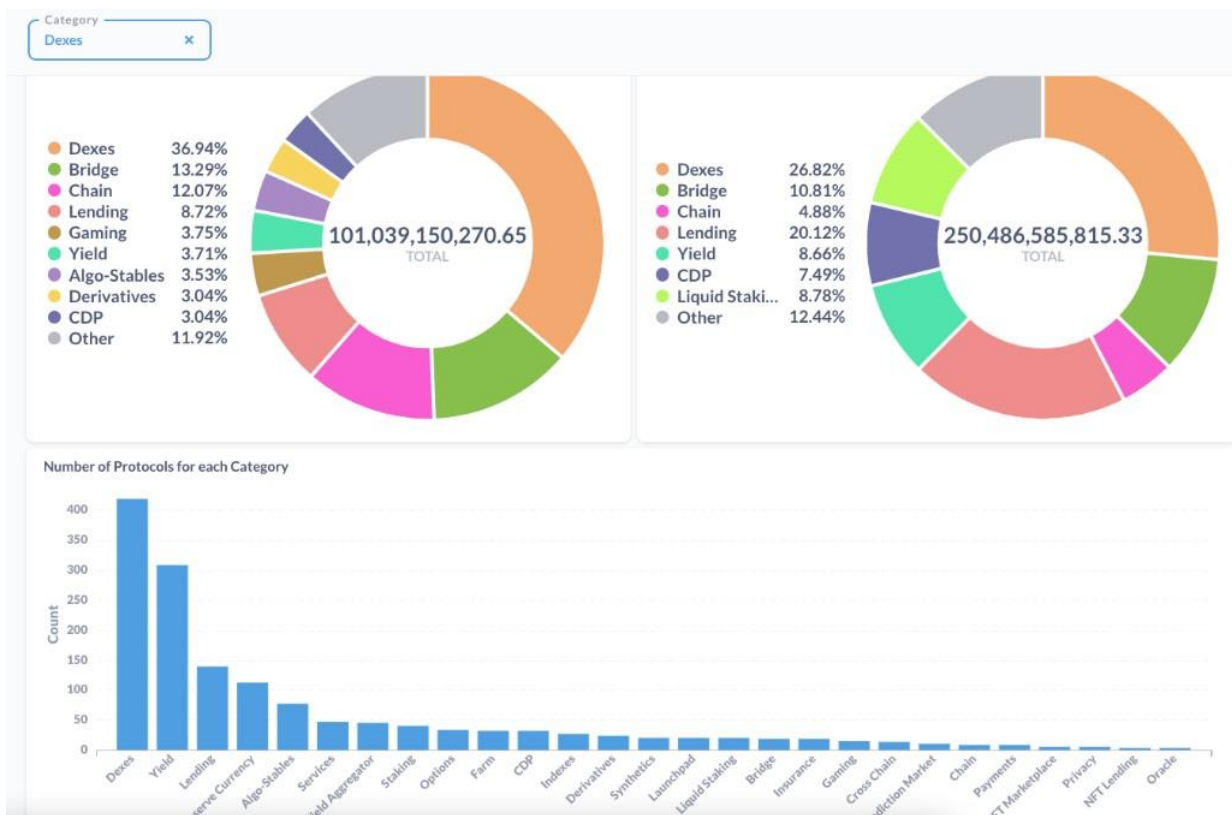


Рис.3.3 – Головне вікно програми при обраному фільтрі

Продовження відображень на екрані при обраному фільтрі зображено далі, на рис.3.4:

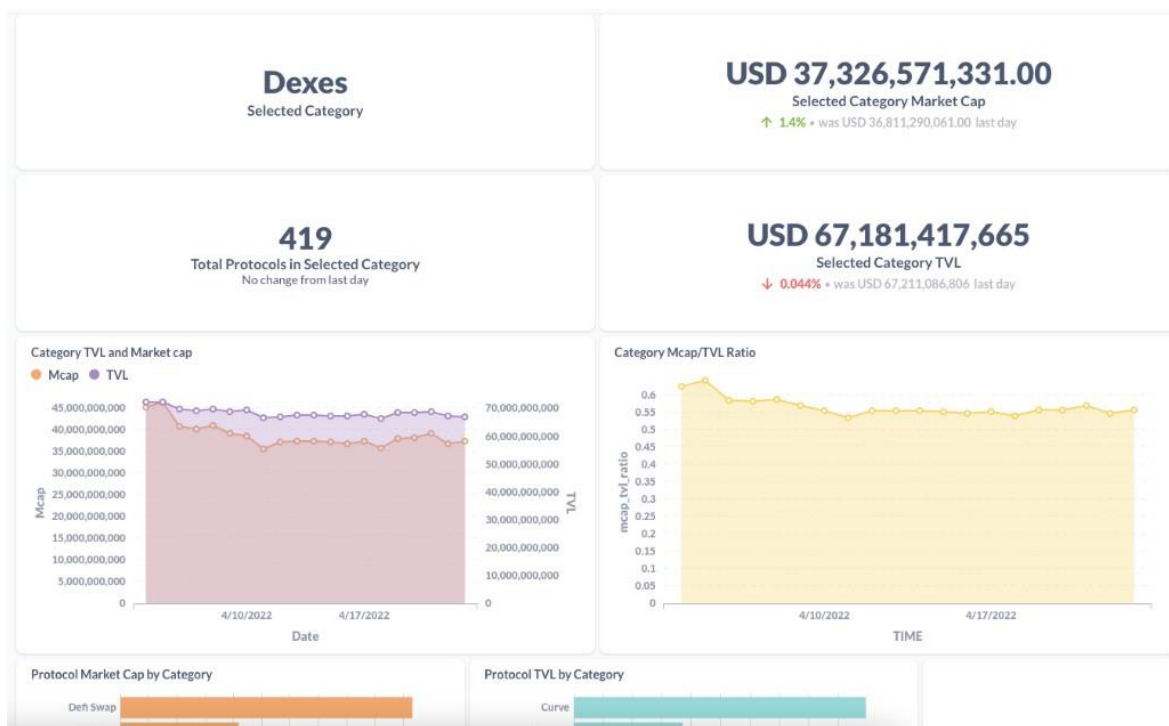


Рис.3.4 – Вікно з графічними оцінками групи криптовалют

При обранні іншого фільтру, наприклад, «Ethereum» дані обновляються, що зображено на рис.3.5:



Рис. 3.5 – Вікно з статистичними оцінками моделі

На рис.3.6 зображено кореляційний аналіз ринкової капіталізації протоколу, сегментований за категоріями. З рисунку видно, які криптокатегорії корелюють.

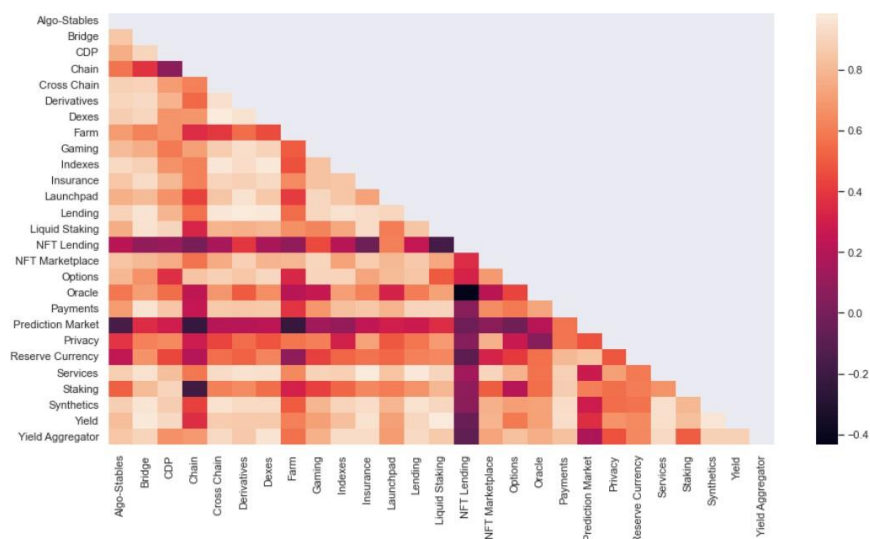


Рис. 3.6 – Кореляція між різними типами сегментів криптовалют



### 3.3.1 Опис даних для дослідження

API Binance — це метод, який дозволяє підключатися до серверів Binance через Python або кілька інших мов програмування. З його допомогою можливо отримувати дані потрібної періодичності, проводити аналіз та прогнозувати курс на майбутнє.

Зокрема, Binance має RESTful API, що використовує HTTP-запити для надсилання запитів та отримання даних. Крім того, доступний також WebSocket, який забезпечує потокову передачу даних, таких як ціни та оновлення облікового запису.

Використовуючи даний API можливо отримати дані в зручному форматі для подальшого аналізу та навчання, тобто pandas DataFrame, приклад зображено на рис.3.7:

|                     | open    | high    | low     | close   | 20sma     |
|---------------------|---------|---------|---------|---------|-----------|
| date                |         |         |         |         |           |
| 2020-06-10 13:25:00 | 9759.92 | 9764.00 | 9758.45 | 9761.43 | 9753.5815 |
| 2020-06-10 13:30:00 | 9761.63 | 9772.85 | 9761.33 | 9770.35 | 9755.3755 |
| 2020-06-10 13:35:00 | 9770.34 | 9779.00 | 9766.04 | 9766.04 | 9757.1975 |
| 2020-06-10 13:40:00 | 9766.04 | 9766.26 | 9754.60 | 9760.32 | 9758.6920 |
| 2020-06-10 13:45:00 | 9760.94 | 9761.50 | 9755.68 | 9755.68 | 9758.9915 |

Рис.3.7 – Приклад отриманого фрейму

Необхідно вирішити задачу прогнозування курсу обраної та вказаної криптовалюти на  $x$  кроків вперед. Набір даних розбитий на тренувальний і тестовий.

Запит має наступний вигляд:

```
request historical candle (or klines) data bars =
client.get_historical_klines('BTCUSD', '1d', timestamp,
limit=1000), де вказано тип криптовалюти та якого еквівалент в доларах,
період, який цікавить та ліміт в вигражі даних.
```

На рис. 3.8 наведено структуру отримання даних по 1 одиниці:

```
[
  [
    1499040000000, // Open time
    "0.01634790", // Open
    "0.80000000", // High
    "0.01575800", // Low
    "0.01577100", // Close
    "148976.11427815", // Volume
    1499644799999, // Close time
    "2434.19055334", // Quote asset volume
    308, // Number of trades
    "1756.87402397", // Taker buy base asset volume
    "28.46694368", // Taker buy quote asset volume
    "17928899.62484339" // Ignore.
  ]
]
```

Рис.3.8 – Структура отримання даних

В таблиці 3.1 наведено опис даних для дослідження, отриманих за допомогою Binance-API.

Таблиця 3.1. Опис вхідних даних

| Змінна     | Значення змінної                     |
|------------|--------------------------------------|
| open_time  | Час відкриття торгів                 |
| open       | Початковий курс                      |
| high       | Найвищий курс                        |
| low        | Найнижчий курс                       |
| close      | Фінальна ціна                        |
| volume     | Обсяг                                |
| close_time | Час закриття торгів                  |
| qav        | Обсяг котирування активу             |
| not        | Кількість торгів                     |
| tbav       | Обсяг базового активу для покупки    |
| tbqv       | Обсяг активу для котирування покупки |
| ignore     | Проігноровано                        |

Дані взято за 5 останніх років (період з 01.09.2017 по 01.09.2022).

### 3.3.2 Аналіз даних

При вигразці даних кожного типу криптовалюти будемо графік розподілу їх та розбиваємо на навчальний та тренувальний набір в співвідношенні 80 на 20.

