

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.048:520.88

До захисту допущено
Завідувач кафедри ММСА
_____ О.Л.Тимощук
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра
за спеціальністю 122 Комп'ютерні науки
на тему: «Система прогнозування курсів криптовалют на основі
методів рекурентних нейронних мереж»

Виконав:
студент II курсу, групи КА-03 мп
Малік Тимур Імтіязович _____

Керівник:
доцент кафедри ММСА
д.т.н., доц.
Недашківська Н.І. _____

Рецензент:
доцент кафедри системного проектування
КПІ ім. Ігоря Сікорського, к.т.н.
Безносик О.Ю. _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань

Студент _____

Київ
2021

**Національний технічний університет
України «Київський політехнічний
інститут імені Ігоря Сікорського»**

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

Рівень вищої освіти – другий
(магістерський) Спеціальність – 122
«Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис)

(ініціали, прізвище)

«___» _____ 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Маліку Тимуру Імтіязовичу

- 1. Тема дисертації:** «Система прогнозування курсів криптовалют на основі методів рекурентних нейронних мереж», науковий керівник дисертації Недашківська Надія Іванівна, доктор технічних наук, затверджені наказом по університету від «02» листопада 2021 р. № 3651-с.
- 2. Термін подання студентом дисертації** 12.12.2021
- 3. Об'єкт дослідження:** криптовалютний ринок.
- 4. Предмет дослідження:** прогнозування цін криптовалют, методи машинного навчання, рекурентні нейронні мережі.
- 5. Перелік завдань, які потрібно зробити:**
 - 1) Здійснити огляд технічної літератури за темою роботи;
 - 2) Дослідити актуальність обраної теми;
 - 3) Ознайомитись із існуючими методами;
 - 4) Здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки;
 - 5) Дослідити сучасний стан криптовалютного ринку та його основні проблеми;
 - 6) Запропонувати методи прогнозування цін на криптовалютному ринку;

- 7) Розробити та реалізувати систему, що дозволяє прогнозувати ціну криптовалюти;
- 8) Провести аналіз результатів;
- 9) Провести аналіз ринкових можливостей запуску стартап проекту;
- 10) Розробити концептуальні висновки;
- 11) Підготувати ілюстративний матеріал;
- 12) Оформити пояснювальну записку.

6. Орієнтовний перелік ілюстративного матеріалу:

1. Постановка завдання дослідження;
2. Практичний приклад роботи програми, розробленої в ході дослідження;
3. Структура програмного продукту;
4. Приклад результатів використання системи;
5. Розрахункові таблиці з результатами.

7. Дата видачі завдання: 1 вересня 2021 року.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Отримання завдання на магістерську дисертацію	01.09.2021 – 05.09.2021	Виконано
2.	Огляд технічної літератури за темою	06.09.2021 – 12.09.2021	Виконано
3.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів. Дослідження актуальності обраної теми	13.09.2021 – 19.09.2021	Виконано
4.	Перший розділ. Аналіз існуючих методів прогнозування на основі цін криптовалютного ринку	20.09.2021 – 26.09.2021	Виконано
5.	Другий розділ. Математична постановка задачі. Вибір методів прогнозування	27.09.2021 – 03.10.2021	Виконано
6.	Третій розділ. Розробка системи підтримки прийняття рішень	04.10.2021 – 10.10.2021	Виконано
7.	Розширення функціоналу системи	11.10.2021 – 17.10.2021	Виконано
8.	Аналіз результатів	18.10.2021 – 24.10.2021	Виконано
9.	Проведення аналізу ринкових можливостей стартап – проекту	25.10.2021 – 31.10.2021	Виконано
10.	Підготовка ілюстративного матеріалу	01.11.2021 – 14.11.2021	Виконано
11.	Оформлення пояснювальної записки	15.11.2021 – 26.11.2021	Виконано

Студент _____

Т.І. Малік

Науковий керівник дисертації _____

Н.І. Недашківська

РЕФЕРАТ

Магістерська дисертація містить: 118с., 22 табл., 44 рисунки , 1 дод. та 44 джерела.

МГУА, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ ЦІНИ, КРИПТОВАЛЮТА.

Об'єктом дослідження є вибірка цін криптовалюти «Litecoin» за кожен день з 2017 по 2021 рік.

Предметом дослідження є методи інтелектуального аналізу даних на основі рекурентних нейронних мереж та метод МГУА. Програмною мовою була обрана Python.

Побудовано моделі прогнозування вартості криптовалюти «Litecoin» на основі останніх модифікацій рекурентних нейронних мереж LSTM та GRU, таких як двонаправлені (bidirectional) LSTM та GRU, розширені (extended) LSTM, залежні (dependent) Bi-RNN, вказаних моделей з додаванням механізму уваги (attention-based mechanism), методів машинного навчання, а також методу групового врахування аргументів (МГУА). Розроблено програмний продукт для прогнозування вартості криптовалют. Виконано навчання моделей рекурентних нейронних мереж. Проведено експерименти щодо підбору параметрів цих моделей. Використовуючи множину критеріїв, отримано оцінки якості побудованих моделей та отриманих прогнозів. Порівняно результати, отримані МГУА та моделями рекурентних нейронних мереж.

При виконанні роботи було встановлено два методи, що дають найкращі результати які достатньо близькі до реальних. Напрямок розвитку роботи є в розширенні функціоналу, зменшенні похибки прогнозування ціни та часу навчання моделі.

ABSTRACT

The master's dissertation contains: 118 pages, 22 tables, 44 drawings, 1 appendix. and 44 sources.

GMDH, RECURRENT NEURAL NETWORKS, INTELLECTUAL DATA ANALYSIS, MACHINE LEARNING, PRICE FORECASTING, CRYPTOCURRENCY.

The object of the study is a sample of prices of cryptocurrency "Litecoin" for each day from 2017 to 2021.

The subject of research is the methods of data mining based on recurrent neural networks and the method of GMDH. Python was chosen as the programming language.

Litecoin cryptocurrency forecasting models are built on the basis of the latest modifications of recurrent neural networks LSTM and GRU, such as bidirectional LSTM and GRU, extended LSTM, dependent (dependent) Bi-RNN, these models with the addition of attention mechanism attention-based mechanism), machine learning methods, and the method of group argumentation (GMDH). A software product for predicting the value of cryptocurrencies has been developed. Training of models of recurrent neural networks is executed. Experiments were conducted to select the parameters of these models. Using a set of criteria, estimates of the quality of the constructed models and the obtained forecasts are obtained. The results obtained by GMDH and models of recurrent neural networks are compared.

When performing the work, two methods were established that give the best results that are quite close to the real ones. The direction of development of work is in expansion of functionality, reduction of errors of forecasting of the price and time of training of model.

ЗМІСТ

РЕФЕРАТ	5
ВСТУП	9
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД КРИПТОВАЛЮТНОГО РИНКУ	12
1.1 Історія виникнення криптовалют	12
1.2 Перспективи та проблеми використання криптовалют	13
1.3 Вплив коронавірусу COVID-19 на криптовалютний ринок	14
1.4 Майнінг	15
1.5 Відомі криптовалюти	20
1.5.1 Ethereum (ETH)	20
1.5.2 Litecoin (LTC)	22
1.5.3 Cardano (ADA)	25
1.5.4 Polkadot (DOT)	27
1.5.5 Solana (SOL)	29
1.6 Економетричні моделі і методи прогнозування часових рядів	31
1.6.1 Авторегресійні моделі	31
1.6.2 Гетероскедастичні моделі	36
1.7 Постановка задачі дослідження	40
1.8 Висновки до першого розділу	40
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ	42
2.1 Історія нейронних мереж	42
2.2 Класифікація нейронних мереж	44
2.3 Архітектури нейронних мереж	45
2.3.1 Типи багат шарових нейронних мереж	46
2.3.2 Мережі із зворотними зв'язками	47
2.3.3 Формальний нейрон	48
2.4 Нейронні мережі прогнозування часових рядів	50
2.4.1 Рекурентні нейронні мережі	50
2.4.2 LSTM мережі	53

2.4.3	Різновиди LSTM мереж	56
2.4.4	Gated Recurrent Unit.....	57
2.4.5	Двонаправлені рекурентні нейронні мережі Bi-RNN та Bi-LSTM.....	58
2.4.6	Extended LSTM	61
2.4.7	Dependent Bi-RNN (DBRNN).....	64
2.4.8	Метод групового урахування аргументів	66
2.5	Висновки за розділом 2	69
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ЗА ДОПОМОГОЮ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ.....		70
3.1	Загальний огляд програмного продукту.....	70
3.2	Інтерфейс програми.....	71
3.3	Вхідні дані	71
3.4	Експерименти	72
3.5	Висновок до третього розділу	86
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ		87
4.1	Вступ та постановка задачі стартап проекту.....	87
4.2	Карта стартап проекту	89
4.3	Технологічний аудит ідеї проекту.....	92
4.4	Аналіз ринкових можливостей запуску стартап-проекту.....	95
4.5	Розроблення ринкової стратегії проекту	102
4.6	Висновки до 4 розділу	106
ВИСНОВОК.....		107
ПЕРЕЛІК ПОСИЛАНЬ.....		108
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ		113

ВСТУП

В даний час заробляти на інвестиціях в криптовалюту ризиковано. Крім того, скористатися цією валютою в побутових цілях занадто важко. Проте, ряд авторитетних експертів впевнений в блискучі перспективи такого роду грошей. Для початку, визначимо, що собою становить криптовалюта. Отже, криптовалюта є різновидом електронних грошей (перш за все, цифрових / віртуальних). Одиниця валюти є монета, яка захищена від підробки. Вона являє собою зашифровану інформацію, скопіювати яку неможливо. Кripto валюта емітується в мережі і не пов'язана зі звичайною валютою або державної валютної системою. Вперше термін «криптовалюта» стали використовувати після появи платіжної системи «Bitcoin», розробленої в 2009 році. До середини 2013 року програмне забезпечення практично всіх криптовалют базувалося на відкритому вихідному коді системи «Bitcoin». В липні 2013 року з'явилися нові платформи, які підтримують різну інфраструктуру - біржову торгівлю, магазини, та інше. До таких крипто платформам відносяться: BitShares, Mastercoin, NXT та інші платформи[1].

Кожна криптовалюта побудована за технологією блокчейн. Кripto валюти не мають центрального органу управління. Усі операції перевіряються мережею учасників, тобто іншими користувачами. Кожна операція комплектується з іншими для формування блоку, з яких складається безперервний ланцюг. Кожен блок має криптографічне посилання на попередника, що робить неможливим зміну інформації в одному блоці без необхідності внесення змін в усі наступні. Тому підробити або скасувати запис неможливо або дуже витратно[2]. Всі криптовалюти використовуються псевдонімно — зміст транзакцій відкритий, не шифрований, але в них є інформація лише про номери рахунків і немає інформації про власників цих рахунків. Проте, особу користувача може бути встановлено, якщо відома необхідна додаткова інформація. Ведуться розробки, де планується замінити

псевдонімність на анонімність[3]. Існують різні способи отримання криптовалют online. Нові (емітовані) кількості зазвичай розподіляються за встановленими процедурами, специфічними для кожної з криптовалют (Майнінг, форжинг, ICO). Майнінг і форжинг спрямовані на побудову блокчейну: творці нових блоків нагороджуються деякою кількістю емітованої криптовалюти і при цьому зазвичай не існує іншого шляху введення її в обіг. ICO є способом залучення фінансування через продаж партій нової криптовалюти, які спочатку були згенеровані і отримані організатором ICO.

Після початкового розподілу нової емісії інші бажаючі можуть отримати криптовалюта від тих, хто їй вже володіє — в обмін на звичайні гроші, або за надані товари або послуги, або в якості пожертвувань або як позику. Обмін можна проводити безпосередньо між зацікавленими особами без посередників або за допомогою будь-якої з численних майданчиків обміну цифрових валют. Станом на 2018 р. найбільшими майданчиками торгів є Coinbase, Bitfinex, BitStamp, Bittrex. Майданчики-лідери весь час змінюються в залежності від коливань ринку. Так свого часу лідерами по черзі були японські, китайські та корейські біржі. Користувачі можуть отримати криптовалюта тільки від тих, хто ними вже володіє: в обмін на звичайні гроші, на надані товари / послуги, або як пожертвування. Обмін можна проводити безпосередньо між зацікавленими особами без посередників, але на практиці зазвичай використовують будь-яку з численних майданчиків обміну цифрових валют. Котирування різних майданчиків в один і той же час можуть сильно відрізнятися. В даний час жоден не має відповідної реєстрації та біржової ліцензії. Таким чином, угоди на них, їхні клієнти не потрапляють під дію біржового законодавства країн [4].

В даний час нейронні мережі широко використовуються в різних галузях. Також вони часто застосовуються при прогнозуванні.

У даній роботі проаналізовано типи нейронних мереж та виконано прогнозування курсів криптовалют, за допомогою рекурентних нейронних мереж та алгоритму МГУА.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД КРИПТОВАЛЮТНОГО РИНКУ

1.1 Історія виникнення криптовалют

Криптографія з метою конфіденційних платежів почала використовуватися з 1990 року в системі «DigiCash» Девіда Чома, компанія якого збанкрутувала в 1998 році. Однак, його платіжна система була централізованою, а вперше термін «криптовалюта» почав використовуватися після появи пірингової платіжної системи «Bitcoin» і використовує хешування SHA-256 і систему proof-of-work.