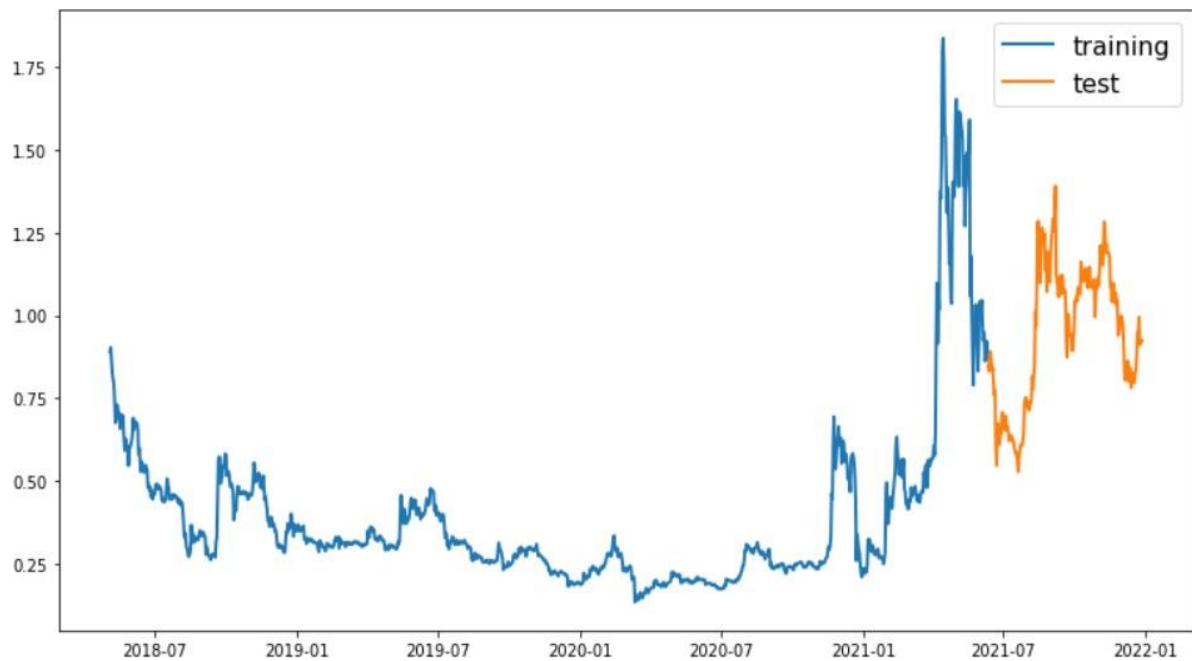


Рис. 3.9 – Розподіл криптовалюти Gafi

Далі наведено фрагмент коду, який нормалізує дані:

```
def normalise_zero_base(continuous):
    return continuous / continuous.iloc[0] - 1

def normalise_min_max(continuous):
    return (continuous - continuous.min()) / (data.max() - continuous.min())
```

На рис 3.10 зображений розподіл нормалізованих змінних:

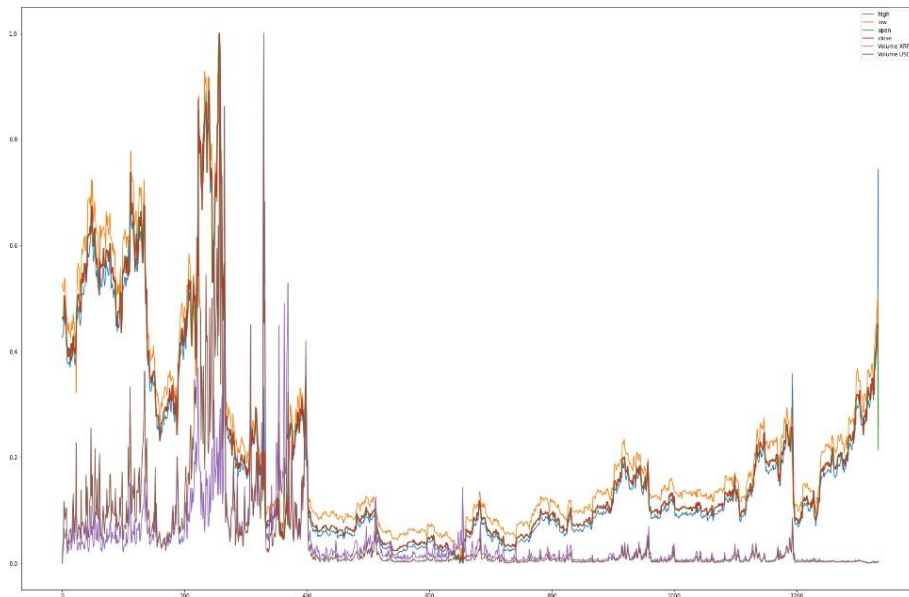


Рис. 3.10 – Розподіл змінних після нормалізації

### 3.3.3 Обробка даних

Процес збору даних та підготовки складається з таких етапів:

1. Вигрузка та фільтр потрібного періоду по конкретній криптовалюти
2. Нормалізація ряду
3. Визначення розміру вікна та розбиття даних на тренувальний та тестовий набір (обраний розмір вікна - 5)

### 3.3.4 Побудова моделей прогнозування

Вхідний набір даних був поділений в співвідношенні 80% до 20%, набір для тренування та відповідно для тесту.

Для прогнозування курсу наступних криптовалют (Bitcoin, Ethereum, Ripple, Binance, GAFI, ALICE, BAKE, GODS) було побудовано наступні регресійні моделі:

- Лінійна регресія

- Багатошарові перцептрони
- ARIMA
- SARIMA
- ARIMAX
- SARIMAX
- LSTM
- RNN (з модифікаціями шарів LSTM, GRU, Bi-LSTM)

На рис. 3.11 зображено кореляцію між змінними в датасеті:

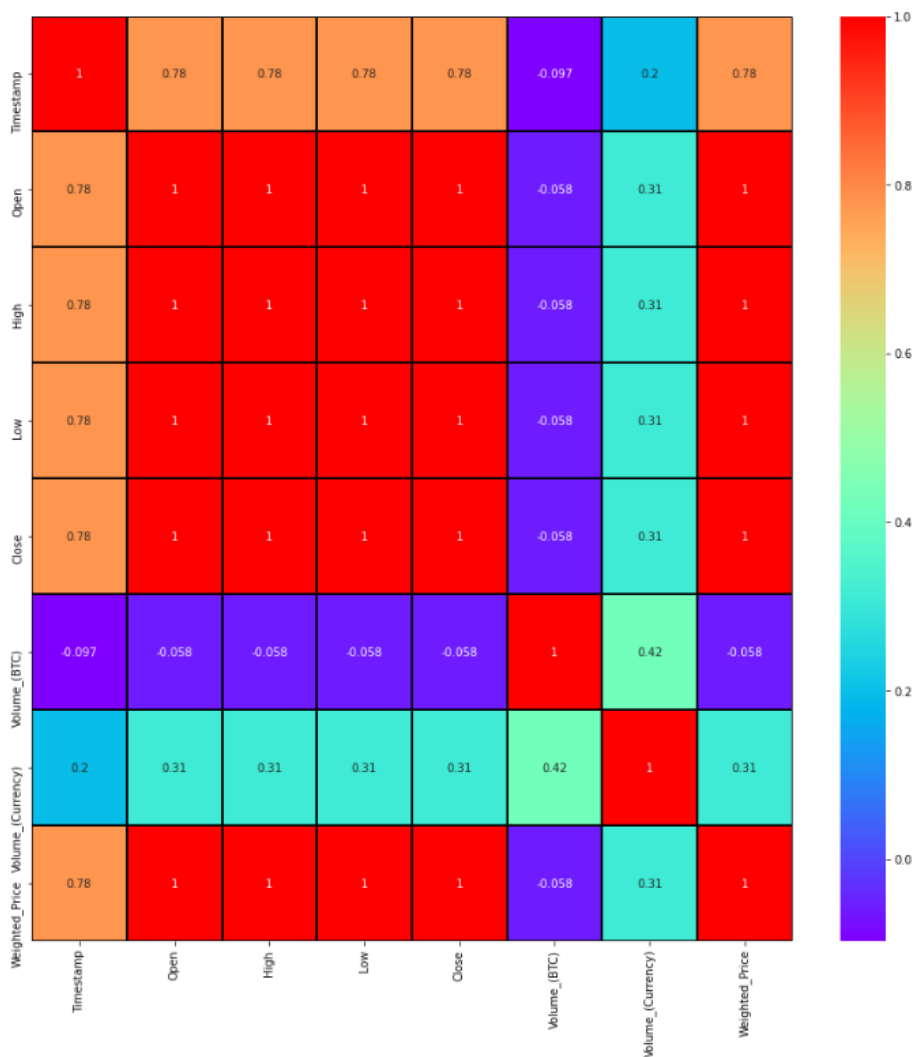


Рис. 3.11 – Залежність вхідних даних від цільової змінної

Перевірено стаціонарність ряду. На рис 3.12 зображено графіки, що

слідують тому, що ряд не є стаціонарним.



Рис. 3.12 – Тест Дікі-Фуллера для перевірки стаціонарності

Середнє, дисперсія і значення варіації ряду змінюються з часом, як бачимо на рис. 3.10 – саме тому ряд не є стаціонарним

При навчанні LSTM без модифікацій було отримано такі результати на виведених епохах, що зображені на рис. 3.13:

| No | Epoch | Model dropout | RMSE results |
|----|-------|---------------|--------------|
| 1  | 10    | 0             | 631.749630   |
| 2  | 100   | 0             | 455.981070   |
| 3  | 1000  | 0             | 825.375050   |
| 4  | 200   | 0             | 360.645110   |
| 5  | 400   | 0             | 354.183680   |
| 6  | 500   | 0             | 288.598660   |
| 7  | 800   | 0             | 292.789670   |
| 8  | 2000  | 0             | 477.914280   |
| 9  | 5000  | 0             | 474.930575   |
| 10 | 500   | 0.1           | 602.140637   |
| 11 | 500   | 0.5           | 313.662300   |

Рис. 3.13 – Якість моделей на проміжних епохах

В роботі використовувались наступні метрики:

- RMSE – це квадратична оцінка, що вимірює середню величину помилки. Це квадратний корінь із середнього квадрата відмінностей між прогнозом і фактичним спостереженням.
- MAE вимірює середню величину помилок у векторі прогнозів без урахування знаку. Це середнє значення за тестовою вибіркою абсолютних

відмінностей між прогнозом і фактичним значенням, де всі різниці мають рівну вагу.

В таблиці 3.2 зображено найкращі показники по навченим моделям для кожної з монет:

Таблиця 3.2 – Метрики точності прогнозування для кожної з монет з найрацями моделями

| Monet    | RMSE      | MAE%  |
|----------|-----------|-------|
| Bitcoin  | 2343.2200 | 4.0   |
| Ethereum | 203.8900  | 5.31  |
| Ripple   | 0.0933    | 4.83  |
| Binance  | 32.6100   | 5.64  |
| GAFI     | 1.4353    | 11.21 |
| ALICE    | 0.054     | 7.64  |
| BAKE     | 0.034     | 18.25 |
| GODS     | 0.087     | 5.16  |

В таблиці 3.3 зображено найкращі моделі для кожного типу монет:

Таблиця 3.3 – Моделі з найвищою точністю для кожної крипто-монети

|          | Модель    | Епоха | RMSE      | MAE%  |
|----------|-----------|-------|-----------|-------|
| Bitcoin  | RNN best* | 50    | 2343.2200 | 4.0   |
| Ethereum | RNN best* | 100   | 203.8900  | 5.31  |
| Ripple   | RNN best* | 10    | 0.0933    | 4.83  |
| Binance  | RNN best* | 100   | 32.6100   | 5.64  |
| GAFI     | RNN best* | 200   | 1.4353    | 11.21 |
| ALICE    | RNN best* | 100   | 0.054     | 7.64  |
| BAKE     | RNN best* | 10    | 0.034     | 18.25 |
| GODS     | RNN best* | 100   | 0.087     | 5.16  |

де RNN best\* - рекурентна нейронна мережа з двома послідовними шарами GRU, та двома послідовними шарами bi-LSTM. Детальна структура нейромережі зображена в Додатку В.

### 3.4 Покращення якості побудованих моделей за рахунок додання зовнішніх даних

Було проведено дослідження (зображено на рис. 3.14) щодо залежності постів в Twitter та курсу криптовалют на прикладі криптовалюти Bitcoin:

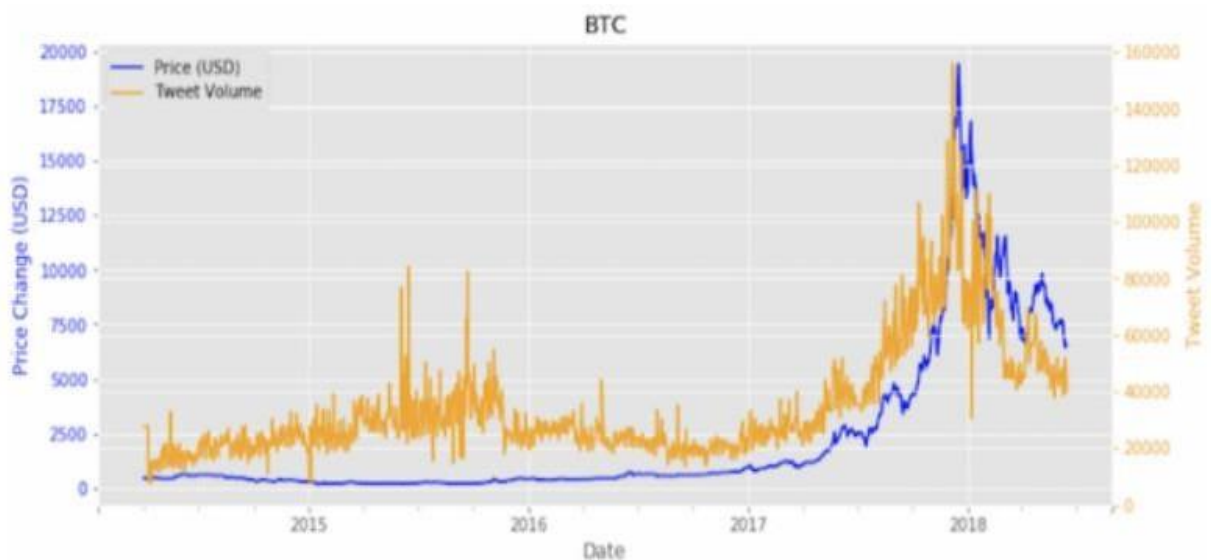


Рис. 3.14 – Залежність кількості твітів на ціни BTC [25]

Як видно з графіку, має сенс додати змінну взятую з публікацій Twitter щодо реакції на криптовалюти - тобто серед всіх постів відфільтрувати, спочатку ті, що стосуються криптовалюти, потім класифікувати їх на позитивні чи негативні і далі додати ці змінні в набір для тренування.