З тих пір як в світі з'явився «Bitcoin», почалася історія криптовалюта, яка до сьогодні стала дуже багатою. Адже цифрові монети і криптовалютні проекти знайшли застосування в багатьох сферах людської діяльності, і для оцінки їх перспективності досить звичайного аналізу причин їх появи і процесу розвитку. Першою і найбільш успішною криптовалюта вважається «Bitcoin» - саме після її успіху на ринку з'явилося безліч інших проектів, аналогічних грошових систем.

До виникнення криптовалюти багато недоліків банківських і електронних платіжних систем ще не були виявлені, адже з можливостями децентралізованих систем їх ще не порівнювали. Але вже були припущення, що альтернативна платіжна система мала б більше переваг, тому ринок б її прийняв і в разі її децентралізованого пристрою. Серед достоїнств передбачуваної платіжної системи в обов'язковому порядку були відсутність плати за транзакції, можливість використання кожною людиною і повна анонімність переказів. Тобто децентралізована платіжна система дала б користувачам нечувану раніше свободу, в обхід умов світових фінансових інститутів і банків.

Який не візьми геніальний винахід в історії людства, перед його появою на світ йшли роки дослідів і досліджень, а саме перед запуском мережі «Bitcoin» протягом сорока років програмісти працювали над вдосконаленням

методів шифрування і захищених режимів передачі даних. Ці розробки не торкалися безпосередньо «Bitcoin», але були реалізовані в більш ранніх проектах, в тому числі «Hashcash» і «b-money», якими займався Хел Фінні з компанії IBM.

1.2 Перспективи та проблеми використання криптовалют

Вивчивши властивості і особливості криптовалюта можна визначити деякі її переваги і недоліки в порівнянні зі звичайними видами валют. Серед переваг можна відзначити такі як: відкритий код алгоритму, що дозволяє добувати її кожному; не схильність інфляції; її не можна скопіювати, так як вона являє собою набір зашифрованої інформації. Як недоліки можна відзначити наступне: відсутність регулюючих механізмів; висока волатильність у зв'язку зі специфікою її використання; непрацездатність електронного крипто гаманця може привести до втрати всіх крипто монет, що знаходяться в ньому.

Слід зазначити, що про юридичний статус і економічному сенсі криптовалюта ведуться численні дискусії. Залежно від країни криптовалюту можуть розглядатися, як платіжний засіб або специфічний товар. Криптовалюти можуть мати обмеження в обороті такі як: заборона операцій з ними для кредитних організацій. Комерційні банки, податкові, судові та інші державні органи не можуть впливати на транзакції будь-яких учасників платіжної системи [6].

У ряді країн офіційно дозволені операції з криптовалюта. У Німеччині «Bitcoin» визнаються в якості розрахункової грошової одиниці, в Японії - Закон засобом платежу, в інших країнах операції з «Bitcoin» заборонені для банків, при цьому дозволені для фізичних осіб. В Швейцарії на криптовалюту діють такі ж правила, як і з іноземними валютами. У Росії ніяких обмежень

на використання «Bitcoin» немає. При цьому, негативний ставлення до криптовалют неодноразово висловлювали багато представників органів державної влади та інші. Також деякий відсоток людей вважає, що цифрові гроші за своєю суттю близькі до фінансових пірамід.

1.3 Вплив коронавірусу COVID-19 на криптовалютний ринок

Епідемія коронавірусу зачепила всі сфери життя в усьому світі. У зв'язку з карантином практично зупинилася світова економіка, і коли вона повністю відновиться, поки прогнозувати не береться ніхто. Карантин підштовхнув держави прискоритися в створенні нових платіжних інструментів, які не мали б фізичної форми. Всі великі країн відчуло важливість впровадження електронних платіжних засобів. У період карантину, коли банкноти стали таким же способом передачі інфекції, як і будь-які інші матеріали, керівництво держав забило на сполох і вирішило масово популяризувати безконтактні цифрові платежі. Це видно на прикладі Китаю, який в разі прискорив запуск своєї національної цифрової валюти. Зараз можливість запуску CBDC розглядають Туреччина, Німеччина, Франція і деякі інші країни. Крах фондового ринку змусив інституційних інвесторів переглянути свої підходи до інвестування і звернути увагу на криптовалюта. Відомо, що Китай знищив кілька мільйонів фіатних юанів просто тому, що банкноти вступали в контакт з людьми, зараженими коронавірусом. У випадку з криптовалюта це неможливо. Тому Піднебесна прискорила темпи створення власної криптовалюта, і вже в травні співробітники міських адміністрацій деяких провінцій отримали зарплати в цифрових валютах. Також було помітно підвищення інтересу до криптовалюта з боку користувачів. Тільки на одній крипто біржі в Росії зафіксовано рекордну кількість унікальних користувачів. Крім того, середній

розмір денного обороту на людину безперервно збільшується з лютого 2020 року, в той час як протягом усього 2019 року його практично не змінювався. Але є і негативні сторони. У березні, після краху фондового і сировинного ринків, обрушився і ринок криптовалют. Деякі проекти, особливо в сфері DeFi, не змогли пережити кризи і були змушені піти з ринку. Деякі інвестори втратили значну частину прибутку після падіння курсу криптовалют. Волатильність ринку, яка і до пандемії була досить високою, ще більше збільшилася. Все це негативно впливає на ринок крипто валют. Тому дати однозначну оцінку впливу пандемії коронавіруса на ринок поки неможливо[7].

1.4 Майнінг

Майнінг біткойнів — це процес, за допомогою якого нові біткойни вводяться в обіг; це також спосіб підтвердження нових транзакцій мережею і критична складова підтримки та розвитку реєстру блокчейн. «Майнінг» виконується за допомогою складного апаратного забезпечення, яке вирішує надзвичайно складну обчислювальну математичну задачу. Перший комп'ютер, який знайшов рішення проблеми, отримує наступний блок біткойнів, і процес починається знову.

Майнінг криптовалюти — це копіткий, дорогий і лише спорадично приносить користь. Тим не менш, майнінг має магнітну привабливість для багатьох інвесторів, які цікавляться криптовалютою, оскільки майнери винагороджуються за свою роботу криптотокенами. Можливо, це тому, що підприємці сприймають видобуток як копійки з неба, як каліфорнійські золотошукачі в 1849 році.

Основною перевагою для багатьох майнінгів є перспектива отримати винагороду біткойнами. Тим не менш, вам, звичайно, не обов'язково бути

майнером, щоб володіти токенами криптовалюти. Ви також можете купувати криптовалюту за допомогою фіатної валюти; ви можете торгувати ним на біржі, використовуючи іншу криптовалюту (як приклад, використовуючи Ethereum або NEO для покупки біткойн); ви навіть можете заробити це, роблячи покупки, публікуючи дописи в блозі на платформах, які платять користувачам криптовалютою, або навіть створюючи крипто-рахунки, які приносять відсотки.

Прикладом платформи крипто-блогів є Steemit, який схожий на Medium, за винятком того, що користувачі можуть винагороджувати блогерів, сплачуючи їм у власній криптовалюті під назвою STEEM. Потім STEEM можна обміняти в іншому місці на біткойн.

Винагорода за біткойн, яку отримують майнери, є стимулом, який мотивує людей допомагати в головній цілі майнінгу: легітимізувати та контролювати транзакції біткойн, забезпечуючи їх дійсність. Оскільки ці обов'язки розподілені між багатьма користувачами по всьому світу, біткойн є «децентралізованою» криптовалютою або такою, яка не покладається на центральний орган влади, як-от центральний банк чи уряд, для контролю за своїм регулюванням.

Окрім наповнення кишені майнерів та підтримки екосистеми біткойн, майнінг виконує ще одну життєво важливу мету: це єдиний спосіб випустити нову криптовалюту в обіг. Іншими словами, майнери в основному «карбують» валюту. Наприклад, станом на вересень 2021 року в обігу перебувало близько 18,82 мільйона біткойнів із загальної кількості 21 мільйона[33].

Окрім монет, викарбуваних за допомогою блоку genesis (самий перший блок, який був створений засновником Сатоші Накамото), кожен із цих біткойнів з'явився завдяки майнерам. За відсутності майнерів біткойн як мережа все ще існував би і був би придатним для використання, але ніколи б не було додаткових біткойнів. Однак, оскільки швидкість «видобутку» біткойнів з часом зменшується, остаточний біткойн не буде

розповсюджуватися приблизно до 2140 року. Це не означає, що транзакції перестануть перевірятися. Майнери продовжуватимуть перевіряти транзакції та отримуватимуть плату за це, щоб зберегти цілісність мережі біткойн[34].

Винагорода за майнінг біткойн зменшується вдвічі приблизно кожні чотири роки. Коли біткойн був вперше видобутий у 2009 році, видобуток одного блоку давав 50 BTC. У 2012 році цей показник зменшився вдвічі до 25 BTC. До 2016 року цей показник знову зменшився вдвічі до 12,5 BTC. 11 травня 2020 року винагорода знову зменшилася вдвічі до 6,25 BTC (Рисунок 1.1).

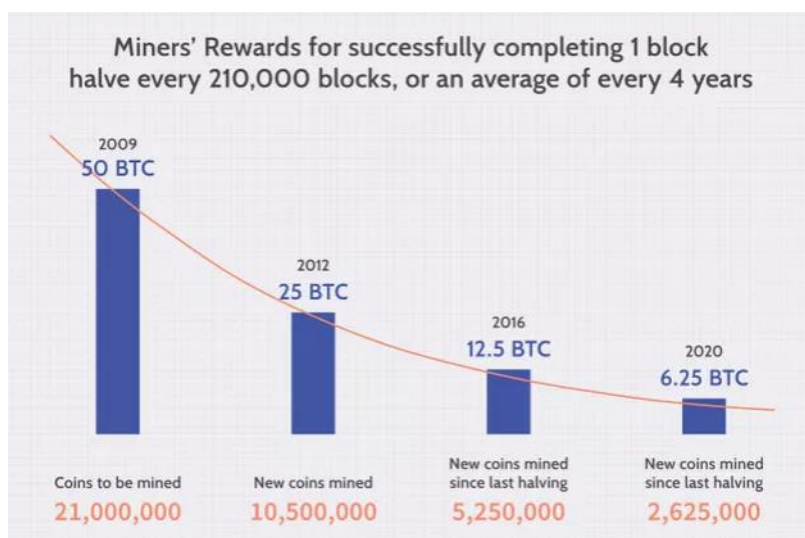


Рисунок 1.1 – Графік зміни винагороди за майнінг [38]

Якщо ви хочете відстежувати, коли саме відбудеться ці уполовинення, ви можете звернутися до Bitcoin Clock , який оновлює цю інформацію в режимі реального часу. Цікаво, що ринкова ціна біткойна протягом всієї його історії, як правило, майже відповідала скороченню нових монет, що вводяться в обіг. Це зниження рівня інфляції збільшило дефіцит, і історично разом з ним зростала ціна.

Хоча на початку історії біткойна люди могли змагатися за блоки зі звичайним домашнім персональним комп'ютером, це вже не так. Причина цього полягає в тому, що складність видобутку біткойн змінюється з часом.

Щоб забезпечити безперебійне функціонування блокчейну та його здатність обробляти та перевіряти транзакції, мережа біткойн прагне створювати один блок кожні 10 хвилин або близько того. Однак, якщо один

мільйон майнінгових установок змагаються за вирішення проблеми хешування, вони, швидше за все, знайдуть рішення швидше, ніж сценарій, за яким 10 майнінгових установок працюють над однією проблемою. З цієї причини біткойн призначений для оцінки та регулювання складності майнінгу кожні 2016 блоків або приблизно кожні два тижні.

Коли для майнінгу біткойнів працює більше обчислювальної потужності, рівень складності майнінгу підвищується, щоб підтримувати виробництво блоків на стабільному рівні. Менша обчислювальна потужність означає зниження рівня складності. При сьогоdnішньому розмірі мережі персональний комп'ютер, який видобуває біткойн, майже напевно нічого не знайде.

Все це означає, що для конкурентоспроможного майнінгу майнерам тепер необхідно інвестувати в потужне комп'ютерне обладнання, таке як графічний процесор (блок графічного процесора) або, що більш реалістично, інтегральна схема (ASIC) для конкретного застосування. Ціна може становити від 500 доларів до десятків тисяч. Деякі майнери, зокрема Ethereum, купують окремі відеокарти (GPU) як недорогий спосіб об'єднати операції з майнінгу.

Винагорода за майнінг виплачується майнеру, який першим знайде рішення головоломки, і ймовірність того, що учасник знайде рішення, дорівнює частці загальної потужності майнінгу в мережі.

Учасники з невеликим відсотком майнінгової потужності мають дуже малий шанс самостійно знайти наступний блок. Наприклад, карта для майнінгу, яку можна було придбати за пару тисяч доларів, становила б менше 0,001% потужності майнінгу в мережі. З таким невеликим шансом знайти наступний блок може пройти багато часу, перш ніж майнер знайде блок, а труднощі підйому ще більше погіршать ситуацію. Майнер може ніколи не відшкодувати свої інвестиції. Відповіддю на цю проблему є майнінгові пули.

Майнінг-пули управляються третіми сторонами та координують групи майнерів. Працюючи разом в пулі та розподіляючи виплати між усіма

учасниками, майнери можуть отримувати постійний потік біткойнів, починаючи з дня активації своїх майнерів.