На рис. 3.15 зображено розподіл твітів, що мають 3 характеристики: нейтральний, позитивний чи негативний за період з 2020-10-01 – 2022-09-01:



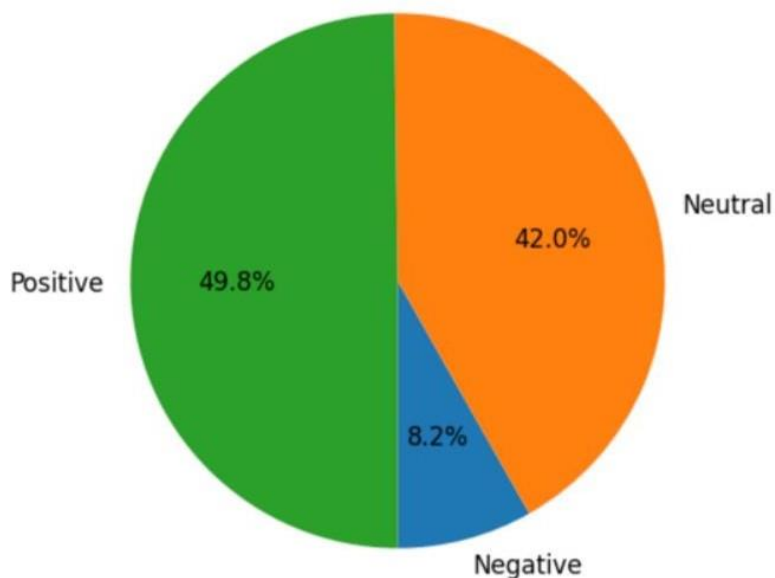


Рис. 3.15 – Розподіл типу твітів по класам «емоційності»

За допомогою наступних методів було отримано такі результати класифікації, рис.3.16:

| SVM                            |          |          |         |
|--------------------------------|----------|----------|---------|
|                                | Positive | Negative | Neutral |
| TP Rate                        | 0.30     | 0.84     | 0.02    |
| FP Rate                        | 0.16     | 0.76     | 0.01    |
| Naive Bayes Classification     |          |          |         |
| TP Rate                        | 0.26     | 0.87     | 0.00    |
| FP Rate                        | 0.14     | 0.80     | 0.00    |
| J48 Decision Tree              |          |          |         |
| TP Rate                        | 0.28     | 0.85     | 0.02    |
| FP Rate                        | 0.14     | 0.78     | 0.01    |
| Random Forest                  |          |          |         |
| TP Rate                        | 0.30     | 0.85     | 0.03    |
| FP Rate                        | 0.14     | 0.77     | 0.02    |
| MultiClass Classifier          |          |          |         |
| TP Rate                        | 0.01     | 0.99     | 0.00    |
| FP Rate                        | 0.03     | 0.94     | 0.00    |
| Iterative Classifier Optimizer |          |          |         |
| TP Rate                        | 0.28     | 0.87     | 0.00    |
| FP Rate                        | 0.14     | 0.79     | 0.00    |

Рис. 3.16 – Результати якості моделей класифікації Твітів

Як бачимо, найращі результати вийшли в методі Random Forest, тому обрали його як метод класифікації твітів.

Далі, для кожного твіту визначили чи є він позитивним чи негативним в класі Крипто, та надалі для кожного періоду порахували кількість позитивних та негативних твітів.

До основного датасету додали 2 нові колонки:

| datestamp           | cnt_positive_tw | cnt_negative_tw |
|---------------------|-----------------|-----------------|
| 2022-01-01 00:00:00 | 75436           | 34532           |
| 2022-01-01 00:06:00 | 79532           | 26432           |

З новими даними моделі були перетреновані, і надалі отримали наступні результати, відображені в таблиці 3.4:

Таблиця 3.4 – Результати якості моделей з новими даними

| Monet    | Old RMSE  | Old MAE% | RMSE      | MAE% | Growth RMSE |
|----------|-----------|----------|-----------|------|-------------|
| Bitcoin  | 2343.2200 | 4.0      | 2234.7500 | 3.7  | 4,63        |
| Ethereum | 203.8900  | 5.31     | 198.7400  | 5.0  | 2,53        |
| Ripple   | 0.0933    | 4.83     | 0.0744    | 3.8  | 20,26       |
| Binance  | 32.6100   | 5.64     | 30.5232   | 5.1  | 6,40        |
| GAFI     | 1.4353    | 11.21    | 1.1232    | 10.7 | 21,74       |
| ALICE    | 0.054     | 7.64     | 0.0542    | 7.65 | -0,37       |
| BAKE     | 0.034     | 18.25    | 0.031     | 17.8 | 8,82        |
| GODS     | 0.087     | 5.16     | 0.079     | 4.9  | 9,20        |

Як бачимо з таблиці 3.4 в середньому на 9,1% маємо приріст по метриці RMSE, що доводить що дані з твітеру підсили прогнозуючу здатність моделі.

### 3.4 Висновки до третього розділу

У третьому розділі було описано власний застосунок для прогнозування курсу криптовалют. Розроблена архітектура застосунку складається з наступних елементів: пристрої вводу-виводу, інтерфейс користувача, API збору даних, система обробки та аналізу інформації, блок моделювання, блок прогнозування та виведення результатів. На основі запропонованої системи обробки даних та моделювання в рамках магістерського проекту було розроблено застосунок мовою програмування Python3.8. Програмний продукт дає можливість завантажувати дані з вказанням криптовалюти та періоду, використовувати моделі прогнозування часовий рядів, обчислювати метрики якості побудованих моделей, відображати графічні характеристики моделей регресії та продивлюватися в реальному часі реальні ціни та порівнювати їх з прогнозом.

Визначені технічні характеристики персонального комп'ютера для оптимальної роботи програмного продукту, а саме: тактова частота процесору, об'єм оперативної пам'яті, об'єм дискової пам'яті, версії операційної системи, додаткові пристрої, що підтримують роботу програмного продукту, та пристрої вводу-виводу, що потрібні для взаємодії користувача та системи. Проведено ґрунтовний огляд інтерфейсу користувача. Розроблений програмний застосунок протестовано на реальних даних, взятих за допомогою Binance API. Побудовано та обрано найкращу модель. За критерієм RMSE та за MAE найкращий результат показала рекурентна нейронна мережа з шарами GRU та Bi-LSTM.

## РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Фінансова індустрія визнає трансформаційний вплив технології блокчейн для отримання нових доходів, підвищення ефективності процесів, покращення досвіду кінцевих користувачів і зниження ризиків у бізнес-операціях. Одним із напрямків, які активно розвиваються, є GameFi — комбінація слів game та finance, блокчейн-ігри типу «грай, щоб заробити», які пропонують гравцям економічні стимули. Екосистема GameFi використовує криптовалюти, незамінні токени (NFT) і технологію блокчейн для створення віртуального ігрового середовища.

GameFi стрімко завойовує традиційну ігрову індустрію після появи Axie Infinity. Технологія приваблює геймерів, пропонуючи їм можливість заробляти гроші, розважаючись.

Проаналізувавши методи аналізу та існуючі рішення, а також тенденції ринку та технологій, був обраний новий, перспективний і «не дуже досліджений» напрямок GameFi. Технологія стрімко розвивається та приваблює геймерів, пропонуючи їм можливість заробляти гроші, розважаючись. Стартап, що пропонує користувачам за підписку отримати прогнози на курс монет може стати успішним на ринку та стати основним вибором для переважної кількості цільової аудиторії.

### 4.1 План розробки стартапу та масштабування його на ринок

Наведемо план розробки стартапу та виведення його на ринок.

Спочатку треба провести маркетинговий аналіз, який включає в себе:

- конкурентний аналіз, щоб зрозуміти, якими методами вирішення проблем вже користуються люди;
- формування ідеї самого проекту та виділення цільової аудиторії;

- розробити стратегію виведення товару на ринок, базуючись на аналізі ринкового середовища.

Наступним кроком являється організація самого стартапу. На цьому етапі мають бути:

- складений весь план та побудований таймлайн розробки та запуску продукту;
- запланований обсяг виробництва та оцінений потенційний обсяг ресурсу, який буде потрібен для виконання плану;
- розраховані витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Далі необхідно виконати фінансово-економічний аналіз та оцінити ризики стартап-проекту, в межах якого:

- визначити обсяг інвестиційних витрат;
- розрахувати основні фінансово-економічні показники проекту (собівартість, ціну продукту/послуги, податковий збір та чистий прибуток) та визначити показники інвестиційної привабливості проекту (рентабельність продажів, період окупності проекту);
- визначити основні ризики проекту та способи для їх запобігання.

Фінальним кроком являється розробка заходів з комерціалізації продукту. Цей крок являється важливим для масштабування та збільшення розмірів продукту.

Для того, щоб залучити інвесторів та знайти різні способи фінансування проекту, необхідно:

- провести дослідження на предмет інтересів потенційних інвесторів та бізнесів;
- скласти інвестиційну пропозицію, яка включає в себе як опис самого продукту та його теперішні розміри, так і можливі шляхи розширення та розвитку;

- обрати канали комунікації із потенційно зацікавленими персонами.

Далі наведемо результати виконання кожного з описаних кроків.

#### 4.2 Опис ідеї стартап-проекту

Стартап-проект полягає у вирішенні проблеми прогнозування курсу криптовалют, що допоможе приймати рішення з покупки/продажу. Суть продукту стартапу полягає у тому, що є сайт, на якому відображаються основні показники по монетам та прогноз їхньої вартості через певні періоди часу.

У таблиці 4.1 наведена інформаційна карта стартапу.

Таблиця 4.1 – Інформаційна карта стартап-проекту

|                           |   |
|---------------------------|---|
| Назва проекту             | CFV (CryptoFutureVision)  |
| Автори проекту            | Яцько Яна Віталіївна  |
| Коротка анотація          | Веб-сайт з інформацією про основні статистики крипто-монет GameFi сегменту та прогнози їх курсу на деякі проміжки часу в майбутньому. |
| Термін реалізації проекту | 12 місяців  |

Продовження таблиці 4.1

|                                   |  |
|-----------------------------------|--|
| Необхідні ресурси                 | Приміщення з комп'ютерами, доступом до Інтернету, доступ до електромережі<br>Програмне забезпечення для розробки, хмарне сховище для даних,<br>Фінансові кошти на оплату заробітної плати виконавцям на термін 12 місяців, а також на такі витрати як: оренда приміщення, комунальні послуги, оренда хмарного сховища тощо |
| Опис проблеми, яку вирішує проект | Продукт вирішує задачу прогнозування курсу криптовалют GameFi сегменту   |
| Головні цілі та завдання проекту  | Метою проекту є створення веб-сайту, на якому буде розміщена інформація про основні статистики криптовалюти та криптопрогноз щодо їх вартості  |
| Очікувані результати              | Привернення технологічних компаній до стартапу, автономна система прогнозування, візуалізації, та користувацький інтерфейс та залучення інвестицій   |

#### 4.3 Технологічний аудит ідеї проекту

Тепер можна розібрати ідею стартапу та провести конкурентний аналіз. У таблиці 4.2 наведений опис ідеї стартапу.

Таблиця 4.2 – Опис ідеї стартапу

| Зміст ідеї  | Напрямки застосування   | Вигоди для користувача   |
|---|---|--|
| <p>Основна ідея являється створення комплексного веб-сайту, що поміщає в себе аналітику існуючих тенденцій, прогнози курсу криптовалют та має зрозумілий користувацький інтерфейс</p> | <p>Опрацювання даних GameFi сегменту, створення аналітичної та прогнозної системи з власним інтерфейсом</p>   | <p>Користувач зможе отримувати агреговані статистики та прогнози курсу</p>   |
|   | <p>Система моніторингу дійсних значень, на базі цього створення удосконалених моделей, збір опитувальників для клієнтів які можливі характеристики їх цікавлять</p> | <p>Моделі прогнозування будуть перенавчатись і покращувати точність, аналітична платформа буде більш широкою та зручною для користування</p> |

Далі проведемо порівняльний аналіз конкурентів проекту та наведемо результати у таблиці 4.3.



Таблиця 4.3 – Порівняльний аналіз конкурентів проекту

| п/п | Техніко-економічні характеристики ідеї                         | (потенційні) товари/концепції конкурентів |                  |                                  |                          |
|-----|--|---|------------------|----------------------------------|--------------------------|
|     |  | Власний проект                            | Messari          | Glassnode                        | Santiment                |
|     | Якісна аналітика статичних та динамічних показників та прогноз | Є аналітика та прогноз                    | Тільки аналітика | Аналітика та прогноз за підписку | Розширена аналітика      |
|     | Доступність по ціні  | Безкоштовний при запуску MVP              | Безкоштовний     | Потрібна платна підписка         | Потрібна платна підписка |
|     | Наявність монет GameFi   | Присутня                                  | Відсутній        | Частково присутня                | Частково присутня        |

Далі аналізуємо реальність технічно здійснити ідею проекту ( таблиця 4.4).