Ризики майнінгу часто є ризиками фінансового та регуляторного ризику. Як уже згадувалося, видобуток біткойнів та майнінг загалом є фінансовим ризиком, оскільки можна докласти всіх зусиль, щоб придбати обладнання для майнінгу на сотні або тисячі доларів, лише щоб не отримати віддачу від своїх інвестицій. Тим не менш, цей ризик можна зменшити, приєднавшись до майнінгових пулів. Якщо ви думаєте про майнінг і живете в районі, де це заборонено, вам слід переглянути. Також може бути гарною ідеєю дослідити законодавство у вашій країні та загальні настрої щодо криптовалюти, перш ніж інвестувати в обладнання для майнінгу[38].

Додатковим потенційним ризиком від зростання майнінгу біткойнів (а також інших систем підтвердження роботи) є збільшення споживання енергії, що вимагається комп'ютерними системами, на яких працюють алгоритми майнінгу. У той час як ефективність мікročіпів для чіпів ASIC різко зросла, зростання самої мережі випереджає технологічний прогрес[37]. Як наслідок, є занепокоєння щодо впливу на навколишнє середовище та вуглецевого сліду видобутку біткойнів[35].

Оскільки це повністю цифрові записи, існує ризик копіювання, підробки або подвійного витрачання однієї і тієї ж монети кілька разів. Майнінг вирішує ці проблеми, роблячи надзвичайно дорогими та ресурсомісткими спроби зробити одну з цих речей або іншим чином «зламати» мережу. Справді, набагато вигідніше приєднатися до мережі як майнер, ніж намагатися її підірвати.

Окрім введення в обіг нових BTC, майнінг виконує вирішальну роль у підтвердженні та перевірці нових транзакцій на блокчейні Bitcoin. Це важливо, оскільки немає центрального органу, такого як банк, суд, уряд чи будь-який інший, який визначає, які операції є дійсними, а які ні. Натомість процес майнінгу досягає децентралізованого консенсусу за допомогою підтвердження роботи (PoW).

На початку існування біткойн будь-хто міг просто запустити програму майнінгу зі свого ПК або ноутбука. Але в міру того, як мережа збільшувалася і більше людей зацікавилася майнінгом, складність алгоритму майнінгу ставала складнішою. Це пов'язано з тим, що код для біткойн націлений на пошук нового блоку в середньому кожні десять хвилин. Якщо задіяно більше майнерів, шанси на те, що хтось швидше вирішить правильний хеш, збільшуються, і, таким чином, підвищується складність для відновлення цієї 10-хвилинної мети. А тепер уявіть, якби до мережі приєдналися ще тисячі чи навіть мільйони потужностей майнінгу. Це багато нових машин, які споживають енергію.

Законність майнінгу біткойн повністю залежить від вашого географічного розташування. Концепція біткойна може поставити під загрозу домінування фіатних валют і державний контроль над фінансовими ринками. З цієї причини біткойн в деяких місцях абсолютно незаконний.

Власність і майнінг біткойн є законними в багатьох країнах. Декількома прикладами місць, де це було незаконно, згідно зі звітом за 2018 рік, були Алжир, Єгипет, Марокко, Болівія, Еквадор, Непал і Пакистан[36].

Загалом використання та майнінг біткойн залишається законним у більшій частині земної кулі.

1.5 Відомі криптовалюти

1.5.1 Ethereum (ETH)

Кожному героєві потрібен суперник. У випадку з біткоїном це Ethereum (ETH). В українській мові отримало назву ефір, або ефіріум. Ефіром у Стародавній Греції називали верхній шар атмосфери, а хімії так позначають деякі класи хімічних сполук. Ethereum - це форма цифрових грошей, яка використовується в основному на платформі Ethereum. Ефір можна досить

швидко надіслати знайомим через інтернет - зазвичай за набагато меншу плату, ніж у інших альткоїнів.

Основна мета ефіру - не тільки транзакції цифрових активів, а й підтримка роботи блокчейну Ethereum. Ця мережа є основою безлічі децентралізованих додатків, які називають одним загальним терміном — DApps. DApps служать для того, щоб створювати ефективніші альтернативи технологіям, які ми використовуємо кожен день. Щоразу, коли в блокчейні Ethereum здійснюються транзакції та закриваються операції, з користувача стягується комісія в криптовалюті ефір.

Другу за популярністю криптовалюту створив канадський програміст російського походження Віталік Бутерін. Він дізнався про криптовалюту та блокчейни приблизно через два роки після запуску біткоїну. Бутерін залучив кількох інших фахівців як співзасновників власної валюти, і влітку 2014 відбувся перший продаж tokenів ETH. У результаті зібрано близько \$18 млн.

У 2016 році розпочалося перше значне зростання курсу. Ціна на ETH за півроку злетіла приблизно з \$0,95 (за даними на 1 січня) до \$20,59 долара (16 червня) — зростання склало близько 2060%. Пізніше курс ETH скоригувався і монета торгувалася на рівні близько \$7-8 до кінця року. Ніхто з інвесторів не підозрював, що криптовалютний бум не за горами.

Починаючи з 2017 року ефір почав свою переможну ходу вгору зі скромного рівня \$8,33. 20 грудня 1 ETH коштував \$868 доларів – на 10 320% більше, ніж на початку року. Але це був далеко не кінець. Після короткого спуску до \$675, ETH практично подвоїв свою вартість і зріс до \$1 432,88 до 13 січня 2018 року. Після цього криптовалютний ажіотаж скінчився, і до грудня 2018 ETH вже торгувався нижче \$100. Проте були й періоди, коли ефір знову торгувався вищим за \$200. У червні 2019 року альткоїн навіть торгувався вище за \$320. Багато в чому це було з тим, деякі великі компанії, зокрема Microsoft, Amazon і JPMorgan, почали розробляти проекти у мережі Ethereum.

У 2020 році почалося нове піднесення ринку. 11 травня 2021 криптовалюта ефір досягла свого піку вартості - \$ 4 168. Після цього ринок знову почав просідати. Зараз ETH торгується близько \$2 180[39].

Ethereum торгується на усіх основних біржах, оскільки це друга за величиною криптовалюта у світі. Як і у випадку з біткоїном, їм можна розплачуватися за товари та послуги в деяких країнах.

Особливої ролі блокчейн ефіру набув децентралізованих фінансах (DeFi). Він вважається основним блокчейном для додатків у цій сфері. Однак постійно з'являються проекти, які хочуть зайняти свою нішу у DeFi. Наприклад, платформа Polkadot, яку вже прозвали "вбивцею ефіру".

Зараз у криптосфері очікують оновлення ефіру до версії 2.0 та перехід на новий алгоритм роботи Proof-of-Stake (PoS). Коли це станеться, ціна монети поповзе вгору. Очікується, що до кінця року вона підніметься приблизно до \$3 600.

Є, звісно, й більш неймовірні прогнози. Так, роздрібний інвестор Нікіл Шамапан припустив, що ETH зможе досягти вартості \$150 тис. до січня 2023 року. З ним схильний погодитися колишній генеральний директор BitMEX Apture Хеус. На його думку, імовірність досягнення цього події становить 70%.

У той же час треба пам'ятати, що вартість криптовалют залежить від регулювання сфери в тій чи іншій країні. Наприклад, нещодавнє рішення влади Китаю заборонити використання криптовалюту значно вплинуло і на ціни, і на майнінг[39].

1.5.2 Litecoin (LTC)

Лайткоїн (LTC) – криптографічна віртуальна (електронна) монета, в основі якої лежить програмне забезпечення Bitcoin та аналогічні з ним

принципи (анонімність, безпека, висока швидкість транзакцій). Це другий форк Bitcoin після Namecoin, що має незначні відмінності, порівняно з попередником.

Ця криптовалюта розроблена колишнім працівником Google Чарльзом Лі у 2011 році.

За всієї зовнішньої та технологічної схожості називати Litecoin безпосередньо клоном головної криптовалюти світу не можна. Він має абсолютно інший алгоритм роботи, що дозволяє системі активніше взаємодіяти з пам'яттю, що робить майнінг цієї монети простіше.

Особливість форка полягає у відкритості вихідного коду, що дозволяє змінювати або копіювати програмне забезпечення всім бажаючим.

До жовтня 2013 року один LTC коштував від \$1,7 до \$2. Проте суттєве зростання наприкінці 2013 р. призвело до подорожчання монети до \$42, а капіталізація нової криптовалюти досягла позначки \$1 млрд. Саме тоді її поспішили оголосити спадкоємицею Bitcoin. Після ажіотажного зростання стався спад.

Нове ралі Litecoin стартувало у 2017 р., коли її курс пробив позначку \$350. Капіталізація Litecoin перевищила \$3 млрд, що забезпечує токenuсьоме місце у рейтингу криптовалют.

За місяць з жовтня по листопад 2020 р. на тлі постійного зростання біткоїну курс лайткоїну піднявся з \$60 до \$85. Капіталізація монети зросла майже вдвічі і зараз становить понад \$5,7 млрд.

Почнемо з децентралізації. Завдяки відсутності єдиного центру управління учасники платформ отримують однакові права та можливості. Це дозволяє системі не залежати від впливу країн і підприємств.

Біткоїн і лайткоїн мають ще одну специфічну рису — це груповий майнінг. Майнеться Litecoin так само, як і біткоїн. Як правило, майнери об'єднуються в пули, що дозволяє зробити процес видобутку криптовалюти дешевшим і ефективнішим у всіх сенсах.

Одна з основних відмінностей Bitcoin та Litecoin стосується загальної кількості монет, які може зробити кожна криптовалюта.

Bitcoin пропонує своїм користувачам максимум 21 млн. BTC, а Litecoin – 84 млн. LTC. Кількість монет у системі Litecoin у чотири рази більша, як і швидкість обробки блоків. Експерти прогнозують генерацію лише 75% від загальної кількості до кінця 2020 року.

Досить важлива відмінність між Bitcoin і Litecoin полягає в алгоритмі хешування, який кожен використовує для вирішення блоку, а також кількості монет, що розподіляються кожен раз, коли рішення знайдено.

Біткоїн використовує алгоритм SHA-256, який вважається більш складним. У Litecoin замість SHA256 для майнінгу вшитий Scrypt, розроблений офіцером безпеки FreeBSD Коліном Персівалем. Спочатку цей алгоритм створювався для онлайн-сервісу Tarsnap, але й криптовалюти підійшов ідеально.

Використання Scrypt дозволяє Litecoin швидше взаємодіяти з оперативною пам'яттю. Це значно прискорює майнінг. Кожен блок в середньому обробляється за 2,5 хвилини, а Bitcoin на це потрібно в 4 рази більше часу. Це дозволяє швидше генерувати цифрову валюту.

Початкова мета використання Scrypt полягала у тому, щоб дозволити майнерам одночасно добувати як Bitcoin, і Litecoin. Вибір використання алгоритму частково полягав і в тому, щоб не надавати значних переваг майнерів пулам з GPU, FPGA і ASIC над майнерами по CPU.

Насправді виявилось, що FPGA та ASIC-пристрої для майнінгу Litecoin з оригінальним алгоритмом виходять не тільки складніше, а й дорожче у виробництві, ніж аналогічні пристрої для видобутку Bitcoin, який використовує SHA-256.

З іншого боку, завдяки алгоритму Scrypt система може ефективніше взаємодіяти з пам'яттю. Швидкість обробки блоків тут у чотири рази вища, ніж у Bitcoin, що помітно прискорює майнінг, а водночас і поява нових монет у системі[40].

1.5.3 Cardano (ADA)

Cardano (ADA) – це блокчейн-платформа, створена за співпраці компанії Input Output Hong Kong (IOHK) та Чарльза Хоскінсона, колишнього співзасновника BitShares, Ethereum та Ethereum Classic. Із самого початку позиціонується як альтернатива Ethereum.

Платформа спеціалізується на запуску смарт-контрактів, децентралізованих додатків, сайдчейнів та багатопартійних (безпечних) обчислень.

Внутрішня криптовалюта Cardano називається ADA. Платформа названа на честь італійського математика Джероламо Кардано, а криптовалюта на честь англійського математика Ади Лавлейс.

Точкою відліку історії нової криптовалютної платформи стала пропозиція японських інвесторів, які зацікавилися ідеєю використання блокчейн-технології для смарт-контрактів та уклали контракт із Чарльзом Хоскінсом на розробку нової криптовалюти з високою масштабованістю.

У ході з ICO-компанії з 2015 по 2017 роки команді проекту вдалося зібрати \$62 млн. Буквально через тиждень після запуску ринкова капіталізація Cardano зросла до \$500 млн, а добовий оборот ADA на торгових майданчиках досяг \$100 млн, настільки високий був інтерес блокчейн-спільноти до нової монети.

Найбільші суми капіталовкладень надходили із країн Південно-Східної Азії. Криптовалюта популярна в Японії, Південній Кореї та Таїланді.

Cardano пишеться мовою Haskell, мовою програмування з високим ступенем відмовостійкості. Застосований протокол Proof-of-Stake Ouroboros передбачає участь у голосуванні всіх власників криптовалюти. Він схожий на новий протокол Ethereum Casper, але, за заявою авторів Ouroboros, є єдиним алгоритмом консенсусу, безпеку якого можна довести за допомогою математики.

У tokenів ADA обмежена емісія. Заплановано випуск 45 млрд. монет. Вже згенеровано близько 30 млрд. Приблизно 10% їх віддали компаніям, які проектуєть цю платформу і займаються її розробкою. Інші розподілені серед власників-майнерів. Емісія повністю залежить від нових блоків. Сьогодні після підписання кожного блоку виробляється 2000 монет. Далі показники поступово зменшуватимуться. Завершити випуск всього обсягу монет планують у 2023 році.