Таблиця 4.4 – Технологічна здійсненність продукту

| п/п   | Ідея проекту  | Технології і реалізації                                    | Наявність технологій           | Доступність технологій |
|---|---|--|--------------------------------|------------------------|
|   | Створення комплексного веб-сайту, що поміщає в себе аналітику існуючих тенденцій, | Використання мови програмування Python                     | Наявні                         | Доступні               |
|   | прогнози курсу криптовалют та має зрозумілий користувачький інтерфейс             | Використання тулів для дизайну веб-інтерфейсів Figma       | Наявні, потребують покращення  | Доступні               |
|   |   | Використання мов HTML, CSS, JavaScript для створення сайту | Наявні, необхідні допрацювання | Доступні               |
|   |   | Використання бази даних MySQL для збереження даних         | Наявні                         | Доступні               |
| <p>Обрана технологія реалізації ідеї проекту: Python, реляційна база даних MySQL, HTML, CSS, JavaScript та застосунок Figma. Для розробки веб-сайту та утримання бази мають бути залучені кошти інвесторів.</p> |   |  |                                |                        |

#### 4.4 Аналіз ринкових можливостей запуску стартап-проекту

Далі проведемо попередній аналіз ринку для запуску стартап-проекту (таблиця 4.5).

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту

| № п/п | Показники ринку (найменування)                           | Характеристика     |
|-------|--|--------------------|
| 1     | Кількість головних гравців, од                           | 4                  |
| 2     | Загальний обсяг продаж, грн/ум.од                        | 6000               |
| 3     | Динаміка ринку (якісна оцінка)                           | Позитивна, зростає |
| 4     | Наявність обмежень для входу (вказати характер обмежень) | Відсутні           |
| 5     | Специфічні вимоги до стандартизації та сертифікації      | Відсутні           |
| 6     | Середня норма рентабельності в галузі (або по ринку), %  | 12%                |

Тепер проведемо характеристику потенційних клієнтів, які можуть бути зацікавлені в проекті (таблиця 4.6).

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

| № п/п | Потреби, що формує ринок                        | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів         | Вимоги споживачів до товару              |
|-------|---|--|---|--|
|       | Розумне вкладення коштів в крипто-актив         | Особисте використання користувачем         | Цікавить примножити кошти   | Простота використання, точність прогнозу |
|       | Покупка довгострокових активів для інвестування | Компанії, холдинги, власне використання    | Особлива потреба у збереженні вартості вкладених грошей та їх примноженні | Точність довгострокового прогнозу        |
|       | Гра в GameFi ігри та швидкі торги               | Геймери                                    | Потреба у короткостроковому прогнозуванні                                 | Точність короткострокового прогнозу      |

Обрахуємо фактори загроз (таблиця 4.7) та можливостей (таблиця 4.8). Проаналізуємо загрози, щоб зрозуміти можливі перешкоди при запуску продукту на ринок. Фактори можливостей же треба обрахувати, щоб знати усі сприятливі умови та по можливості ними скористатися.

Таблиця 4.7 – Фактори загроз

| п/п | Фактор         | Зміст загрози   | Можлива реакція компанії   |
|-----|----------------|---|--|
|     | Конкуренція    | Хоча ринок є відкритим і неосвоєним, на ньому вже є кілька великих гравців, які відомі на ринку, які вже мають свою цільову групу покупців  | Знайти точки додаткової цінності для користувача, та додати особливість – якісний прогноз                    |
|     | Ціна збуту     | Конкуренти можуть коштувати менше через нижчу якість  | Сфокусуватися на якості роботи застосунку та продумати маркетингову стратегію                                |
|     | Якість аналізу | Через комплексність задачі, моделі можуть мати проблеми на певних предметних областях, так наприклад якщо модель навчена на недостатньому об'ємі даних, вона буде давати неточний прогноз | Мати достатню кількість даних для навчання та типи моделей для кожного типу та частоти періоду прогнозування |

Таблиця 4.8 – Фактори можливостей

| п/п | Фактор                      | Зміст можливості  | Можлива реакція компанії  |
|-----|-----------------------------|---|---|
|     | Універсальність             | Продукт не залежить від апаратної платформи, адже це веб-сайт               | Зробити акцент при маркетингу, продовжувати розвиток як окремого продукту                       |
|     | Простота у використанні     | Зрозумілий користувацький інтерфейс   | Реалізувати зручний інтерфейс для користувача   |
|     | Якість та гарантії          | Надавати найбільш якісні прогнози та потрібну аналітику                     | Пропонувати моделі з найкращими результатами, а також надавати усю необхідну технічну підтримку |
|     | Безкоштовний сервіс при MVP | Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку | Розгорнути широкий маркетинг, а також активно боротися за клієнтів конкурентів                  |

Далі розглянемо питання конкуренції, а саме визначимо її тип та рівень (таблиця 4.9).

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища                | У чому проявляється дана характеристика                           | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною ) |
|---|---|---|
| 1. Вказати тип конкуренції: недосконала конкуренція | Представлено мало продуктів та експертів                          | Зробити максимальним збут застосунку  |
| 2. За рівнем конкурентної боротьби: міжнародний     | Наявні проекти, розроблені та можуть бути доступні у всьому світі | Розширити цільову аудиторію, розробити інтерфейс на різних мовах                        |
| 3. За галузевою ознакою: внутрішньогалузева         | Можуть працювати з різними галузями                               | Покращити персоналізацію  |
| 4. Конкуренція за видами товарів: товарно-родова    | Конкуренція з аналізами інших систем та експертів                 | Підтримувати та покращувати якість існуючих функцій                                     |
| 5. За характером конкурентних переваг: нецінова     | Різні компанії пропонують різну якість                            | Розробляти якісніші алгоритми і моделі  |
| 6. За інтенсивністю: марочна                        | Вже представлені компанії із сильним                              | Предметно створити комунікаційну стратегію  |

|  |         |                           |
|--|---------|---------------------------|
|  | брендом | для вибудови свого бренду |
|--|---------|---------------------------|

Далі необхідно виконаємо аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера (таблиця 4.10).

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

|                  |   |   |                             |   |                                     |
|------------------|---|---|-----------------------------|---|-------------------------------------|
| Складові аналізу | Прямі конкуренти у галузі                                       | Потенційні конкуренти                                     | Постачальники               | Клієнти                                   | Товарозаміники                      |
|                  | Інші існуючі системи та продукти                                | Якість, ціни, кількість користувачів, капіталовкладення   | Фактори сили постачальників | Контроль якості, порівняння цін           | Сила бренду, якість, ціна, масштаби |
| Висновки         | Конкуренція з невеликою інтенсивністю, а також підігрітий ринок | Можливості входження на ринок, нові потенційні конкуренти | Постачальники відсутні      | Клієнти не диктують умови роботи на ринку | Товарозаміники відсутні             |

Маючи результати аналізу конкуренції (таблиця 4.10), характеристики ідеї стартап-проекту (таблиця 4.5), характеристики потенційних клієнтів і їх вимоги до продукту (таблиця 4.6) та фактори ринкового середовища (таблиці



4.7 і 4.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності (таблиця 4.11).

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

| № п/п | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|-------|-------------------------------|---|
| 1     | Універсальність               | Продукт не залежить від апаратної платформи   |
| 2     | Простота у використанні       | Зручний інтерфейс   |
| 3     | Якість та гарантії            | Надавати найбільш консультації, опрацьовувати зворотній зв'язок                                     |
| 4     | Безкоштовний сервіс при MVP   | Максимально швидко набрати базу своїх клієнтів та заявити про себе на ринку                         |

Тепер можна провести аналіз сильних та слабких сторін продукту (таблиця 4.12).

Таблиця 4.12 – Порівняльний аналіз сильних та слабких сторін системи

| № п/п | Фактор конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів |    |    |   |   |   |   |  |
|-------|-------------------------------|-----------|-----------------------------|----|----|---|---|---|---|--|
|       |                               |           | -3                          | -2 | -1 | 0 | 1 | 2 | 3 |  |
| 1     | Універсальність               | 20        | +                           |    |    |   |   |   |   |  |
| 2     | Простота у використанні       | 16        |                             | +  |    |   |   |   |   |  |
| 3     | Якість та гарантії            | 10        |                             | +  |    |   |   |   |   |  |
| 4     | Безкоштовний сервіс при MVP   | 17        |                             |    | +  |   |   |   |   |  |

Далі проведемо SWOT-аналіз продукту (таблиця 4.13).

Таблиця 4.13 – SWOT-аналіз стартап-проекту

|  |   |
|--|---|
| <p>Сильні сторони</p> <p>Універсальність</p> <p>Простота у використанні</p> <p>Якість та гарантії</p> <p>Безкоштовний сервіс при MVP</p>                   | <p>Слабкі сторони</p> <p>Відсутність сильного бренду</p> <p>Не сформована база клієнтів</p> <p>Не підключені альтернативні канали маркетингу</p>        |
| <p>Можливості</p> <p>Покращення системи</p> <p>Персоналізація та ведення персональної аналітики</p> <p>Інтеграція з відомими платформами, типу Binance</p> | <p>Загрози</p> <p>Нові системи та експерти</p> <p>Збут</p> <p>Неточність прогнозів для деяких криптовалют, як наслідок – недоцільність використання</p> |

Дякуючи проведенню SWOT-аналізу, ми змогли визначити сильні та слабкі сторони, можливості та загрози, пов'язані з конкуренцією та плануванням стартап-проекту. Далі спроектуємо альтернативну ринкову поведінку для інтеграції стартап-проекту на ринок та приблизний час реалізації системного комплексу, з урахуванням потенційних проектів, що можуть бути виведені на ринок та наведемо результати у таблиці 4.14.

Таблиця 4.14 – Альтернативи ринкового впровадження стартап проекту

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|-------|--|--------------------------------|-------------------|
| 1     | Вихід на ринок з нижче якістю                                  | 70%                            | 4 місяці          |
| 2     | Пропонувати одразу платне використання                         | 50%                            | 6 місяців         |
| 3     | Представлення користувачам системи без інтерфейсу              | 60%                            | 5 місяці          |

У даному пункті був проведений детальний аналіз ринку та продукту. Також відповідно до результатів проведеного конкурентного аналізу, визначених факторів ринку та його сприятливість, описання ідеї та характеристик стартап-проекту, робимо висновок висновок, що існують дуже сприятливі умови для виходу продукту на ринок.

#### 4.5 Розроблення ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, у першу чергу, необхідно проаналізувати цільову аудиторію проекту (таблиця 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

| № п/п                       | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит у межах цільової групи (сегменту) | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|-----------------------------|--|---|---|--------------------------------------|--------------------------|
| 1                           | Персональні користувачі                          | Висока                                  | 80%   | Висока                               | Середня                  |
| 2                           | Великі бізнеси                                   | Низька                                  | 5%  | Низька                               | Низька                   |
| 3                           | Малі та середні бізнеси                          | Середня                                 | 12%   | Середня                              | Середня                  |
| 4                           | Держава  | Низька                                  | 3%  | Низька                               | Висока                   |
| Які цільові групи обрано: 1 |  |   |   |                                      |                          |

Маючи аналіз цільових груп, далі визначимо базову стратегію розвитку продукту (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії розвитку

| № п/п | Обрана альтернатива розвитку | Стратегія охоплення ринку    | Ключові конкурентоспроможні позиції | Базова стратегія розвитку* |
|-------|------------------------------|------------------------------|-------------------------------------|----------------------------|
| 1     | 1                            | Диференційованого маркетингу | Масштабування та максимізація       | Оптимальних витрат         |

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.17, 4.18).

Таблиця 4.17 – Визначення базової стратегії конкурентної поведінки

|  |  |   |                                   |
|--|--|---|-----------------------------------|
| Чи є проект «першопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки* |
| Ні                                     | Так  | Ні  | Виклику лідера, Новизна на ринку  |

Таблиця 4.18 – Визначення стратегії позиціонування

|  |                           |  |   |
|--|---------------------------|--|---|
| Вимоги до товару цільової аудиторії                              | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проекту   | Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)          |
| Універсальність<br>Простота у використанні<br>Якість результатів | Оптимальні витрати        | Універсальність<br>Простота у використанні<br>Якість та гарантії<br>Безкоштовне використання при MVP | Система, яка краще всіх покращує якість користувацьких зображень<br>Система з простим інтерфейсом |

#### 4.6 Розроблення маркетингової програми стартап-проекту

Після проведеного комплексного аналізу, можемо повноцінно описати ключові переваги концепції потенційного товару (таблиця 4.19) та побудувати концепцію маркетингових комунікацій (таблиця 4.20).

Таблиця 4.19 – Ключові переваги концепції потенційного товару

| № п/п | Потреба            | Вигода, яку пропонує товар                  | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)                            |
|-------|--------------------|---|---|
| 1     | Наявність прогнозу | Прогноз проведений та відображений          | На відміну від інших конкурентів, є аналітика та прогноз в різних часових проміжках                     |
| 2     | Універсальність    | Система не залежить від апаратної платформи | Такого виду систему може використовувати будь який користувач і навіть мобільний користувач             |
| 3     | Простий інтерфейс  | Система проста і зрозуміла у використанні   | Система із інтуїтивно зрозумілим інтерфейсом, вдалим дизайном, зрозумілим розташуванням блоків на сайті |

Таблиця 4.20 – Концепція маркетингових комунікацій

| № п/п | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти   | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення                                     | Концепція рекламного звернення  |
|-------|---------------------------------------|--|--|--|---|
| 1     | Пошук спеціалізованих систем          | b2b продажі<br>Зв'язок через теплі контакти<br>Таргетована реклама у соціальних мережах<br>Публікація в спеціалізованих виданнях, журналах | Точність<br>Якість<br>Універсальність      | Поєднати повідомлення про те, що це якісна система, яка є незалежною | Таргетована реклама на цільову аудиторію  |
| 2     | Пошук доступного та дешевого продукту | Рекламні банери в Інтернеті, форуми, реклами від інфлюенсерів  | Простота<br>Безкоштовне використання MVP   | Вселити довіру у бренд та продукт                                    | Реклама у лідерів думок<br>Вивіски в публічних місцях<br>Таргетована реклама на цільову аудиторію |

#### 4.7 Висновки до розділу 4

Даний розділ був присвячений дослідженню стартап-проекту. В якості такого був представлений веб-сайт, що включає в себе аналітику статистичних та динамічних характеристик криптовалют а також їх прогноз на декількох часових проміжках.

У рамках розділу було досліджено розробку стратегій виходу на ринок та маркетинг-стратегії для цього. Зокрема, даний ринок являється сприятливим з невеликою кількістю представлених компаній конкурентів. Оскільки вони дають лише частину функцій, а запропонована система є універсальною та доступною, то у стартап-проекту є всі шанси стати монополістами на ринку.

Також були опрацьовані сильні та слабкі сторони проекту, SWOT аналіз, аналіз конкурентів та цільової аудиторії. На основі всіх досліджень був сформований концепт маркетингової стратегії для обраних цільових аудиторій.



## ВИСНОВКИ

Дана робота націлена на аналіз, побудову та використання прогнозуючих моделей для курсу криптовалют GameFi сегменту. Після ознайомлення з теоретичним матеріалом щодо понять крипто-маркету та моделей прогнозування безперервних величин, основними статистичними та математичними методами побудови моделей для прогнозування курсу криптовалют, було побудовано програму обробки та аналізу табличних даних та побудовано моделі штучного інтелекту. В якості практичного прикладу застосування алгоритму, було розроблено інтерфейс з використанням технологій Python3.8 у середовищі розробки DataSpell 2022 та Jupyter Notebook. У даній програмі було реалізовано та обрано найкращу модель для прогнозування курсу криптовалют, а також збагачено дані з Binance даними з Twitter для покращення якості моделей.

Результати магістерської дипломної роботи:

- створено програму для прогнозування курсу криптовалют
- запропоновано новий ресурс даних та його впровадження в існуючу структуру;
- виконано порівняльний аналіз моделей та обрано найкращі для кожної з криптовалют.

Подальшими напрямками роботи є наступні проблеми:

- вдосконалення розроблених методів прогнозування;
- використання нових готових рішень в сфері моделювання;
- додання додаткового функціоналу до застосунку, наприклад прогнозування в реальному часі, перенавчання моделей в застосунку;
- використання даних з Google щодо новин та основних трендів.

Розроблений програмний продукт показав гарні результати в області прогнозування курсу криптовалют, що підтверджує придатність використання даного застосунку в сфері криптотрейдингу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hellman E. The vast potential for blockchain in libraries. Go To Hellman. URL: <https://go-to-hellman.blogspot.com/2018/06/the-vast-potential-for-blockchain-in.html> (дата звернення: 13.10.2022).
2. Adams C. A blockchain model for other countries. Invest in Blockchain. URL: <https://www.investinblockchain.com/estonia-blockchain-model/> (дата звернення: 16.09.2022).
3. Graglia, Michael J., Mellon C. Blockchain and property in 2018: at the end of the beginning. Newamerica. URL: [https://www.newamerica.org/documents/2121/Graglia\\_Mellon\\_blockchain.pdf](https://www.newamerica.org/documents/2121/Graglia_Mellon_blockchain.pdf) (дата звернення: 02.11.2022).
4. Coleman L. Has the time for regulation arrived?. Cryptocoins News. URL: <https://www.cryptocoinsnews.com/bitfinexs-lesson-has-the-time-for-regulation-arrived/> (дата звернення: 19.10.2022).
5. Hansen J. D. Virtual currencies: international actions and regulations. Perkins Coie. URL: <https://www.perkinscoie.com/en/news-insights/virtual-currenciesinternational-actions-and-regulations.html> (дата звернення: 12.10.2022).
6. Voorhees E. The true cost of bitcoin transactions. E. Voorhees // Bitcoin Journal. URL: <http://bitcoin.xyz/erik-voorhees-true-cost-bitcoin-transactions-2> (дата звернення: 14.09.2022).
7. Investors Underground: Introduction to Technical Analysis [Електронний ресурс] // Investors Underground — Режим доступу: <https://www.investorsunderground.com/technical-analysis/> — (16.09.22).
8. Thuy Nguyen T. T., Vuong Dang X. Supervised support vector machine in predicting foreign exchange trading. Ijisa. 2018. Т. 10, № 9. С. 48–56.
9. A. Radityo, Q. Munajat, and I. Budi. Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods. Advanced

- Computer Science and Information Systems (ICACISIS), 2017 International Conference on, 2017, pp. 433–438
10. Keizer Söze. Bitcoin and Cryptocurrency Technologies: Bitcoin and cryptocurrency investing, cryptocurrency book for beginners / Söze Keizer. – CreateSpace Independent Publishing Platform, 2017. – 506 с. – ISBN 197-906-068-1.
  11. Aimee Vo. Cryptocurrency Trading & Investing: Beginners Guide To Trading & Investing In Bitcoin, Alt Coins & ICOs For Profit / Aimee Vo. – CreateSpace Independent Publishing Platform, 2017. – ISBN 197-792-453-0.
  12. Saifedean Ammous. The Bitcoin Standard: The Decentralized Alternative to Central Banking / Ammous Saifedean. – 1st edition. – Wiley, 2018. – 304 с. – ISBN 111-947-386-1.
  13. Coinmarketcap: Cryptocurrencies by Market Capitalization. URL: <https://coinmarketcap.com>. (дата звернення: 12.10.2022).
  14. Bakker Joris. Technical Analysis in the Cryptocurrency Market / Joris Bakker. – Erasmus School of Economics, 2017. – 53 с.
  15. Andrius Mudinas, Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward / Mudinas Andrius, Zhang Dell, Levene Mark. – Birkbeck: University of London, 2018. – 10 с.
  16. Sarkis Aгаian. Financial Sentiment Analysis Using Machine Learning Techniques / Aгаian Sarkis, Kolm Petter. – New York: International Journal of Investment Management and Financial Innovations, 2017. – 9 с. – ISSN: 2381-1196
  17. Hofman, D., Shannon, C., McManus, B., Lam, K., Assadian, S., Ng, R., Lemieux, V. Building Trust & Protecting Privacy: Analyzing Evidentiary Quality in a Blockchain Proof-of-Concept for Health Research Data Consent Management.

18. Horrigan, John B. Libraries at the Crossroads. URL: <https://www.pewinternet.org/2015/09/15/libraries-at-the-crossroads> (дата звернення: 05.10.2022).
19. Ibáñez, L-D, O'Hara, K., Simperl, E. On Blockchains and the General Data Protection Regulation. EU Blockchain Observatory and Forum: Workshop on GDPR, data policy and compliance. University of Amsterdam. 2018. IEEE Blockchain.
20. Kaur, Harleen M, Afshar Alam, Roshan Jameel, Ashish Kamur Mourya, and Victor Chang. A Proposed Solution and Future Direction for Blockchain-Based Heterogeneous Medicare Data in Cloud Environment. Journal of Medical Systems. 42. no. 8: 156.  
Kim, Bohyun. "Blockchain: Merits, Issues, and Suggestions for Compelling Use Cases." ACRL TechConnect. 2018. <https://acrl.ala.org/techconnect/post/blockchain-merits-issues-and-suggestions-for-compelling-use-cases/>.
21. Cal`es, L., Chalkis, A., Emiris, I.Z., Fisikopoulos, V. Practical volume computation of structured convex bodies, and an application to modeling portfolio dependencies and financial crises. In: Proc. Inter. Symp. Comput. Geom., Budapest. pp. 19:1–19:15 (2018)
22. Chatzis, S.P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., Vlachogiannakis, N. Forecasting stock market crisis events using deep and statistical machine learning techniques. Expert Systems with Appl. 112, 353–371 (2018)
23. Cho, K., van Merrinboer, B., Gulcehre, C., Bougares, F., et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. URL: <https://doi.org/10.3115/v1/D14-1179> (дата звернення: 03.10.2022).

24. Saad, M., Mohaisen, A. Towards characterizing blockchain-based cryptocurrencies for highly accurate predictions. In: Proc. IEEE Conf. Computer Communications (Infocom) Workshops. pp. 704–709 (2018)
25. Real-Time Cryptocurrency Price Prediction by Exploiting IoT Concept and Beyond: Cloud Computing, Data Parallelism and Deep Learning - Scientific Figure on ResearchGate. URL: [https://www.researchgate.net/figure/Cloud-Computing-Connected-Cryptocurrencies-Predicting-Mechanism-using-the-Deep-Learning\\_fig1\\_340285345](https://www.researchgate.net/figure/Cloud-Computing-Connected-Cryptocurrencies-Predicting-Mechanism-using-the-Deep-Learning_fig1_340285345) (дата звернення: 15.10.2022).
26. Nakano, M., Takahashi, A., Takahashi, S. Bitcoin technical trading with artificial neural network. Physical A: Statistical Mechanics and Appl. 510, 587–609 (2018)
27. Jang, H., Lee, J. An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information. IEEE Access 6, 5427–5437 (2018)
28. Jiang, Z., Liang, J. Cryptocurrency portfolio management with deep reinforcement learning. URL: <https://doi.org/10.1109/IntelliSys.2017.8324237> (дата звернення: 17.10.2022).
29. Y. Bengio, P. Simard, and P. Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult, IEEE Trans. Neural Networks, 1994
30. J. Brownlee. Time series prediction with lstm recurrent neural networks in python with keras. Available Mach. com, p. 18, 2016.
31. R. Mittal, S. Arora, and M. P. S. Bhatia. AUTOMATED CRYPTOCURRENCIES PRICES PREDICTION USING MACHINE LEARNING. 2018
32. A. Judmayer, N. Stifter, K. Krombholz, and E. Weippl. Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms. Synth. Lect. Inf. Secur. Privacy, Trust, 2017.

33. Squark. ROOT MEAN SQUARE ERROR OR RMSE. URL: <https://squarkai.com/root-mean-square-error-or-rmse/#.XWVCLegzZPY> (дата звернення: 24.10.2022).
34. Augmented Dickey-Fuller Test in Python. URL: <https://www.hackdeploy.com/augmented-dickey-fuller-test-in-python> (дата звернення: 24.10.2022).
35. ARIMA models for time series forecasting. URL: <https://people.duke.edu/~rnau/411arim.html> (дата звернення: 09.10.2022).
36. ARIMA models // Forecasting: Principles and Practice. URL: <https://otexts.com/fpp2/arima.html> (дата звернення: 20.10.2022).
37. Cryptocurrency Predictions with ARIMA. URL: <https://www.kaggle.com/taniaj/cryptocurrency-predictions-with-arima> (дата звернення: 16.09.2022).
38. Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.
39. Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term memory. *Neural computation*, 9(8):1735–1780, 1997.
40. Sepp Hochreiter, Bengio Yoshua, Frasconi Paolo, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer and Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
41. J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons.
42. *Proceedings of the National Academy of Sciences*, 81:3088–3092, 1984.
43. Varuna Jayasiri. Vanilla LSTM with numpy. URL: [http://blog.varunajayasiri.com/numpy\\_lstm.html](http://blog.varunajayasiri.com/numpy_lstm.html) (дата звернення: 05.10.2022).
44. M. I. Jordan. Serial order: A parallel distributed processing approach. Technical Report ICS Report 8604, Institute for

45. Cognitive Science, University of California, San Diego, 1986.
46. Anjali Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, and Vivek Ramavajjala. Smart reply: Automated response suggestion for email. CoRR, abs/1606.04870, 2016.
47. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. URL: <http://karpathy.github.io/2015/05/21/rnneffectiveness> (дата звернення: 05.11.2022).
48. Yuliya Kyrychko and Stephen Hogan. On the use of delay equations in engineering applications. 16:943–960, Jul 2010.
49. Omer Levy, Kenton Lee, Nicholas FitzGerald, and Luke Zettlemoyer. Long short-term memory as a dynamically computed element-wise weighted sum. CoRR, abs/1805.03716, May 2018.
50. Qianli Liao and Tomaso A. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. CoRR, abs/1604.03640, Apr 2016.
51. Henry W. Lin and Max Tegmark. Criticality in formal languages and statistical physics. Entropy, 19(7):299, Aug 2017.
52. Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. CoRR, abs/1506.00019, 2015.
53. Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In Jennifer G. Dy and Andreas Krause, editors, ICML, volume 80 of Proceedings of Machine Learning Research, pages 3282–3291. PMLR, Nov 2018.
54. Mallya A. Introduction to RNNs. URL: [http://slazebni.cs.illinois.edu/spring17/lec02\\_rnn.pdf](http://slazebni.cs.illinois.edu/spring17/lec02_rnn.pdf) (дата звернення: 24.10.2022).