Криптовалюта Cardano (ADA) використовує власний блокчейн, званий Cardano Settlement Layer (CSL). CSL – це шар розподіленого реєстру транзакцій для підтримки операцій із криптовалютами гаманцями. Другий шар, званий Cardano Computation Layer (CCL), підтримуватиме смарт-контракти та децентралізовані програми. Така багаторівнева архітектура дозволяє спростити оновлення протоколу та позитивно впливає на прогноз Cardano.

Фінансові державні регулятори зазвичай скептично ставляться до використання анонімних криптовалют у традиційній фінансовій системі. Для подолання цієї недовіри платформа Cardano може використовувати в блокчейне метадані, захищені криптографічним методом, що дозволить взаємодіяти із зовнішніми регуляторами, зберігши достатньо конфіденційність власників криптовалюти.

Один із суттєвих плюсів Cardano – система безпеки Ouroboros. Саме вона забезпечує математичну точність у роботі. За допомогою цього алгоритму вдалося серйозно скоротити енерговитрати під час перевірки блоків. На користь цієї криптовалюти свідчить спеціальна захищена мова програмування.

Крім того, криптовалюта Cardano (ADA) відрізняється швидкими транзакціями та можливістю роботи з цифровими активами. Будь-який користувач може створити та оформити смарт-контракт, що убезпечить його грошові перекази та інші операції.

Певний мінус Cardano – блокчейн все ще перебуває на стадії розробки. Він повинен вирішити свої проблеми масштабованості, оскільки здатний обробляти лише 257 транзакцій за секунду (TPS)[41].

1.5.4 Polkadot (DOT)

Головна задумка проекту Polkadot – підвищити сумісність різних блокчейнів усередині однієї платформи. Проект навіть називали "вбивцею Ethereum". Компанія швидко розвивається. 2020 року вона запустила «полігон» для перевірки кодів — Kusama. Пристрій мережі досить складний, але ми спробуємо пояснити його простими словами.

Перше, що треба сказати про Polkadot - це спочатку платформа для блокчейна, і лише потім криптовалюта.

Polkadot виникла через складнощі в масштабуванні Ethereum. Її створив Гевін Вуд – співзасновник Ethereum, який покинув проект у 2016 році.

На офіційному сайті Polkadot йдеться, що їхня місія — створити повністю децентралізовану мережу, контролювати яку самі користувачі, а не регулюючі державні органи.

Polkadot складається з трьох основних компонентів:

- "Серце" мережі - релейний ланцюг (Relay Chain). Вона допомагає досягти консенсусу та загальної безпеки в мережі різних ланцюжків. Компанія на своєму сайті докладно розповідала, як працює консенсус у її мережі.

- На платформі Polkadot створюються блокчейни та з'єднуються в одну мережу. Ці ланцюжки називаються парачейном. Вони формують єдиний простір із синхронізованих та безпечних мереж. Парачейни можуть мати свої токени, які оптимізовані під потреби мережі.

– Нарешті, мости (Bridge Chain) з'єднують блокчейни, які працюють на Polkadot. Таким чином вони можуть взаємодіяти, наприклад, з мережею Ethereum.

Polkadot має свою експериментальну мережу Kusama, яку в компанії називають «канарієчною». У ній тестують і перевіряють коди, перш ніж вони потраплять до блокчейну основної мережі.

«Канаркова» мережа, як і сама Polkadot, працюють на двох блокчейнах: релейній мережі та парачейні. Перша завершує мережеві транзакції, а другий підтверджує їхню точність. Щоб запустити проект у релейній мережі, необхідно виграти слот на аукціоні.

Нещодавно Kusama провела перший слот-аукціон парачейну, підсумки підбили 22 червня. Переможцем стала мережа Karura. Її підтримали понад півтори тисячі користувачів, вклавши 500 тисяч токенів Kusama (KSM). На той момент це понад \$90 млн. Karura прагне реалізувати протокол Acala, який, за задумом, стане децентралізованим фінансовим центром у DeFi-екосистемі у мережі Polkadot.

Найближчим часом планується провести ще кілька слот-аукціонів. За його підсумками, компанія проведе аудит мережі, щоб перевірити, як мережа працює в процесі впровадження протоколів. Потім відбудеться ще один раунд аукціонів.

Модель з експериментальною мережею забезпечує безпеку та якість кодів, які потрапляють до Polkadot. Тому що спочатку, тестуючись у Кусама, він несе в собі підвищені ризики[42].

Токени Polkadot (DOT) та Kusama (KSM) виконують три функції:

- Використовуються всередині мережі для голосування та участі в аукціонах. Власники токенів повністю контролюють протокол.
- Власники DOT своєю активною участю в мережі та стейкінгом токенів забезпечують функціонування базового механізму консенсусу.
- Токени видаються як винагорода активним учасникам мережі.

1.5.5 Solana (SOL)

Solana – це просунутий блокчейн-проект із відкритим вихідним кодом, який прагне використовувати кілька проривних технологій для наступного покоління DApps. Він має максимальну пропускну здатність більше 50 000 TPS і час блоку лише 400 мс.

Проект спрямований на створення масштабованої, безпечної та максимально децентралізованої платформи, яка може підтримувати потенційно тисячі вузлів без шкоди для пропускну здатності, допомагаючи уникнути деяких проблем із якими стикаються конкуруючі системи.

Проект був заснований у 2017 році під час буму ICO та зібрав понад 25 мільйонів доларів у різних раундах приватних та публічних продажів. Платформа запустила mainnet у березні 2020 року, але все ще працює як бета-версія.

Однією з ключових відмінних рис Solana є її система консенсусу Proof of Stake (PoS), яка підкріплюється чимось, названим Tower Consensus. Це варіант системи, відомий як Практична Візантійська Відмовостійкість (PBFT), він дозволяє розподіленим мережам досягати консенсусу, незважаючи на атаки з боку шкідливих вузлів.

Реалізація PBFT в Solana забезпечує глобальне джерело часу блокчейне за допомогою другого нового протоколу, відомого як Proof of History (PoH). По суті це забезпечує хроніку попередніх подій у блокчейні, гарантуючи, що існує загальний запис того, що і коли сталося.

Tower Consensus використовує цей синхронізований годинник для зменшення обчислювальної потужності, необхідної для перевірки транзакцій, оскільки більше не потрібно обчислювати тимчасові мітки попередніх транзакцій. Це допомагає Solana досягти пропускну здатності, яка перевершує більшість конкурентів (докладніше про це пізніше).

Крім цього, Solana включає ряд інших інновацій, які допомагають їй виділитися на тлі конкурентів. Серед них – технологія розпаралелювання транзакцій, відома як Sealevel. Вона дозволяє використовувати паралельне середовище виконання смарт-контрактів, яке оптимізує ресурси та гарантує, що Solana зможе масштабуватися горизонтально між процесорами та SSD-накопичувачами.

Коли справа доходить до децентралізованих програм, швидкість має значення. Про це свідчать проблеми, з якими стикається мережа Ethereum. Однак, в даний час Solana не страждає від цих проблем через свою архітектуру з високою пропускнуою здатністю.

Solana стверджує, що його блокчейн здатний підтримувати більше 50000 транзакцій за секунду (TPS) при піковому навантаженні, що робить його, мабуть, найшвидшим блокчейном, що працює в даний час. Для порівняння: це майже в 1 000 разів швидше за Bitcoin (~ 5-7 TPS) і більш ніж у 3 000 разів швидше за Ethereum (~ 15 TPS).

Більше того, Solana стверджує, що середній час блоку становить від 400 до 800 мілісекунд, а середня комісія за транзакцію становить 0,000005 SOL (або крихітну частку одного цента). Це, у поєднанні з величезною масштабованістю, робить його придатним для обслуговування децентралізованих програм, які можуть потенційно підтримувати десятки тисяч одночасних користувачів, не прогинаючи під навантаженням.

Solana досягає такої масштабованості, не вдаючись до технологій другого рівня або рішень поза ланцюжком, і не використовує жодних форм сегментування. Це робить його одним з небагатьох блокчейнів першого рівня, здатних досягати понад 1000 TPS.

На відміну від деяких платформ практично будь-який може запустити валідатор Solana і допомогти захистити мережу. Цей процес повністю позбавлений дозволів, хоча користувачам потрібно підтримувати деяке базове обладнання для участі, а саме сервер, який відповідає мінімальним специфікаціям, викладеним тут . В даний час, мережа може похвалитися

майже 1000 валідаторами, що робить її однією з найпоширеніших блокчейнів.

Solana, як і переважна більшість платформ смарт-контрактів, має власний токен газу, названий SOL. Усі транзакції та операції зі смарт-контрактами на Solana будуть споживати SOL.

Токен SOL також можна стейкати, щоб допомогти підтримати безпеку мережі та отримувати частину інфляції як винагороду. Хоча, в даний час ця функція недоступна, токени SOL в кінцевому підсумку також будуть використовуватися для управління всередині ланцюга.

Токен був вперше запущений в мережі бета-тестування Solana у березні 2020 року і в даний час входить до топ-20 найбільших криптовалют за ринковою капіталізацією[43].

1.6 Економетричні моделі і методи прогнозування часових рядів

1.6.1 Авторегресійні моделі

Авторегресійна (AR-) модель — модель часових рядів, у якій значення часового ряду лінійно залежить від попередніх значень цього ряду. Авторегресійна залежність порядку p визначається рівнянням [22]:

$$X_t = c + \sum_{i=1}^p a_i X_{t-i} + \varepsilon_t ,$$

де $a_1, \dots, a_p$ — параметри моделі (коефіцієнти авторегресії), c — константа(часто для спрощення передбачається рівною нулю), а ε_t — білий шум. Авторегресивна модель визначає, що вихідна змінна лінійно залежить від її власних попередніх значень і від стохастичного терміну (недосконало передбачуваний термін); таким чином, модель у формі стохастичного різницевого рівняння (або рекурентного співвідношення, яке не слід плутати

з диференціальним рівнянням). Разом з моделлю ковзного середнього (МА) – це окремий випадок і ключовий компонент більш загальних моделей ковзного середнього (ARMA) та авторегресійної інтегрованої ковзної середньої (ARIMA), які є стохастичними і більш складними. Така модель – частковий випадок векторної авторегресійної моделі (VAR), яка складається з системи більш ніж одного взаємозв’язаного стохастичного різницевого рівняння.

Якщо ввести лаговий оператор $L : Lx_t = x_{t-1}$, то авторегресійну модель можна представити так:

$$X_t = c + \sum_{i=1}^p a_i L^i X_t + \varepsilon_t ,$$

$$a(L)X_t = (1 - \sum_{i=1}^p a_i L^i)X_t = c + \varepsilon_t .$$

Стаціонарність авторегресійного процесу можна оцінити, знайшовши корені характеристичного полінома

$$a(z) = 1 - \sum_{i=1}^n a_i z^i$$

Щоб процес був стаціонарним, достатньо, щоб усі корені характеристичного полінома лежали поза межою одиничного кола в комплексній площині $|z| > 1$.

Зокрема, для AR(1)-процесу $a(z) = 1 - rz$, отже корінь цього полінома $z = 1/r$, тому умову стаціонарності можна записати як $|r| < 1$. Тобто коефіцієнт авторегресії (він же в даному випадку коефіцієнт автокореляції) повинен бути строго менше 1 за модулем.

Для AR(2)-процесу можна показати, що умови стаціонарності мають вигляд: $|a_2| < 1$, $a_2 \pm a_1 < 1$.

Стаціонарні AR-процеси допускають розкладання Вольда - подання у вигляді нескінченного МА-процесу[23]:

$$X_t = a^{-1}(L) + a^{-1}(L)\varepsilon_t = \frac{c}{1 - \sum_{i=1}^p a_i} + \sum_{j=0}^{\infty} b_j \varepsilon_{t-j}$$

Перший доданок є математичним очікуванням AR-процесу. Якщо $c=0$, то математичне очікування процесу також дорівнює нулю.

За допомогою AR-моделей можна моделювати сезонність. Такі моделі позначають SAR (Seasonal AR). Наприклад, за наявності квартальних даних та припущення про квартальну сезонність можна побудувати таку модель SAR(4):

$$y_t = a_4 y_{t-4} + \varepsilon_t$$

Фактично це звичайна AR-модель з обмеженням параметрів моделі (рівність нулю параметрів при лагах менше 4). На практиці сезонність може поєднуватися зі звичайною авторегресією, наприклад:

$$y_t = a_1 y_{t-1} + a_4 y_{t-4} + \varepsilon_t$$

У деяких випадках виявляються корисними сезонні моделі, у яких випадкова помилка підпорядковується деякому AR-процесу:

$$y_t = a_4 y_{t-4} + \varepsilon_t, \varepsilon_t = a_1 \varepsilon_{t-1} + u_t$$

Неважко побачити, що таку модель в операторній формі можна записати як:

$$(1 - a_1 L)(1 - a_4 L^4)y_t = u_t$$

Таку модель позначають $AR(1) \times SAR(4)$.

Моделі авторегресії (AR) і ковзного середнього (MA) використовуються для дослідження та прогнозування стаціонарних часових рядів. Модель ARMA узагальнює ці дві простіші моделі часових рядів.