55. Mallya A. LSTM forward and backward pass. URL: <http://arunmallya.github.io/writeups/nn/lstm/index.html> (дата звернення: 14.09.2022).
56. McClellan J. H., Parks T. W., Rabiner L. R. A computer program for designing optimum fir linear phase digital filters. IEEE transactions on audio electroacoustics. 1984. Т. 21, № 6. С. 506–526.
57. Metropolis N., Rosenbluth A., Rosenbluth M. Equation of state calculations by fast computing machines. Science. 1985. Т. 21, № 1087. С. 123–130.
58. Minsky M., Papert S. Perceptrons. MIT press. 1995. 14 груд. С. 30–39.
59. Olah C. Understanding LSTM Networks. colah.github.io. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs> (дата звернення: 15.09.2022).
60. Palangi H., Deng L. Deep sentence embedding using the long short term memory network: analysis and application to information retrieval. CoRR. 2015. Т. 1502, № 06922. С. 312–340.
61. Pascanu R., Mikolov T., Bengio Y. On the difficulty of training recurrent neural networks. International conference on machine learning : International Conference, м. Tokyo, 11 груд. 2013 р. С. 1310–1318.
62. Pascanu R. On recurrent and deep neural networks : doctoral dissertation. Montréal, 2014. 412 с.
63. Pineda F. L. Generalization of backpropagation to recurrent and higher order neural networks : doctoral dissertation. New York, 2018. 456 с.
64. Яцько Я. В., Терентьев О. М. Використання мовної Ві-LSTM для підвищення точності прогнозування курсу криптовалют // Колективна монографія за матеріалами XXI Міжнародної науково практичної конференції (Київ, 14-16 листопада 2022 р.) / За заг. ред. С.О. Довгого. – К.: ТОВ «Видавництво «Юстон», 2022. – 56-58 с.



## ДОДАТОК А. Лістинг програмного модулю

## Module getting data

```
pip install python-binance
```

```
set binance_api="key"  
set binance_secret="secret"
```

```
export binance_api="key"  
export binance_secret="secret"
```

```
import os
```

```
from binance.client import Client
```

```
import pandas as pd
```

```
import json
```

```
# init
```

```
api_key = os.environ.get('binance_api')
```

```
api_secret = os.environ.get('binance_secret')
```

```
client = Client(api_key, api_secret)
```

```
client.API_URL = 'https://testnet.binance.vision/api'
```

```
# get balances for all assets & some account information  
print(client.get_account())
```

```
# get latest price from Binance API
```

```
btc_price = client.get_symbol_ticker(symbol="BTCUSDT")
```

```
# print full output (dictionary)
```

```
print(btc_price)
```

```
# valid intervals - 1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 3d, 1w, 1M
```

```
# get timestamp of earliest date data is available
```

```
timestamp = client._get_earliest_valid_timestamp('BTCUSDT', '1d')
```

```
print(timestamp)
```

```
# request historical candle (or klines) data
```

```
bars = client.get_historical_klines('BTCUSDT', '1d', timestamp, limit=1000)
```

```
# option 1 - save to file using json method
```

```
with open('btcBars.json', 'w') as e:
    json.dump(bars, e)
```

# option 2 - save as CSV file using the csv writer library

```
with open('btcBars.csv', 'w', newline='') as f:
```

```
    wr = csv.writer(f)
    for line in bars:
        wr.writerow(line)
```

# option 3 - save as CSV file without using a library.

```
with open('btcBars2.csv', 'w') as d:
```

```
    for line in bars:
        d.write(f'{line[0]}, {line[1]}, {line[2]}, {line[3]}, {line[4]}\n')
```

# delete unwanted data - just keep date, open, high, low, close

```
for line in bars:
    del line[5:]
```

# option 4 - create a Pandas DataFrame and export to CSV

```
btc_df = pd.DataFrame(bars, columns=['date', 'open', 'high', 'low', 'close'])
btc_df.set_index('date', inplace=True)
print(btc_df.head())
```

# export DataFrame to csv

```
btc_df.to_csv('btcBars3.csv')
```

Module data processing

```
import numpy as np # linear algebra
```

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
import matplotlib.pyplot as plt
```

# Input data files are available in the "../input/" directory.

# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

```
from plotly import tools
```

```
import plotly.plotly as py
```

```
from plotly.offline import init_notebook_mode, iplot
```

```
init_notebook_mode(connected=True)
```

```
import plotly.graph_objs as go
```

```
import gc
```

```

import matplotlib.pyplot as plt
import seaborn as sns

#import os
#print(os.listdir("../input"))

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))

# First thing is to fix the data for bars/candles where there are no trades.
# Volume/trades are a single event so fill na's with zeroes for relevant fields...
data['Volume_(BTC)'].fillna(value=0, inplace=True)
data['Volume_(Currency)'].fillna(value=0, inplace=True)
data['Weighted_Price'].fillna(value=0, inplace=True)

# next we need to fix the OHLC (open high low close) data which is a continuous
timeseries so
# lets fill forwards those values...
data['Open'].fillna(method='ffill', inplace=True)
data['High'].fillna(method='ffill', inplace=True)
data['Low'].fillna(method='ffill', inplace=True)
data['Close'].fillna(method='ffill', inplace=True)

data.head()

# create valid date range
start = datetime.datetime(2018, 1, 1, 0, 0, 0, 0, pytz.UTC)
end = datetime.datetime(2022, 11, 11, 0, 0, 0, 0, pytz.UTC)

# find rows between start and end time and find the first row (00:00 monday
morning)
weekly_rows = data[(data['Timestamp'] >= start) & (data['Timestamp'] <=
end)].groupby([pd.Grouper(key='Timestamp', freq='W-
MON')]).first().reset_index()
weekly_rows.head()

# We use Plotly to create the plots https://plot.ly/python/
trace1 = go.Scatter(
    x = weekly_rows['Timestamp'],
    y = weekly_rows['Open'].astype(float),
    mode = 'lines',
    name = 'Open'

```

```

)

trace2 = go.Scatter(
    x = weekly_rows['Timestamp'],
    y = weekly_rows['Close'].astype(float),
    mode = 'lines',
    name = 'Close'
)

trace3 = go.Scatter(
    x = weekly_rows['Timestamp'],
    y = weekly_rows['Weighted_Price'].astype(float),
    mode = 'lines',
    name = 'Weighted Avg'
)

layout = dict(
    title='Historical Bitcoin Prices (2018-2022) with the Slider ',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                #change the count to desired amount of months.
                dict(count=1,
                    label='1m',
                    step='month',
                    stepmode='backward'),
                dict(count=6,
                    label='6m',
                    step='month',
                    stepmode='backward'),
                dict(count=12,
                    label='1y',
                    step='month',
                    stepmode='backward'),
                dict(count=36,
                    label='3y',
                    step='month',
                    stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(
            visible = True
        ),
        type='date'

```

```

    )
)

data = [trace1,trace2, trace3]
fig = dict(data=data, layout=layout)
iplot(fig, filename = "Time Series with Rangeslider")

```

```

trace1 = go.Scatter(
    x = weekly_rows['Timestamp'],
    y = weekly_rows['Volume_(Currency)'].astype(float),
    mode = 'lines',
    name = 'Bitcoin Price (Open)'
)

```

```

layout = dict(
    title='Historical Bitcoin Volume (USD) (2018-2022) with the slider',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1,
                    label='1m',
                    step='month',
                    stepmode='backward'),
                dict(count=6,
                    label='6m',
                    step='month',
                    stepmode='backward'),
                dict(count=12,
                    label='1y',
                    step='month',
                    stepmode='backward'),
                dict(count=36,
                    label='3y',
                    step='month',
                    stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(
            visible = True
        ),
        type='date'
    )
)

```

```

    )
)

data = [trace1]
fig = dict(data=data, layout=layout)
iplot(fig, filename = "Time Series with Rangeslider")

```

#BTC Volume vs USD visualization

```

trace = go.Scattergl(
    y = weekly_rows['Volume_(BTC)'].astype(float),
    x = weekly_rows['Weighted_Price'].astype(float),
    mode = 'markers',
    marker = dict(
        color = '#FFBAD2',
        line = dict(width = 1)
    )
)
layout = go.Layout(
    title='BTC Volume v/s USD',
    xaxis=dict(
        title='Weighted Price',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    ),
    yaxis=dict(
        title='Volume BTC',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)
data = [trace]
fig = go.Figure(data=data, layout=layout)
iplot(fig, filename='compare_webgl')

```

```

def normalise_zero_base(continuous):
    return continuous / continuous.iloc[0] - 1

```

```

def normalise_min_max(continuous):
    return (continuous - continuous.min()) / (data.max() - continuous.min())

def extract_window_data(continuous, window_len=5, zero_base=True):
    window_data = []
    for idx in range(len(continuous) - window_len):
        tmp = continuous[idx: (idx + window_len)].copy()
        if zero_base:
            tmp = normalise_zero_base(tmp)
        window_data.append(tmp.values)
    return np.array(window_data)

def prepare_data(continuous, aim, window_len=10, zero_base=True,
test_size=0.2):
    X_train = extract_window_data(train_data, window_len, zero_base)
    X_test = extract_window_data(test_data, window_len, zero_base)
    y_train = train_data[aim][window_len:].values
    y_test = test_data[aim][window_len:].values
    if zero_base:
        y_train = y_train / train_data[aim][: -window_len].values - 1
        y_test = y_test / test_data[aim][: -window_len].values - 1

    return train_data, test_data, X_train, X_test, y_train, y_test
import numpy as np

```

### Module building models

```

def build_lstm_model(input_data, output_size, neurons, activ_func='linear',
                    dropout=0.2, loss='mse', optimizer='adam'):
    model = Sequential()
    model.add(LSTM(neurons, input_shape=(input_data.shape[1],
input_data.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(units=output_size))
    model.add(Activation(activ_func))

    model.compile(loss=loss, optimizer=optimizer)
    return model
np.random.seed(245)
window_len = 5
test_size = 0.2
zero_base = True
lstm_neurons = 50
epochs = 20

```

```

batch_size = 32
loss = 'mse'
dropout = 0.24
optimizer = 'adam'
train_data, test_data, X_train, X_test, y_train, y_test = prepare_data(
    data, aim, window_len=window_len, zero_base=zero_base, test_size=test_size)

model = build_lstm_model(
    X_train, output_size=1, neurons=lstm_neurons, dropout=dropout, loss=loss,
    optimizer=optimizer)
modelfit = model.fit(
    X_train, y_train, validation_data=(X_test, y_test), epochs=epochs,
    batch_size=batch_s)

```

```

import matplotlib.pyplot as plt
plt.plot(modelfit.history['loss'],'r',linewidth=2, label='Training loss')
plt.plot(modelfit.history['val_loss'], 'g',linewidth=2, label='Validation loss')
plt.title('LSTM Neural Networks - XRP Model')
plt.xlabel('Epochs numbers')
plt.ylabel('MSE numbers')
plt.show()

```

```

preds = test[aim].values[:-window_len] * (preds + 1)
preds = pd.Series(index=targets.index, data=preds)
line_plot(targets, preds, 'actual', 'prediction', lw=3)

```

```

#mean squared cost fuction
def cost(x,y,w,b):
    n=x.shape[0]
    yhat=0.0
    diff=0.0
    for i in range(n):
        if i!=213:
            yhat=w*x[i]+b#function is linear as y=mx+c
            diff=diff+(yhat-y[i])**2
    diff=diff/(2*n)
    return diff

```

```

#calculate gradient
def calc_gradient(x,y,w,b):
    m=x.shape[0]
    yhat=0.0

```



```

dj_dw=0.0
dj_db=0.0
for i in range(m):
    if i!=213:
        yhat=np.dot(w,x[i])+b
        # dj_dw=dj_dw+(yhat)
        # dj_db=dj_db+(yhat)
        dj_dw=dj_dw+(yhat-y[i])*x[i]
        dj_db=dj_db+(yhat-y[i])
#     if(i%100==0):
#         print(f'Found :{dj_dw,dj_db}')
dj_dw=dj_dw/m
dj_db=dj_db/m
return dj_dw,dj_db

```

```

def gradient_descent(x, y, w, b, pd_hist, w_hist, alpha, iter):
    dw = 0.0
    db = 0.0
    w_prev = 0.0
    for i in range(iter):
        dw, db = calc_gradient(x, y, w, b)
        w = w - alpha * dw
        if abs(w_prev - w) < 0.00001:
            break
        w_prev = w
        pd_hist.append(cost(x, y, w, b))
        w_hist.append(w)
        b = b - alpha * db
        if (i % 10 == 0):
            print(f'Calculating the weight and constant:{w, b}')

    return w, b, pd_hist, w_hist

```

```

x_train=train_norm['x']
y_train=train_norm['y']
w=0.0
b=0.0
pd_hist=[]
w_hist=[]
iterations=8000
alpha=0.2

```

```
w_f,b_f,pd_hist,w_hist=gradient_descent(x_train,y_train,w,b,pd_hist,w_hist,alpha
,iterations)
print(f'Found the weight and constant:{w_f,b_f}')
```

Evaluation  
of  
various  
combinations  
of  
multi - layer  
perceptrons

```
# Import libraries for deep learning
from keras.wrappers.scikit_learn import KerasRegressor
from keras.models import Sequential
from keras.layers import Dense
```

```
# define baseline model
def baseline(v):
    # create model
    model = Sequential()
    model.add(Dense(v * (c - 1), input_dim=v * (c - 1), init='normal',
activation='relu'))
    model.add(Dense(1, init='normal'))
    # Compile model
    model.compile(loss='mean_absolute_error', optimizer='adam')
    return model
```