Розглянемо модель $ARMA(p, q)$, де p і q - цілі числа, які визначають порядок моделі [24]. Моделлю $ARMA(p, q)$ називають наступний процес побудови часового ряду $\{X_t\}$:

$$X_t = c + \varepsilon_t + \sum_{i=1}^p a_i X_{t-i} + \sum_{i=1}^q b_i \varepsilon_{t-i},$$

де c - константа, $\{\varepsilon_t\}$ - білий шум, під яким розуміють послідовність однаково розподілених, зазвичай за нормальним законом з нульовим середнім, і незалежних випадкових величин, а $a_1, \dots, a_p$ та $b_1, \dots, b_q$ - дійсні числа, авторегресійні коефіцієнти та коефіцієнти ковзного середнього, відповідно.

Таку модель можна інтерпретувати як лінійну модель множинної регресії, в якій як пояснюючі змінні виступають минулі значення залежної змінної, а як регресійний залишок — ковзне середнє з елементів $\{\varepsilon_t\}$. Слід зазначити, що ARMA-процеси мають більш складну структуру порівняно з AR- або MA-процесами. Проте, ARMA-процеси мають меншу кількість параметрів, що є однією з їх переваг [22].

Побудова моделі ARMA на основі числового ряду потребує знання значень p і q , які задають порядок моделі. Для знаходження значень p і q застосовується автокореляційна функція і часткова автокореляційна

функція. Для визначення коефіцієнтів в моделі ARMA застосовуються методи найменших квадратів та максимальної правдоподібності.

До класичних ARMA-моделей можна додати деякі екзогенні фактори x . Розглянемо моделі ARMAX(p, q, k), у яких беруть участь як поточні значення цих чинників, так і значення лагів, де k – це кількість лагів для екзогенних факторів [23]. За умови присутності одного екзогенного фактору, модель ARMAX(p, q, k) записується так:

$$a(L)y_t = c + b(L)\varepsilon_t + d(L)x_t$$

де $a(L), b(L), d(L)$ – поліноми порядку p, q, k , відповідно. Такі моделі можна інтерпретувати інакше як моделі ADL(p, k) з випадковими помилками $MA(q)$.

Розглянемо ще одну модель аналізу часових рядів ARIMA, яка отримала назву інтегрованої моделі авторегресії з ковзним середнім. Ця модель є розширенням моделей ARMA і застосовується для нестационарних часових рядів. Відомо, що нестационарні часові ряди можна зробити стаціонарними шляхом взяття різниць деякого порядку від початкового часового ряду. В цьому випадку часові ряди називають інтегрованими або різницево-стаціонарними.

У моделі ARIMA(p, d, q) різниці порядку d у часовому ряду відповідають моделі ARMA(p, q). Модель ARIMA(p, d, q) для нестационарного часового ряду X_t має вигляд[25]:

$$\Delta^d X_t = c + \sum_{i=1}^p a_i \Delta^d X_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t$$

де ε_t – стаціонарний часовий ряд; c, a_i, b_j – параметри моделі; Δ^d – оператор різниці порядку d для часового ряду.

Також дана модель інтерпретується як $ARMA(p+d, q)$ – модель з d одиничними коренями. При $d = 0$ маємо звичайні $ARMA$ -моделі.

Оператор різниці порядку d полягає у послідовному розрахунку (d разів) різниці першого порядку. Спочатку різниця береться від початкового часового ряду, потім від різниць першого порядку, потім від різниць другого порядку тощо.

Модель $ARIMA(p, d, q)$ також може розглядатися як модель $ARMA(p+d, q)$ з d одиничними коренями. При $d = 0$ маємо звичайні $ARMA$ -моделі.

Розглянемо етапи аналізу часового ряду на основі моделі $ARIMA$ [23, 24, 25]. Спочатку оцінюється чи є ряд стаціонарним, чи ні. Використовуючи різні тести, оцінюють кількість одиничних коренів та визначають порядок інтегрованості часового ряду. При цьому на практиці часто використовують моделі з першим чи другим порядком інтегрованості. На другому етапі, якщо порядок інтегрованості більше нуля, ряд перетворюється на стаціонарний шляхом взяття різниць відповідного порядку. На третьому етапі будується $ARMA$ -модель для перетвореного стаціонарного ряду.

1.6.2 Гетероскедастичні моделі

Авторегресійна умовна гетероскедастичність — модель, що використовується в економетриці для аналізу часових рядів (насамперед фінансових), у яких умовна дисперсія ряду є залежною від минулих значень цього ряду, а також від інших чинників [22, 23]. Гетероскедастичні моделі використовуються для опису волатильності на фінансових ринках. Це задачі, у яких ряди мають періоди високої волатильності, що тривають деякий час, та періоди низької волатильності. При цьому довгострокову, безумовну, так звану середню волатильність можна вважати стабільною.

Моделі ARCH вперше були запропоновані Робертом Енглем у 1982 році. Вже 1986 року Боллерслев запропонував узагальнення цих моделей (GARCH). Надалі різні автори запропонували інші варіанти моделей даного типу, враховують ті чи інші особливості.

Нехай часовий ряд u_t являє собою наступний процес:

$$u_t = \varepsilon_t \sqrt{a_0 + \sum_{i=1}^q a_i u_{t-i}^2}$$

де ε_t — білий шум. Тоді як умовне, і безумовне математичне очікування цього процесу дорівнюватиме нулю. Умовна дисперсія цього процесу дорівнюватиме:

$$\sigma^2 = V(u_t | u_{t-1}, \dots, u_{t-q}) = a_0 + \sum_{i=1}^q a_i u_{t-i}^2$$

Така модель умовної дисперсії називається ARCH(q)-моделлю. Для недопущення від'ємних значень дисперсії передбачається, що коефіцієнти моделі невід'ємні, причому константа строго більша нуля. Якщо цей процес стаціонарний, то безумовна дисперсія постійна і рівна, очевидно:

$$\sigma^2 = \frac{\sigma_\varepsilon^2}{1 - \sum_{i=1}^q a_i}$$

Необхідна умова стаціонарності — сума коефіцієнтів моделі строго менша за одиницю. Інтегрований ARCH або нестаціонарний процес має місце, якщо сума коефіцієнтів дорівнює одиниці. Для ARCH-процесів характерне додатне значення коефіцієнту ексцесу, яке свідчить про так звані «товсті хвости». Так, наприклад, для ARCH(1)-процесу зсув від ексцесу

нормального розподілу дорівнює $6a_1^2/(1 - 3a_1^2)$, якщо $a_1 < 1/\sqrt{3}$. Оцінка параметрів ARCH(q)-моделі може бути здійснена за допомогою звичайного МНК.

В моделі ARCH умовна дисперсія залежить лише від квадратів попередніх значень часового ряду. Узагальнити цю модель можна зробивши припущення, що умовна дисперсія залежить також від попередніх значень умовної дисперсії. Це так званий узагальнений ARCH (Generalized ARCH - GARCH) [25, 26]. І тут GARCH(p, q) модель (де p — порядок GARCH-членів σ^2 та q – порядок ARCH-членів u^2) описується наступним чином:

$$\sigma_t^2 = a_0 + \sum_{i=1}^q a_i u_{t-i}^2 + \sum_{j=1}^p b_j \sigma_{t-j}^2$$

Необхідна умова стаціонарності $\sum_{i=1}^q a_i + \sum_{i=1}^p b_i < 1$. Безумовна дисперсія стаціонарного GARCH(p, q)-процесу буде постійна та дорівнює:

$$\sigma^2 = \frac{\sigma_\varepsilon^2}{1 - \sum_{i=1}^q a_i - \sum_{i=1}^p b_i}$$

Якщо сума коефіцієнтів дорівнює одиниці, маємо інтегрований GARCH — IGARCH, безумовна дисперсія якого нескінченна.

Асиметричні моделі GARCH [26] мають на меті врахувати асиметрію, що спостерігається іноді на фінансових ринках. Вважається, що «негативні шоки», до яких можна віднести погані новини тощо, зазвичай у більшій мірі впливають на волатильність, ніж «позитивні шоки» (наприклад, хороші новини). Таким чином, волатильність буде вищою на ринку, що падає, аніж на ринку, що зростає. Подібний ефект називається ефектом левериджу або важеля. При зниженні ціни акцій відбувається збільшення фінансового левериджу компаній, зростає волатильність, що в свою чергу відповідає

збільшенню рівня ризиків. У рамках класичних GARCH-моделей цей ефект пояснити неможливо, оскільки умовна дисперсія залежить від квадратів попередніх значень ряду.

Модель EGARCH запропонована Нельсоном у 1991 році. У цій моделі крім врахування асиметрії також вирішується проблема додатної визначеності моделі, оскільки замість умовних дисперсій моделі беруть участь їх логарифми[23]:

$$\ln \sigma_t^2 = a_0 + \sum_{i=1}^q a_i g(z_{t-i}) + \sum_{j=1}^p b_j \ln \sigma_{t-j}^2,$$

$$g(z_t) = \delta_1 z_t + \delta_2 (|z_t| - \sqrt{\frac{2}{\pi}}), \quad z_t \sim iid(0,1)$$

Асиметрична GARCH (AGARCH) модель запропонована Енглom у 1990 році.

$$\sigma_t^2 = a_0 + \sum_{i=1}^q a_i (u_{t-i} - \gamma)^2 + \sum_{j=1}^p b_j \sigma_{t-j}^2$$

У GARCH-моделях використовуються різні розподіли для кращої відповідності емпіричних особливостей фінансових рядів. Зокрема, часто застосовується розподіл Стюдента з малим числом ступенів свободи. Він характеризується більш «товстими хвостами» в порівнянні з нормальним розподілом. Моделі з розподілом Стюдента позначають GARCH-t [26]. Використовують також спеціальний скошений розподіл Стюдента з метою врахування асиметрії, який називається також t-розподілом Хансена. Такі моделі позначають GARCH-HT [26].

1.7 Постановка задачі дослідження

1. Виконати дослідження криптовалютного ринку, ознайомитися з історичними та теоритичними відомостями, що впливають на формування криптовалютного ринку.

2. Провести порівняльний аналіз сучасних модифікацій рекурентних нейронних мереж LSTM та GRU, таких як двонаправлені (bidirectional) LSTM та GRU, розширені (extended) LSTM, залежні (dependent) Bi-RNN, вказаних моделей з додаванням механізму уваги (attention-based mechanism), методів машинного навчання, а також методу групового врахування аргументів (МГУА).

3. Розробити програмний продукт для прогнозування вартості криптовалют. Виконати побудову, навчання моделей рекурентних нейронних мереж. Провести експерименти щодо підбору параметрів моделей. Обрати найкращу модель серед досліджених, оцінити якість отриманих прогнозів. Порівняти результати, отримані МГУА та моделями рекурентних нейронних мереж.

4. Розробити стартап-проект.

1.8 Висновки до першого розділу

Популярність електронних платежів в світі зростає швидкими темпами, щомісяця з'являються нові проекти і платіжні системи. Валютний ринок істотно змінився, коли торгувати стали криптовалютами. Варто відзначити, що технології в сфері криптовалют, які активно розвиваються, мають величезні переваги перед використанням в банківських установах і платіжних системах. Але в той же час ніхто не дасть гарантованого прогнозу щодо стабільності криптовалютного ринку і курсу цифрових валют. Йому

притаманні стихійність і відсутність будь-яких регуляторів, тому при вкладеннях потрібно без азарту і емоцій оцінювати перспективи обраної монети і користуватися інформацією тільки з перевірених ресурсів. Тому було прийнято рішення створити систему прогнозування курсу криптовалют.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

2.1 Історія нейронних мереж

Основні етапи в історії дослідження та застосування штучних нейронних мереж:

– 1943 - У. Маккалок і У. Піттс формалізують поняття нейронної мережі в фундаментальній статті про логічне обчислення ідей та нервової активності.

– 1948 - Н. Вінер разом із соратниками публікує роботу про кібернетику. Основною ідеєю є уявлення складних біологічних процесів математичними моделями.

– 1949 - Д. Хебб пропонує перший алгоритм навчання.

– У 1958 Ф. Розенблатт винаходить одношаровий перцептрон і демонструє його здатність розв'язувати задачі класифікації. Перцептрон набув популярності - його використовують для розпізнавання образів, прогнозування погоди тощо.

– У 1960 році Уїдроу разом зі своїм студентом Хоффом на основі дельта-правила (формули Уїдроу) розробили Адалін, який одразу почав використовуватися для завдань передбачення та адаптивного управління. Наразі Адалін (адаптивний суматор) грає роль стандартного елемента багатьох систем обробки сигналів.

– 1963 року в Інституті проблем передачі інформації АН СРСР. А. П. Петровим проводиться докладне дослідження завдань «важких» перцептрон.

– 1969 року М. Мінський публікує формальний доказ обмеженості перцептрону і показує, що він нездатний вирішувати деякі завдання (проблема «парності» та «один у блоці»), пов'язані з інваріантністю уявлень. Інтерес до нейронних мереж різко спадає[8].

– 1972 року Т. Кохонен та Дж. Андерсон незалежно пропонують новий тип нейронних мереж, здатних функціонувати як пам'ять.

– У 1973 році Б. В. Хакімов пропонує нелінійну модель із синапсами на основі сплайнів і впроваджує її для вирішення завдань у медицині, геології, екології.

– 1974 - Пол Дж. Вербос і А. І. Галушкін одночасно винаходять алгоритм зворотного розповсюдження помилки для навчання багат шарових перцептронів.

– 1975 - Фукусіма представляє когнітрон - мережу, що самоорганізується, призначену для інваріантного розпізнавання образів, але це досягається лише за допомогою запам'ятовування практично всіх станів образу.