```
# define smaller model
def smaller(v):
    # create model
    model = Sequential()
    model.add(Dense(v * (c - 1) / 2, input_dim=v * (c - 1), init='normal',
activation='relu'))
    model.add(Dense(1, init='normal', activation='relu'))
    # Compile model
    model.compile(loss='mean_absolute_error', optimizer='adam')
    return model
```

```
# define deeper model
def deeper(v):
```

```

# create model
model = Sequential()
model.add(Dense(v * (c - 1), input_dim=v * (c - 1), init='normal',
activation='relu'))
model.add(Dense(v * (c - 1) / 2, init='normal', activation='relu'))
model.add(Dense(1, init='normal', activation='relu'))
# Compile model
model.compile(loss='mean_absolute_error', optimizer='adam')
return model

```

```

# Optimize using dropout and decay
from keras.optimizers import SGD
from keras.layers import Dropout
from keras.constraints import maxnorm

```

```

def dropout(v):
    # create model
    model = Sequential()
    model.add(Dense(v * (c - 1), input_dim=v * (c - 1), init='normal',
activation='relu', W_constraint=maxnorm(3)))
    model.add(Dropout(0.2))
    model.add(Dense(v * (c - 1) / 2, init='normal', activation='relu',
W_constraint=maxnorm(3)))
    model.add(Dropout(0.2))
    model.add(Dense(1, init='normal', activation='relu'))
    # Compile model
    sgd = SGD(lr=0.1, momentum=0.9, decay=0.0, nesterov=False)
    model.compile(loss='mean_absolute_error', optimizer=sgd)
    return model

```

```

# define decay model
def decay(v):
    # create model
    model = Sequential()
    model.add(Dense(v * (c - 1), input_dim=v * (c - 1), init='normal',
activation='relu'))
    model.add(Dense(1, init='normal', activation='relu'))
    # Compile model
    sgd = SGD(lr=0.1, momentum=0.8, decay=0.01, nesterov=False)
    model.compile(loss='mean_absolute_error', optimizer=sgd)
    return model

```

```

est_list = []
# uncomment the below if you want to run the algo
# est_list =
[('MLP',baseline),('smaller',smaller),('deeper',deeper),('dropout',dropout),('decay',d
ecay)]

for name, est in est_list:

    algo = name

    # Accuracy of the model using all features
    for m, i_cols_list in X_all:
        model = KerasRegressor(build_fn=est, v=1, nb_epoch=10, verbose=0)
        model.fit(X_train[:, i_cols_list], Y_train)
        result = mean_absolute_error(numpy.expm1(Y_val),
numpy.expm1(model.predict(X_val[:, i_cols_list])))
        mae.append(result)
        print(name + " %s" % result)

    comb.append(algo)

if (len(est_list) == 0):
    mae.append(1168)
    comb.append("MLP" + " baseline")

##Set figure size
# plt.rc("figure", figsize=(25, 10))

# Plot the MAE of all combinations
fig, ax = plt.subplots()
plt.plot(mae)
# Set the tick names to names of combinations
ax.set_xticks(range(len(comb)))
ax.set_xticklabels(comb, rotation='vertical')
# Plot the accuracy for all combinations
plt.show()

# with GPU
import numpy as np
import pandas as pd
import os

```

```

import time
import gc
import random
from tqdm._tqdm_notebook import tqdm_notebook as tqdm
from keras.preprocessing import text, sequence
import torch
from torch import nn
from torch.utils import data
from torch.nn import functional as F

# disable progress bars when submitting
def is_interactive():
    return 'SHLVL' not in os.environ

if not is_interactive():
    def nop(it, *a, **k):
        return it

    tqdm = nop

def seed_everything(seed=1234):
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True
seed_everything()

CRAWL_EMBEDDING_PATH = '../input/fasttext-crawl-300d-2m/crawl-300d-2M.vec'
GLOVE_EMBEDDING_PATH = '../input/glove840b300dtxt/glove.840B.300d.txt'
NUM_MODELS = 2
LSTM_UNITS = 128
DENSE_HIDDEN_UNITS = 4 * LSTM_UNITS
MAX_LEN = 220

def get_coefs(word, *arr):
    return word, np.asarray(arr, dtype='float32')

```

```

def load_embeddings(path):
    with open(path) as f:
        return dict(get_coefs(*line.strip().split(' ')) for line in tqdm(f))

def build_matrix(word_index, path):
    embedding_index = load_embeddings(path)
    embedding_matrix = np.zeros((len(word_index) + 1, 300))
    unknown_words = []

    for word, i in word_index.items():
        try:
            embedding_matrix[i] = embedding_index[word]
        except KeyError:
            unknown_words.append(word)
    return embedding_matrix, unknown_words

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def train_model(model, train, test, loss_fn, output_dim, lr=0.001,
                batch_size=512, n_epochs=4,
                enable_checkpoint_ensemble=True):
    param_lrs = [{'params': param, 'lr': lr} for param in model.parameters()]
    optimizer = torch.optim.Adam(param_lrs, lr=lr)

    scheduler = torch.optim.lr_scheduler.LambdaLR(optimizer, lambda epoch: 0.6
    ** epoch)

    train_loader = torch.utils.data.DataLoader(train, batch_size=batch_size,
    shuffle=True)
    test_loader = torch.utils.data.DataLoader(test, batch_size=batch_size,
    shuffle=False)
    all_test_preds = []
    checkpoint_weights = [2 ** epoch for epoch in range(n_epochs)]

    for epoch in range(n_epochs):
        start_time = time.time()

        scheduler.step()

        model.train()

```

```

avg_loss = 0.

for data in tqdm(train_loader, disable=False):
    x_batch = data[:-1]
    y_batch = data[-1]

    y_pred = model(*x_batch)
    loss = loss_fn(y_pred, y_batch)

    optimizer.zero_grad()
    loss.backward()

    optimizer.step()
    avg_loss += loss.item() / len(train_loader)

model.eval()
test_preds = np.zeros((len(test), output_dim))

for i, x_batch in enumerate(test_loader):
    y_pred = sigmoid(model(*x_batch).detach().cpu().numpy())

    test_preds[i * batch_size:(i + 1) * batch_size, :] = y_pred

all_test_preds.append(test_preds)
elapsed_time = time.time() - start_time
print('Epoch {}/{} \t loss={:.4f} \t time={:.2f}s'.format(
    epoch + 1, n_epochs, avg_loss, elapsed_time))

if enable_checkpoint_ensemble:
    test_preds = np.average(all_test_preds, weights=checkpoint_weights, axis=0)
else:
    test_preds = all_test_preds[-1]

return test_preds

class SpatialDropout(nn.Dropout2d):
    def forward(self, x):
        x = x.unsqueeze(2) # (N, T, 1, K)
        x = x.permute(0, 3, 2, 1) # (N, K, 1, T)
        x = super(SpatialDropout, self).forward(x) # (N, K, 1, T), some features are
masked
        x = x.permute(0, 3, 2, 1) # (N, T, 1, K)
        x = x.squeeze(2) # (N, T, K)

```

```

return x

class NeuralNet(nn.Module):
    def __init__(self, embedding_matrix, num_aux_targets):
        super(NeuralNet, self).__init__()
        embed_size = embedding_matrix.shape[1]

        self.embedding = nn.Embedding(max_features, embed_size)
        self.embedding.weight = nn.Parameter(torch.tensor(embedding_matrix,
dtype=torch.float32))
        self.embedding.weight.requires_grad = False
        self.embedding_dropout = SpatialDropout(0.3)

        self.lstm1 = nn.LSTM(embed_size, LSTM_UNITS, bidirectional=True,
batch_first=True)
        self.lstm2 = nn.LSTM(LSTM_UNITS * 2, LSTM_UNITS,
bidirectional=True, batch_first=True)

        self.linear1 = nn.Linear(DENSE_HIDDEN_UNITS,
DENSE_HIDDEN_UNITS)
        self.linear2 = nn.Linear(DENSE_HIDDEN_UNITS,
DENSE_HIDDEN_UNITS)

        self.linear_out = nn.Linear(DENSE_HIDDEN_UNITS, 1)
        self.linear_aux_out = nn.Linear(DENSE_HIDDEN_UNITS,
num_aux_targets)

    def forward(self, x):
        h_embedding = self.embedding(x)
        h_embedding = self.embedding_dropout(h_embedding)

        h_lstm1, _ = self.lstm1(h_embedding)
        h_lstm2, _ = self.lstm2(h_lstm1)

        # global average pooling
        avg_pool = torch.mean(h_lstm2, 1)
        # global max pooling
        max_pool, _ = torch.max(h_lstm2, 1)

        h_conc = torch.cat((max_pool, avg_pool), 1)
        h_conc_linear1 = F.relu(self.linear1(h_conc))
        h_conc_linear2 = F.relu(self.linear2(h_conc))

```



```

hidden = h_conc + h_conc_linear1 + h_conc_linear2

result = self.linear_out(hidden)
aux_result = self.linear_aux_out(hidden)
out = torch.cat([result, aux_result], 1)

return out

def preprocess(data):
    """
    Credit goes to https://www.kaggle.com/gpreda/jigsaw-fast-compact-solution
    """
    punct = "/- '?!.,#$$%\\" +
    '∞θ÷α•à-β∅³π‘ř’°£€\×™√²—&'
    def clean_special_chars(text, punct):
        for p in punct:
            text = text.replace(p, ' ')
        return text

    data = data.astype(str).apply(lambda x: clean_special_chars(x, punct))
    return data

train_dataset = data.TensorDataset(x_train_torch, y_train_torch)
test_dataset = data.TensorDataset(x_test_torch)

all_test_preds = []

for model_idx in range(NUM_MODELS):
    print('Model ', model_idx)
    seed_everything(1234 + model_idx)

    model = NeuralNet(embedding_matrix, y_aux_train.shape[-1])
    model.cuda()

    test_preds = train_model(model, train_dataset, test_dataset,
output_dim=y_train_torch.shape[-1],
        loss_fn=nn.BCEWithLogitsLoss(reduction='mean'))
    all_test_preds.append(test_preds)
    print()

#rnn with gru lstm

```

```

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout, GRU, Bidirectional
from keras.optimizers import SGD
import math
from sklearn.metrics import mean_squared_error

# Some functions to help out with
def plot_predictions(test,predicted):
    plt.plot(test, color='red',label='Real Stock Price')
    plt.plot(predicted, color='blue',label='Predicted Stock Price')
    plt.title('Stock Price Prediction')
    plt.xlabel('Time')
    plt.ylabel(' Stock Price')
    plt.legend()
    plt.show()

def return_rmse(test,predicted):
    rmse = math.sqrt(mean_squared_error(test, predicted))
    print("The root mean squared error is {}".format(rmse))

# Scaling the training set
sc = MinMaxScaler(feature_range=(0,1))
training_set_scaled = sc.fit_transform(training_set)

# Since LSTMs store long term memory state, we create a data structure with 60
timesteps and 1 output
# So for each element of training set, we have 60 previous training set elements
X_train = []
y_train = []
for i in range(60,2769):
    X_train.append(training_set_scaled[i-60:i,0])
    y_train.append(training_set_scaled[i,0])
X_train, y_train = np.array(X_train), np.array(y_train)

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

```

```

# The LSTM architecture
regressor = Sequential()
# First LSTM layer with Dropout regularisation
regressor.add(LSTM(units=50, return_sequences=True,
input_shape=(X_train.shape[1],1)))
regressor.add(Dropout(0.2))
# Second LSTM layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# Third LSTM layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# Fourth LSTM layer
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))
# The output layer
regressor.add(Dense(units=1))

# Compiling the RNN
regressor.compile(optimizer='rmsprop',loss='mean_squared_error')
# Fitting to the training set
regressor.fit(X_train,y_train,epochs=50,batch_size=32)

# Now to get the test set ready in a similar way as the training set.
# The following has been done so first 60 entires of test set have 60 previous
values which is impossible to get unless we take the whole
# 'High' attribute data for processing
dataset_total =
pd.concat((dataset["High"][:'2016'],dataset["High"]['2017:']),axis=0)
inputs = dataset_total[(len(dataset_total)-len(test_set) - 60):].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)

# Preparing X_test and predicting the prices
X_test = []
for i in range(60,311):
    X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)

# Visualizing the results for LSTM
plot_predictions(test_set,predicted_stock_price)

```

```

# Evaluating our model
return_rmse(test_set,predicted_stock_price)

# The GRU architecture
regressorGRU = Sequential()
# First GRU layer with Dropout regularisation
regressorGRU.add(GRU(units=50, return_sequences=True,
input_shape=(X_train.shape[1],1), activation='tanh'))
regressorGRU.add(Dropout(0.2))
# Second GRU layer
regressorGRU.add(GRU(units=50, return_sequences=True,
input_shape=(X_train.shape[1],1), activation='tanh'))
regressorGRU.add(Dropout(0.2))
# Third GRU layer
regressorGRU.add(GRU(units=50, return_sequences=True,
input_shape=(X_train.shape[1],1), activation='tanh'))
regressorGRU.add(Dropout(0.2))
# Fourth GRU layer
regressorGRU.add(GRU(units=50, activation='tanh'))
regressorGRU.add(Dropout(0.2))
# The output layer
regressorGRU.add(Dense(units=1))
# Compiling the RNN
regressorGRU.compile(optimizer=SGD(lr=0.01, decay=1e-7, momentum=0.9,
nesterov=False),loss='mean_squared_error')
# Fitting to the training set
regressorGRU.fit(X_train,y_train,epochs=50,batch_size=150)