– 1982 - після періоду забуття, інтерес до нейромереж знову зростає. Дж. Хопфілд показав, що нейронна мережа із зворотними зв'язками може являти собою систему, що мінімізує енергію (так звана мережу Хопфілда). Кохоненом представлені моделі мережі, що навчається без вчителя (нейронна мережа Кохонена), що вирішує завдання кластеризації, візуалізації даних (карта Кохонена, що самоорганізується) та інші завдання попереднього аналізу даних.

– 1986 - Девідом І. Румельхартом, Дж. Є. Хінтоном і Рональдом Дж. Вільямсом і одночасно з С. І. Барцевим та В. А. Охоніним (Красноярська група) перевідкритий та суттєво розвинений метод зворотного розповсюдження помилки. Почався вибух інтересу до нейронним мережам, що навчаються.

– 2007 Джеффри Хінтоном в університеті Торонто створено алгоритми глибокого навчання багат шарових нейронних мереж. Успіх обумовлений тим, що Хінтон під час навчання нижніх шарів мережі використовував обмежену машину Больцмана (RBM - Restricted Boltzmann Machine).

2.2 Класифікація нейронних мереж

Класифікація нейронних мереж за характером навчання поділяє їх на:

- нейронні мережі, які використовують навчання з учителем;
- нейронні мережі, які використовують навчання без учителя.

Нейронні мережі, які використовують навчання з учителем.

Навчання з вчителем припускає, що для кожного вхідного вектора існує цільовий вектор, що є необхідним виходом. Разом вони називаються навчальною парою. Зазвичай мережа навчається на таких навчальних парах.

Пред'являється вхідний вектор, обчислюється вихід мережі та порівнюється з відповідним цільовим вектором. Далі ваги змінюються відповідно до алгоритму, що прагне мінімізувати помилку. Вектори навчальної множини пред'являються послідовно, обчислюються помилки і ваги підлаштовуються для кожного вектора, поки помилка по всьому навчальному масиву не досягне прийнятного рівня.

Нейронні мережі, які використовують навчання без учителя.

Навчання без вчителя є набагато правдоподібнішою моделлю навчання з точки зору біологічних основ штучних нейронних мереж. Розвинена Кохоненом і багатьма іншими, вона не потребує цільового вектора для виходів і, отже, не вимагає порівняння з наперед ідеальними відповідями.

Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення достатньо близьких вхідних векторів давало однакові виходи. Процес навчання, отже, виділяє статистичні властивості навчальної множини і групує подібні вектори до класів.

Класифікація нейронних мереж за типом налаштування ваг поділяє їх на:

– мережі з фіксованими зв'язками – вагові коефіцієнти нейронної мережі вибираються відразу, з умов завдання;

– мережі з динамічними зв'язками – для них у процесі навчання відбувається налаштування синаптичних ваг.

Класифікація нейронних мереж за типом вхідної інформації поділяє їх на:

– аналогові – вхідна інформація представлена у формі дійсних чисел;
 – двійкові – вся вхідна інформація в таких мережах подається у вигляді нулів та одиниць[8].

2.3 Архітектури нейронних мереж

У **повнозв'язних нейронних мережах** кожен нейрон передає свій вихідний сигнал іншим нейронам, зокрема й собі. Усі вхідні сигнали подаються всім нейронам. Вихідними сигналами мережі можуть бути всі або деякі вихідні сигнали нейронів після кількох етапів функціонування мережі.

У **багатошарових нейронних мережах** нейрони поєднуються в шари. Шар містить сукупність нейронів із єдиними вхідними сигналами. Число нейронів у шарі може бути практично будь-яким і не залежить від кількості нейронів в інших шарах. У прикладі на рисунку 2.1 мережа складається з шарів, пронумерованих зліва направо. Зовнішні вхідні сигнали подаються на входи нейронів вхідного шару (його часто нумерують як нульовий), а виходами мережі є вихідні сигнали останнього шару. Крім вхідного та вихідного шарів у багатошаровій нейронній мережі є один або кілька прихованих шарів. Зв'язки від виходів нейронів деякого шару q до входів нейронів наступного шару $(q+1)$ називаються послідовними.

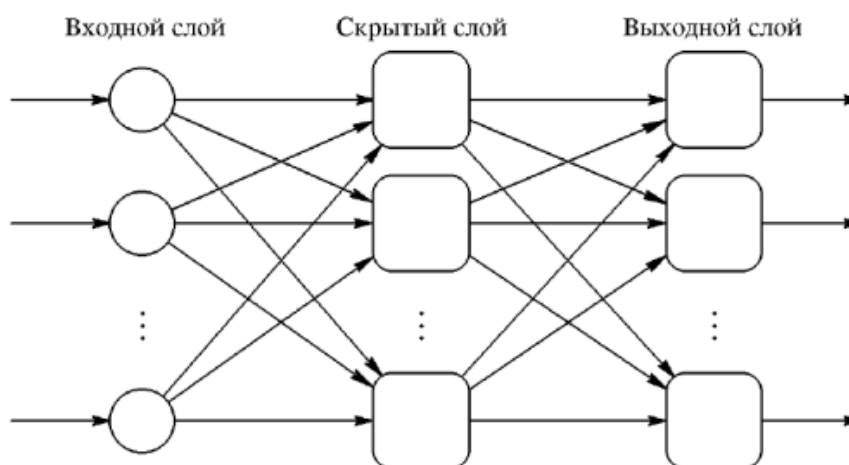


Рисунок 2.1 – Структура багатошарової нейронної мережі [8]

2.3.1 Типи багатошарових нейронних мереж

Монотонні. Це окремий випадок мереж з додатковими умовами на зв'язки та нейрони. Кожен шар, крім останнього (вихідного), розбитий на два блоки: збуджуючий і гальмуючий. Зв'язки між блоками теж поділяються на гальмуючі та збуджуючі. Якщо від нейронів блоку до нейронів блоку ведуть тільки збуджуючі зв'язки, то це означає, що будь-який вихідний сигнал блоку є монотонною неспадною функцією будь-якого вихідного сигналу блоку. Якщо ж ці зв'язки тільки гальмуючі, то будь-який вихідний сигнал блоку є незростаючою функцією-будь-якого вихідного сигналу блоку. Для нейронів монотонних мереж необхідна монотонна залежність вихідного сигналу нейрона від параметрів вхідних сигналів.

Мережі без зворотного зв'язку. У таких мережах нейрони вхідного шару отримують вхідні сигнали, перетворюють їх і передають нейронам першого прихованого шару, і так далі аж до вихідного, що видає сигнали для інтерпретатора та користувача. Зазвичай у таких мережах кожен вихідний сигнал q -го шару подається на вхід всіх нейронів $(q+1)$ -го шару; однак

можливий варіант з'єднання елементів q -го шару з елементами довільного шару.

Серед багатошарових мереж без зворотних зв'язків розрізняють **повнозв'язні** (вихід кожного нейрона q -го шару пов'язаний із входом кожного нейрона $(q+1)$ -го шару) та частково **повнозв'язні**.

2.3.2 Мережі із зворотними зв'язками

У **мережах із зворотними зв'язками** інформація з наступних шарів передається на попередні. Розрізняють такі типи нейронних мереж з зворотними зв'язками:

- **циклічні** відрізняються тим, що шари замкнуті в кільце: останній шар передає свої вихідні сигнали першому; всі шари рівноправні і можуть як отримувати вхідні сигнали, так і видавати вихідні;

- **повнозв'язні** складаються з шарів, кожен з яких є повнозв'язною мережею, а сигнали передаються як від шару до шару, так і всередині шару; у кожному шарі цикл роботи розпадається на три частини: прийом сигналів з попереднього шару, обмін сигналами всередині шару, вироблення вихідного сигналу та передача до наступного шару;

- **інші повнозв'язні**, що за своєю структурою аналогічні попереднім, але функціонують по-іншому: на кожному етапі нейрони всіх шарів приймають сигнали від нейронів як свого шару, і наступних.

На рисунку 2.2 нижче зображено мережу Елмана та мережу Жордана.

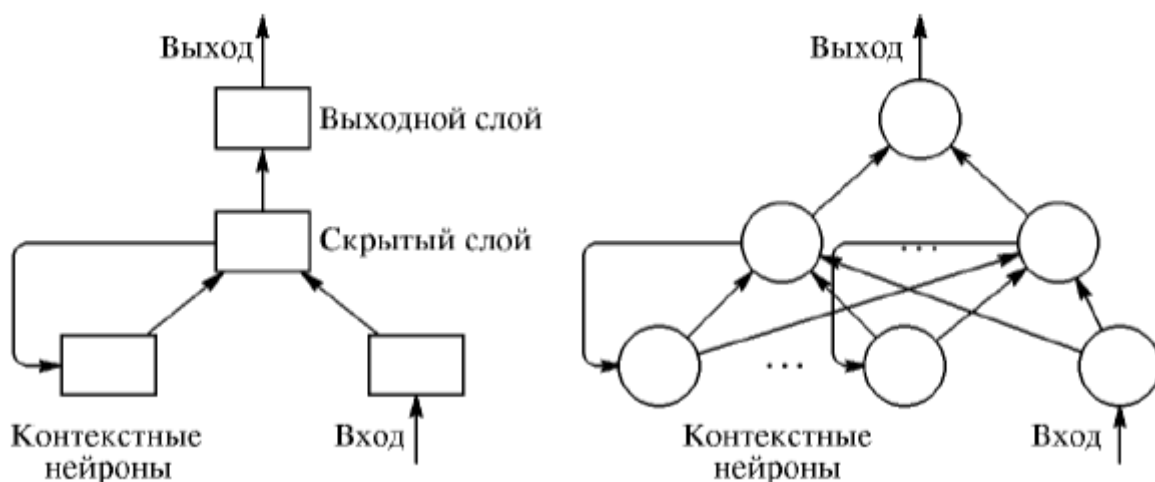


Рисунок 2.2 – Мережі Елмана та Жордана [8]

Нейронні мережі можна розділити за типами структур нейронів на однорідні та гетерогенні. **Однорідні** мережі складаються з нейронів одного типу з єдиною функцією активації, а **гетерогенну** мережу утворюють нейрони з різними функціями активації.

Ще одна класифікація ділить нейронні мережі на **синхронні** та **асинхронні**. У першому випадку в кожний момент часу лише один нейрон змінює свій стан, у другому – стан змінюється одразу у цілої групи нейронів, як правило, у всього шару. Алгоритмічно перебіг часу в нейронних мережах визначається ітераційним виконанням однотипних дій над нейронами.

2.3.3 Формальний нейрон

У нейрона є кілька вхідних каналів і лише один вихідний канал. По вхідних каналах на нейрон надходять дані завдання, але в виході формується результат роботи. Нейрон обчислює зважену суму вхідних сигналів, потім перетворює отриману суму з допомогою заданої нелінійної функції. Множину, що складається з порогового рівня та всіх ваг, називають множиною параметрів нейрона. На рисунку 2.3 введені такі позначення: X_1 ,

$X_2, \dots, X_n$ - вхідний сигнал (патерн), $w_1, w_2, \dots, w_n$ - вагові коефіцієнти, b - поріг нейрона.

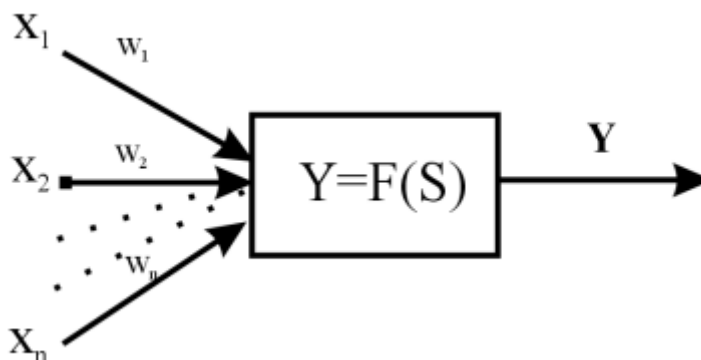


Рисунок 2.3 – Структура формального нейрону[8]

Спочатку нейрон обчислює зважену суму $S = \sum_i w_i X_i - b$, далі застосовуючи **функцію активації** $F(S)$ обчислює вихідний сигнал Y .

Функція активації нейрона – це функція, яка обчислює вихідний сигнал нейрона. На вхід цієї функції подається зважена сума сигналів та ваги цих сигналів.

Розглянемо найчастіше використовувані функції активації.

а) **Порогова функція**. Це проста кусково-лінійна функція. Якщо вхідне значення менше порогового, то значення функції активації дорівнює мінімальному допустимому, інакше максимально допустимому.

б) **Лінійний поріг**. Це нескладна кусково-лінійна функція. Має дві лінійні ділянки, де функція активації тотожно дорівнює мінімально допустимому і максимально допустимому значенню і є ділянка, на якій функція строго монотонно зростає.

в) **Сігмоїдальна функція або сигмоїда (sigmoid)**. Це монотонно зростаюча диференційована S-подібна нелінійна функція. Сигмоїда дозволяє посилювати слабкі сигнали та не насичуватися від сильних сигналів.

г) **Гіперболічний тангенс (hyperbolic tangent, tanh)**. Ця функція приймає на вході довільне дійсне число, а на виході дає дійсне число в інтервалі від -1 до 1. Подібно до сигмоїди, гіперболічний тангенс може

насичуватися. Проте, на відміну сигмоїди, вихід цієї функції центрований щодо нуля.