# Preparing X_test and predicting the prices
X_test = []
for i in range(60,311):
    X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
GRU_predicted_stock_price = regressorGRU.predict(X_test)
GRU_predicted_stock_price = sc.inverse_transform(GRU_predicted_stock_price)

```

## Twitter module

```

import pandas as pd
import seaborn as sns
import re, nltk

```

```

nltk.download('punkt')
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn import model_selection, naive_bayes, svm
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import roc_auc_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from matplotlib import pyplot
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score, accuracy_score
import string
from nltk.corpus import stopwords
nltk.download('stopwords')
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB, GaussianNB
from sklearn.metrics import f1_score
plt.style.use('fivethirtyeight')
plt.style.use('dark_background')
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_score
import numpy as np
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from lime import lime_tabular
from tensorflow.keras.layers import Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.layers import LSTM, Bidirectional
from tensorflow.keras.layers import Dense, Dropout

df = pd.read_csv(r'../input/twitter/Tweets.csv')
df.head()

df['sentiment'].unique()

```

# I am tokenizing the tweet and also taking tokens from second index onwards as initial to gives airline name and '@' and lowering thm and later making it back a sentence

```
def clean_the_tweet(text):
    tokens = nltk.word_tokenize(re.sub("[^a-zA-Z]", " ", text))
    tokens = [token.lower() for token in tokens]
    return ' '.join(tokens[2:])

def text_process(msg):
    nopunc = [char for char in msg if char not in string.punctuation]
    nopunc = " ".join(nopunc)
    return ' '.join([word for word in nopunc.split() if word.lower() not in stopwords.words('english')])

def check_scores(clf, X_train, X_test, y_train, y_test):
    model = clf.fit(X_train, y_train)
    predicted_class = model.predict(X_test)
    predicted_class_train = model.predict(X_train)
    test_probs = model.predict_proba(X_test)
    test_probs = test_probs[:, 1]
    yhat = model.predict(X_test)
    lr_precision, lr_recall, _ = precision_recall_curve(y_test, test_probs)
    lr_f1, lr_auc = f1_score(y_test, yhat), auc(lr_recall, lr_precision)

    print("Train confusion matrix is: ", )
    print(confusion_matrix(y_train, predicted_class_train))

    print()
    print("Test confusion matrix is: ")
    print(confusion_matrix(y_test, predicted_class))
    print()
    print(classification_report(y_test, predicted_class))
    print()
    train_accuracy = accuracy_score(y_train, predicted_class_train)
    test_accuracy = accuracy_score(y_test, predicted_class)

    print("Train accuracy score: ", train_accuracy)
    print("Test accuracy score: ", test_accuracy)
    print()
    train_auc = roc_auc_score(y_train, clf.predict_proba(X_train)[:, 1])
    test_auc = roc_auc_score(y_test, clf.predict_proba(X_test)[:, 1])
```

```

print("Train ROC-AUC score: ", train_auc)
print("Test ROC-AUC score: ", test_auc)
fig, (ax1, ax2) = plt.subplots(1, 2)

ax1.plot(lr_recall, lr_precision)
ax1.set(xlabel="Recall", ylabel="Precision")

plt.subplots_adjust(left=0.5,
                    bottom=0.1,
                    right=1.5,
                    top=0.9,
                    wspace=0.4,
                    hspace=0.4)

print()
print('Are under Precision-Recall curve:', lr_f1)

fpr, tpr, _ = roc_curve(y_test, test_probs)

ax2.plot(fpr, tpr)
ax2.set(xlabel='False Positive Rate', ylabel='True Positive Rate')

print("Area under ROC-AUC:", lr_auc)
return train_accuracy, test_accuracy, train_auc, test_auc

def grid_search(model, parameters, X_train, Y_train):
    # Doing a grid
    grid = GridSearchCV(estimator=model,
                       param_grid=parameters,
                       cv=2, verbose=2, scoring='roc_auc')
    # Fitting the grid
    grid.fit(X_train, Y_train)
    print()
    print()
    # Best model found using grid search
    optimal_model = grid.best_estimator_
    print('Best parameters are: ')
    print(grid.best_params_)

    return optimal_model

# removing neutral tweets

df = df[df['sentiment']!= 'neutral']

```

```

df['cleaned_tweet'] = df['text'].apply(clean_the_tweet)

df.head()
df['sentiment'] = df['sentiment'].apply(lambda x: 1 if x == 'positive' else 0)
df.head()

# Cleaning the tweets, removing punctuation marks
df['cleaned_tweet'] = df['cleaned_tweet'].apply(text_process)
df.reset_index(drop=True, inplace = True)
df.head()

r_train_accuracy, r_test_accuracy, r_train_auc, r_test_auc =
check_scores(RandomForestClassifier(random_state=0).fit(x_train, y_train),
x_train,x_test,y_train,y_test)

```

## Interface module

### Req

```

configparser==5.2.0
db-dtypes==0.4.0
Flask==2.0.3
google-api-core==2.7.1
google-auth-oauthlib==0.5.1
google-auth==2.6.2
google-cloud-bigquery-storage==2.13.0
google-cloud-bigquery==2.34.2
google-cloud-core==2.2.3
google-cloud-secret-manager==2.9.2
google-cloud-storage==2.2.1
google-crc32c==1.3.0
google-resumable-media==2.3.2
googleapis-common-protos==1.56.0
grpc-google-iam-v1==0.12.3
grpcio-status==1.44.0
grpcio==1.44.0
gunicorn==20.1.0
numpy==1.22.2
oauthlib==3.2.0
pandas-gbq==0.17.4
pandas==1.4.1
psycopg2-binary==2.9.3
psycopg2==2.9.3
pyarrow==7.0.0
python-dateutil==2.8.2

```



```

PyYAML==6.0
requests==2.27.1
scipy==1.8.0
SQLAlchemy==1.4.32

from json import JSONDecodeError
import time
import sys
import os
sys.path.append('src/')
from src.global_crypto.etl_global_data_bq import global_crypto_main
from src.category_crypto.etl_category_data_bq import category_crypto_main
from src.defillama.etl_defillama_data_bq import defillama_main
from google.cloud import bigquery

## For CCloud SQL or PostGresVM
# from src.global_crypto.etl_global_data import global_crypto_main
# from src.category_crypto.etl_category_data import category_crypto_main
# from src.defillama.etl_defillama_data import defillama_main
from flask import Flask, Response
import requests

# For BQ
## Turn on this section for Cloud SQL or Postgres VM
app = Flask(__name__)
@app.route("/")
def main():
    project_id = "
    bq_dataset = "
    client = bigquery.Client()
    try:
        # print("Testing coingecko ping")
        # gecko_ping = requests.get('https://api.coingecko.com/api/v3/ping')
        # print(gecko_ping.json()['gecko_says'])
        # print('Coingecko is ok')

    print('Ingesting global crypto data')
    global_crypto_main(client, project_id, bq_dataset)
    print('Successfully ingested global crypto data')

    print('Ingesting category data')
    category_crypto_main(client, project_id, bq_dataset)
    print('Successfully category data')

```

```

print('Ingesting DefiLlama data')
defillama_main(client, project_id, bq_dataset)
print('Successfully ingested Defillama Data!')

except Exception as e:
    print('Something has failed, ensuring atomicity. Good bye')
    # to intentionally break the code for Cloud Scheduler to retry
    var1 = var2

return ('Done',200)

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=int(os.environ.get("PORT", 8080)))

## Turn on this section for Cloud SQL or Postgres VM
# app = Flask(__name__)
# @app.route("/")
# def main():
#     try:
#         print("Testing coingecko ping")
#         gecko_ping = requests.get('https://api.coingecko.com/api/v3/ping')
#         print(gecko_ping.json()['gecko_says'])
#         print('Coingecko is ok')

#         print('Ingesting global crypto data')
#         global_crypto_main()
#         print('Successfully ingested global crypto data')

#         print('Ingesting category data')
#         category_crypto_main()
#         print('Successfully category data')

#         print('Ingesting DefiLlama data')
#         defillama_main()
#         print('Successfully ingested Defillama Data!')

#     except Exception as e:
#         print('Something has failed, ensuring atomicity. Good bye')
#         # to intentionally break the code for Cloud Scheduler to retry
#         var1 = var2

#     return ('Done',200)

# if __name__ == '__main__':

```

```

# app.run(debug=True, host='0.0.0.0', port=int(os.environ.get("PORT", 8080)))

### For Local development
# def main():

# # Test if coingecko is working
# try:
#     gecko_ping = requests.get('https://api.coingecko.com/api/v3/ping')
#     print(gecko_ping.json()['gecko_says'])
# except Exception as e:
#     print(str(e))
#     print('Coingecko is down')

# # Get global crypto data
# try:
#     global_crypto_main()
# except Exception as e:
#     print(str(e))
#     print('Something wrong with global data coingecko')
# # except JSONDecodeError as je:
#     print("global data failed")
#     # print('JSONDecodeError, coingecko down')
#     pass

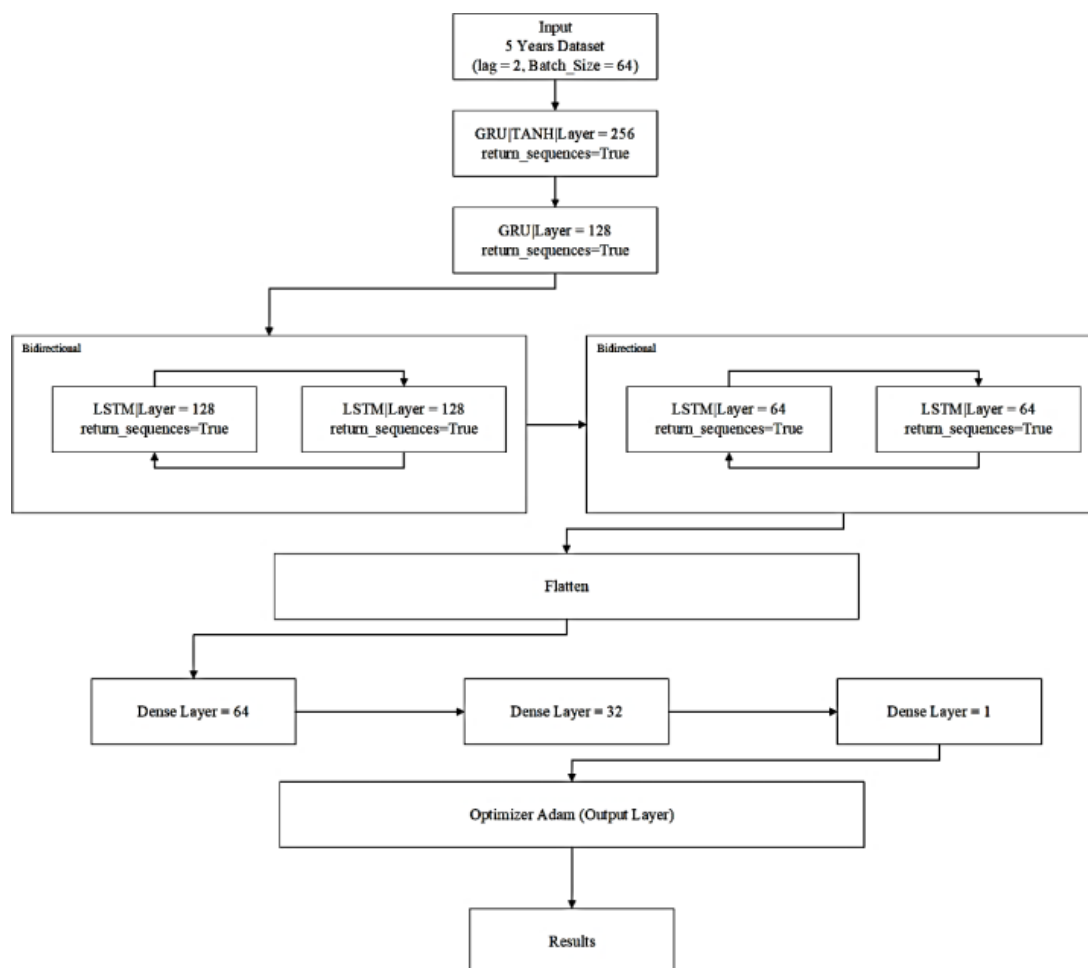
# # Get Cateogry Data
# try:
#     category_crypto_main()
# # except JSONDecodeError as je:
# except Exception as e:
#     print(str(e))
#     print('Something wrong with category data CMC')
#     # print('JSONDecodeError, CMC down')

# # Get Defi Llama Data
# try:
#     defillama_main()
# # except JSONDecodeError as je:
# except Exception as e:
#     print(str(e))
#     print('Something wrong with Defillama')
#     # print('JSONDecodeError, CMC down')

# if __name__ == '__main__':
#     main()

```

## ДОДАТОК Б. Архітектура RNN, запропонована в програмі [66]



ДОДАТОК В. Архітектура мережі для обробки текстових даних Twitter та визначення класу (позитивний/негативний твіт)