Недоліки формального нейрона:

- Передбачається, що нейрон миттєво обчислює свій вихід, тому за допомогою таких нейронів не можна моделювати безпосередньо системи із внутрішнім станом.
- Формальні нейрони на відміну від біологічних не можуть обробляти інформацію синхронно.
- Немає чітких алгоритмів вибору функції активації.
- Неможливо регулювати роботу мережі.
- Відсутня формалізація понять «поріг» та «вагові коефіцієнти». У реальних нейронів поріг змінюється динамічно, залежно від активності нейрона і загального стану мережі, а вагові коефіцієнти змінюються в залежності від сигналів, що проходять.

2.4 Нейронні мережі прогнозування часових рядів

2.4.1 Рекурентні нейронні мережі

Рекурентні нейронні мережі - вид нейронних мереж, де зв'язки між елементами утворюють спрямовану послідовність. Завдяки цьому з'являється можливість обробляти серії подій у часі або послідовні просторові ланцюжки. На відміну від багатошарових перцептронів, рекурентні мережі можуть використовувати свою внутрішню пам'ять для обробки послідовностей довільної довжини. Тому мережі RNN застосовуються у таких завданнях, де щось цілісне розбито на сегменти, наприклад, розпізнавання рукописного тексту чи розпізнавання мови. Було запропоновано багато різних архітектурних рішень для рекурентних мереж від найпростіших до складних. Останнім часом найбільшого поширення

набули мережа з довготривалою та короткочасною пам'яттю (LSTM) та керований рекурентний блок (GRU). На рисунку 2.4 частина нейронної мережі **A** отримує деякі дані **X** на вхід та подає на вихід деяке значення **H**. Циклічний зв'язок дозволяє передавати інформацію від поточного кроку мережі до наступного.

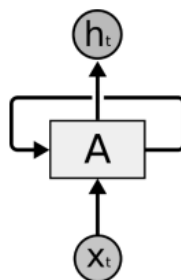


Рисунок 2.4 – Частина нейронної мережі [9]

Існує багато різновидів, рішень та конструктивних елементів рекурентних нейронних мереж. Проблема рекурентної мережі полягає в тому, що якщо враховувати кожен крок часу, то стає необхідним для кожного кроку часу створювати свій шар нейронів, що викликає серйозні обчислювальні складності. Крім того, багатошарові реалізації виявляються обчислювально нестійкими, тому що в них як правило зникають або стрімко зростають ваги.

Якщо обмежити розрахунок фіксованим часовим вікном, то отримані моделі не відображатимуть довгострокові тренди. Різні підходи намагаються вдосконалити модель історичної пам'яті та механізм запам'ятовування та забування.

Рекурентні нейронні мережі не так сильно відрізняються від звичайних нейронних мереж. Їх можна уявити собі, як безліч копій однієї та тієї ж мережі, причому кожна копія передає повідомлення наступній копії. На рисунку 2.5 показано розгорнутий цикл рекурентної мережі.

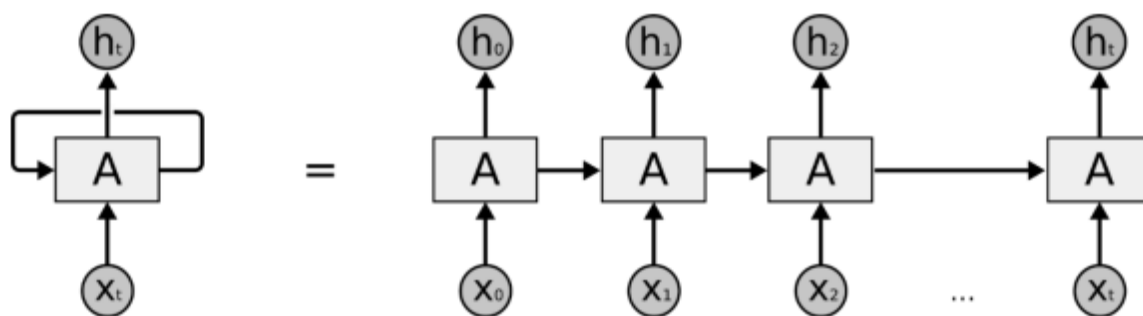


Рисунок 2.5 – Розгорнутий цикл РНМ [9]

Така “ланцюгова” сутність показує, що рекурентні нейронні мережі за своєю природою тісно пов'язані з послідовностями та списками.

Базова архітектура, що була розроблена у 1980-х роках, будується з вузлів, кожен із яких з'єднаний з усіма іншими вузлами. У кожного нейрона поріг активації змінюється згодом і є дійсним числом. Всі нейрони мають змінні дійснозначні ваги. Вузли поділяються на вхідні, вихідні та приховані.

В задачах навчання з учителем на кожному дискретному кроці часу на вхідні вузли подаються дані, а інші вузли завершують свою активацію, та вихідні сигнали готуються для передачі нейронам наступного рівня. Якщо, наприклад, мережа відповідає за розпізнавання мови, в результаті до вихідних вузлів надходять вже мітки (розпізнані слова).

У задачах навчання з підкріпленням (reinforcement learning), де немає вчителя, який забезпечує цільові сигнали для мережі, використовується функція придатності або функція оцінки (reward function). За цією функцією проводиться оцінка якості роботи мережі, при цьому значення на виході впливає на поведінку мережі на вході.

Зокрема, якщо мережа реалізує гру, на виході вимірюється кількість пунктів виграшу або оцінка позиції. Кожен ланцюжок обчислює помилку як сумарну девіацію за вихідними сигналами мережі. Якщо є набір навчальних прикладів, помилка обчислюється з урахуванням помилок кожного окремого прикладу.

Завдання прогнозування часових рядів – складний тип проблеми прогнозуючого моделювання. У часових рядах виникає потреба оцінювання

залежності послідовностей від вхідних змінних. Потужний тип нейронної мережі, призначений для обробки послідовностей, називається рекурентними нейронними мережами.

Мережа з довготривалою пам'яттю або мережа LSTM - це тип рекурентної нейронної мережі, що використовується в глибокому навчанні. На її основі можна успішно навчати дуже великі архітектури.

2.4.2 LSTM мережі

Мережі довготривалої пам'яті (Long Short Term Memory) – зазвичай просто називають "LSTM" - особливий вид PHM, здатних до навчання довгостроковим залежностям. Вони показують дуже добрі результати при розв'язанні великої різноманітності проблем і в даний момент широко застосовуються. LSTM спеціально спроектовані таким чином, щоб уникнути проблеми довгострокових залежностей. Запам'ятовувати інформацію на тривалий період часу - це практично їхня поведінка за замовчуванням, а не щось таке, що вони лише намагаються зробити.

У мережах LSTM вдалося обійти проблему зникнення або вибухового зростання градієнтів у процесі навчання шляхом зворотного поширення помилки. Мережа LSTM зазвичай управляється за допомогою рекурентних вентилів, які називаються вентилями (gates) «забування». Помилки поширюються назад по часу через потенційно необмежену кількість віртуальних шарів.

Таким чином відбувається навчання в LSTM, при цьому зберігаючи пам'ять тисяч і навіть мільйонів часових інтервалів у минулому. Топології мереж типу LSTM можуть розроблятися відповідно до специфіки завдання. У мережі LSTM можуть враховуватися навіть великі затримки між значними

подіями, і тим самим високочастотні та низькочастотні компоненти можуть змішуватись.

Усі рекурентні нейронні мережі мають форму ланцюга модулів, що повторюються (repeating module). У стандартній PHM (рисунок 2.6) ці модулі, що повторюються, матимуть дуже просту структуру, наприклад, всього один шар гіперболічного тангенсу (tanh).

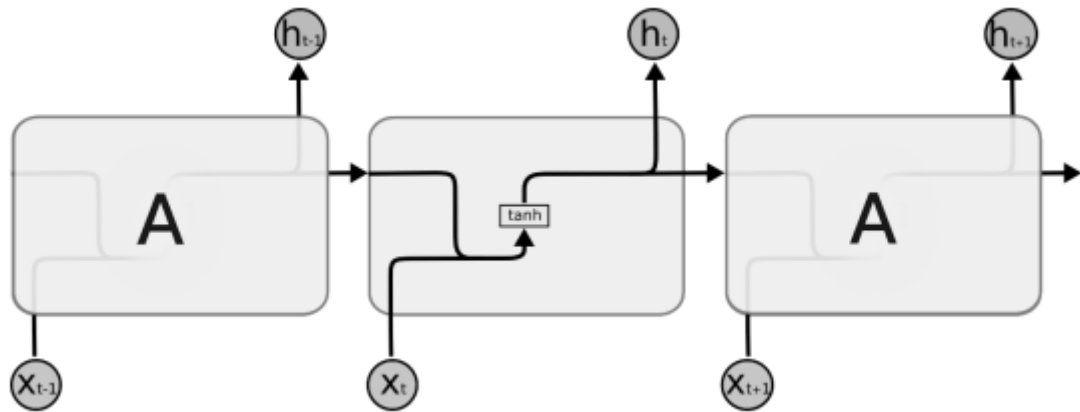


Рисунок 2.6 – Схема модуля PHM [9]

LSTM теж мають таку ланцюгову структуру, але модуль, що повторюється має іншу будову. Замість одного нейронного шару їх чотири, причому вони взаємодіють особливим чином (рисунок 2.7).

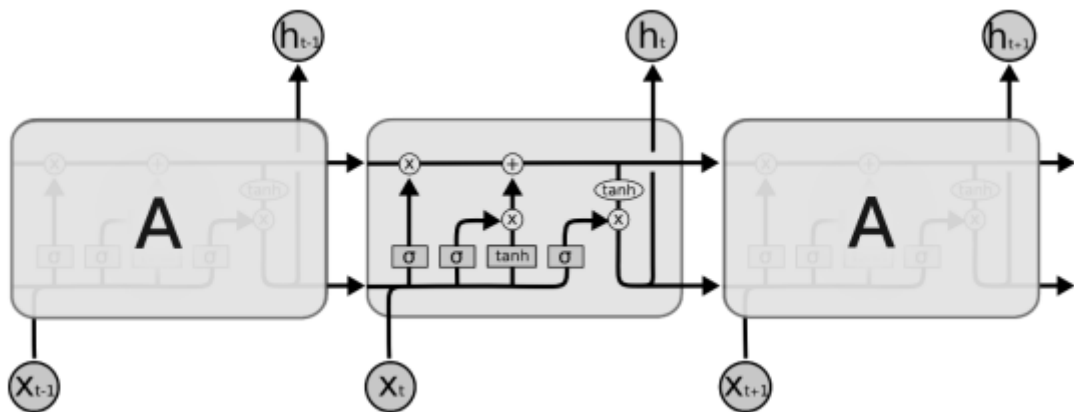


Рисунок 2.7 – Схема модуля LSTM [9]

На рисунку 2.7, кожна лінія передає цілий вектор від виходу одного вузла до входів інших. Кола представляють поточкові оператори, такі як додавання векторів, в той час, як прямокутники - це навчені шари нейронної мережі. На рисунок 2.7 окремі направлені лінії позначають конкатенацію

сигналу, у той час як лінії, що розгалужуються, позначають, що їх вміст копіюється, і копії вирушають у різні місця.

Ключ до LSTM – клітинний стан (cell state) – горизонтальна лінія, що проходить крізь верхню частину діаграми (рисунок 2.8). Клітинний стан – це щось на кшталт стрічки конвеєра. Вона рухається прямо вздовж усього ланцюга тільки з невеликими лінійними взаємодіями. Інформація може просто текти по ній без змін.

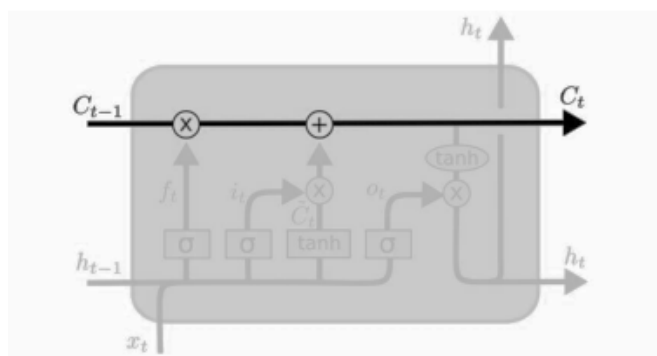


Рисунок 2.8 – Клітинний стан [9]

LSTM має здатність видаляти або додавати інформацію до клітинного стану, проте ця здатність ретельно регулюється структурами, які називаються вентилями (gates). Вентилі – це спосіб вибірково пропускати інформацію. Вони складені із сигмоїдного шару НМ та операції поточкового множення (pointwise multiplication) (рисунок 2.9).

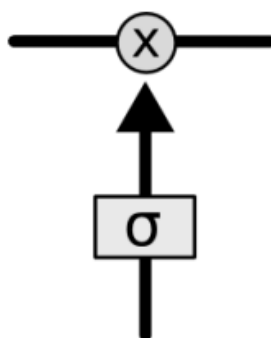


Рисунок 2.9 – Сигмоїдний шар [9]

Сигмоїдний шар подає на вихід числа між нулем та одиницею, описуючи таким чином, наскільки кожен компонент має бути пропущений крізь вентиль. Нуль - "нічого не пропускати", один - "пропускати все". LSTM має три таких вентиля, щоб захищати та контролювати клітинний стан.

2.4.3 Різновиди LSTM мереж

В одному з популярних варіантів LSTM додаються “очні з'єднання” (“peephole connections”). Це означає, що вентилям дозволяється “підглядати” за клітинним станом (рисунок 2.10).

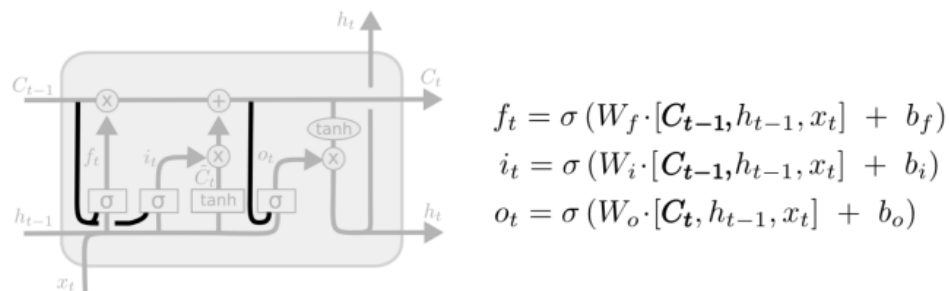


Рисунок 2.10 – «Peephole connections» в модулі LSTM [9]

Інша варіація - використання спарених забуваючих і входних вентилів (рисунок 2.11). Замість того, щоб незалежно вирішувати, що забути і куди ~~ми~~ потрібно додати нову інформацію, ці рішення приймаються одночасно. Забування відбувається тільки у тому випадку, коли отримуємо щось інше на це місце. На вхід отримуються нові значення тільки коли забувається щось старе.

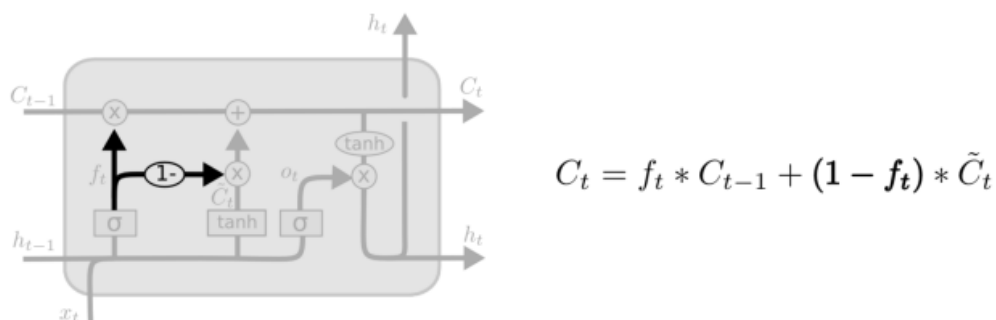


Рисунок 2.11 – Спарені забуваючі в модулі LSTM [9]

2.4.4 Gated Recurrent Unit

Дещо суттєвіше відрізняється від LSTM вентиляна рекурентна одиниця (Gated Recurrent Unit) чи GRU (рисунок 2.12). Вона поєднує забуваючі та вхідні вентиля в один “оновлюючий вентиль” (“update gate”). Вона також зливає клітинний стан із прихованим шаром та вносить деякі інші зміни. Модель, що виходить в результаті, простіша, ніж звичайна модель LSTM і вона набирає популярності [9].

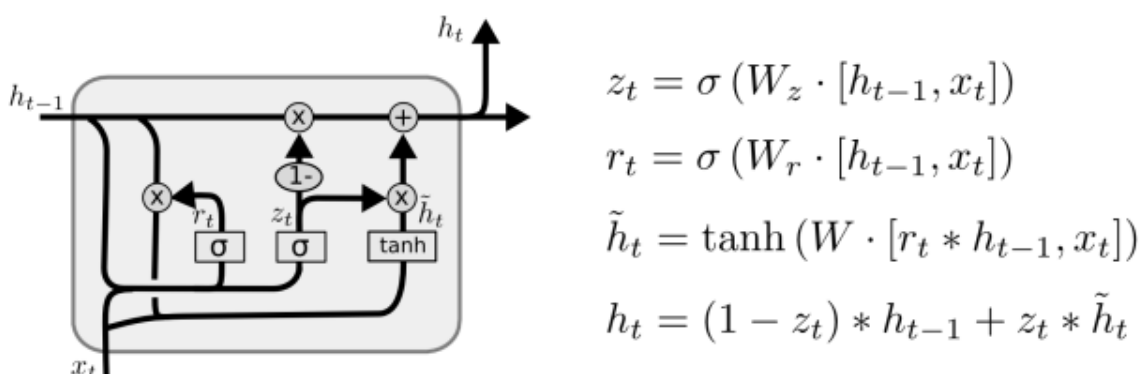


Рисунок 2.12 – GRU модуль [9]

Це лише деякі з найпомітніших варіантів LSTM. Є безліч інших, наприклад, глибинно-вентильні PHM (Depth Gated RNNs). Існує і зовсім інший підхід до вивчення довгострокових залежностей, наприклад, вартові PHM (Clockwork RNNs).

Мережі кодувальників-декодувальників [19, 20], засновані на модулях LSTM або GRU, стали популярними завдяки їхньому успіху в машинному перекладі. Ключова ідея – закодувати початкову послідовність як вектор фіксованої довжини та використати декодер для побудови перекладу.

Одна проблема з мережами кодер-декодер полягає в тому, що їхня продуктивність швидко погіршується в міру збільшення довжини вхідної послідовності. При аналізі часових рядів, однак, зазвичай очікується що прогнози будуть побудовані на основі відносно довгих послідовностей часових рядів.

Для вирішення цієї проблеми будується мережа кодувальника-декодувальника на основі уваги [21], використовуючи механізм уваги для вибору частин прихованих станів на всіх часових кроках.

Нова двоетапна рекурентна нейронна мережа на основі уваги (dual-stage attention-based RNN, DA-RNN) для прогнозування часових рядів запропонована в [17]. На першому етапі розроблено новий механізм уваги для адаптивного отримання відповідних рушійних рядів на кожному часовому кроці, посиляючись на попередній прихований стан кодувальника.

На другому етапі використовується механізм тимчасової уваги (temporal attention mechanism) для вибору відповідних прихованих станів кодувальника на всіх часових кроках. Ці дві моделі уваги добре інтегруються до нейронної рекурентної мережі RNN на основі LSTM і можуть бути спільно навчені з використанням стандартного алгоритму зворотного поширення.

Тому мережа DA-RNN може адаптивно вибирати найбільш релевантні вхідні ознаки, а також відповідним чином фіксувати довгострокові часові залежності часового ряду. Автори [17] доводять ефективність цього підходу, а також показують, що DA-RNN легко інтерпретувати та він є стійким до шумів у вхідних рядах.

2.4.5 Двонаправлені рекурентні нейронні мережі Bi-RNN та Bi-LSTM

Двонаправлені рекурентні нейронні мережі Bi-RNN насправді просто об'єднують дві незалежні РНМ. Ця структура дозволяє мережам мати як зворотну, так і пряму інформацію про послідовність на кожному кроці часу.

Використання двонаправленості буде виконувати введення двома способами: один з минулого в майбутнє, а другий з майбутнього в минуле (рисунок 2.13). Відміна цього підходу від традиційного, односпрямованого полягає в тому, що в LSTM, який працює назад, виконується збереження

інформації з майбутнього, а за допомогою поєднання двох прихованих станів з'являється можливість в будь-який момент часу зберегти інформацію як з минулого, так і з майбутнього [10].

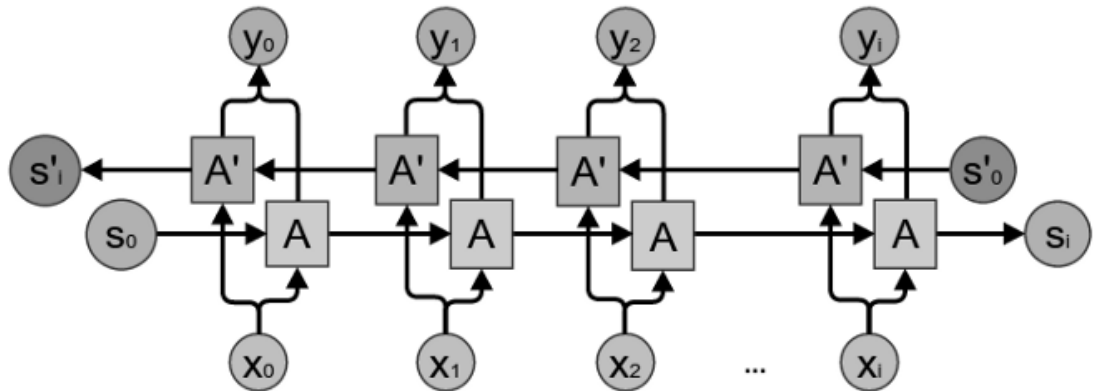


Рисунок 2.13 – Схема двонаправленої рекурентної нейронної мережі [11]

Мережу Bi-RNN можна навчати за допомогою алгоритмів, подібних до RNN, оскільки два спрямованих нейрони не мають жодної взаємодії. Однак, коли застосовується зворотне поширення в часі, необхідні додаткові процеси, оскільки оновлення входних і вихідних шарів неможливо виконати одночасно.

Загальні процедури навчання наступні [11]. Для прямого проходу спочатку передаються прямі стани та стани назад, потім вихідні нейрони. Для зворотного проходу спочатку передаються вихідні нейрони, потім передаються прямі стани та стани назад. Після виконання вперед і назад ваги оновлюються.

Мотивація для Bi-RNN полягає в тому, щоб повністю використовувати послідовність введення, якщо майбутня інформація буде доступною. Це особливо корисно, якщо поточний вихід також є функцією майбутніх входів. Умовна функція щільності ймовірності у Bi-RNN має вигляд:

$$\underline{p}_t = P(\underline{Y}_t | \{\underline{X}_t\}_{t=1}^T) = \underline{W}^f \underline{p}_t^f + \underline{W}^b \underline{p}_t^b, \quad (1)$$

$$\hat{\underline{Y}}_t = \operatorname{argmax}_{\underline{Y}_t} \underline{p}_t, \quad (2)$$

$$\underline{p}_t^f = P(\underline{Y}_t | \{\underline{X}_i\}_{i=1}^t), \quad (3)$$

$$\underline{p}_t^b = P(\underline{Y}_t | \{\underline{X}_i\}_{i=t}^T), \quad (4)$$

де $\underline{W}^f$ і $\underline{W}^b$ – ваги, які можна навчати, $\hat{\underline{Y}}_t$ – прогнозований вихідний елемент на етапі часу t .

Таким чином, вихід є комбінацією оцінки щільності прямого RNN і вихідного результату зворотного RNN. Завдяки двонаправленній конструкції, Bi-RNN може використовувати інформацію всієї вхідної послідовності для прогнозування кожного окремого вихідного елемента.

Кодер-декодер вперше був запропонований для машинного перекладу (МП) у поєднанні з GRU [13]. Він був мотивований впоратися з ситуацією, коли $T' \neq T$. Він має дві комірки RNN: кодер і декодер. Як видно на рисунку 2.14, кодер (позначений Enc) приймає вхідну послідовність довжини T і генерує її вихід $\underline{h}_i^{Enc}$ і прихований стан $\underline{c}_i^{Enc}$, де $i \in \{1, \dots, T\}$.

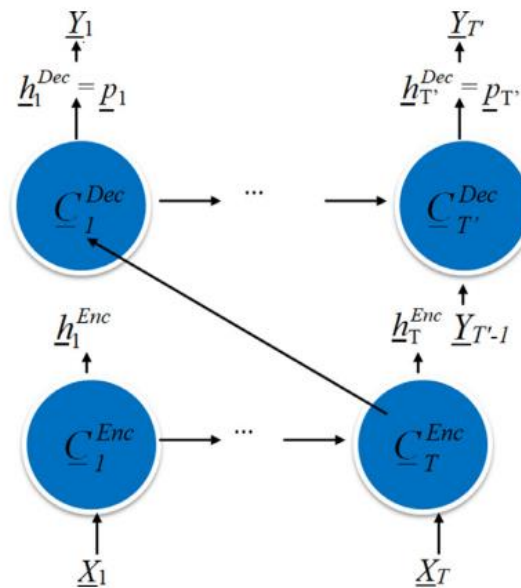


Рисунок 2.14 – Структура кодер-декодер мережі(seq2seq) [14]

У моделі seq2seq прихований стан кодера на часовому кроці T використовується як представлення вхідної послідовності. Потім декодер використовує інформацію про прихований стан для генерування вихідної послідовності довжини T' шляхом ініціалізації свого прихованого стану $\underline{c}_1^{Dec}$

як $\underline{C}_T^{Enc}$. Отже, процес декодування починається після того, як кодер обробив всю вхідну послідовність.

На практиці вхід до декодера на етапі часу 1 є попередньо визначеним символом початку декодування. На наступних часових кроках попередній вихід $\underline{Y}_{t-1}$ буде використовуватися як вхід. Декодер зупинить процес декодування, якщо буде згенерований спеціальний попередньо визначений символ зупинки. Порівняно з Bi-RNN, кодер-декодер не тільки має переваги у своїй здатності обробляти вхідно-вихідні послідовності різної довжини, але також здатний генерувати краще вирівняні вихідні послідовності шляхом явного подання попередніх передбачених вихідних даних назад до свого декодера. Таким чином, кодер-декодер оцінює наступну функцію щільності:

$$\underline{p}_t = P(\underline{Y}_t | \{\hat{\underline{Y}}_i\}_{i=1}^{t-1}, \{\underline{X}_i\}_{i=1}^T), \quad (5)$$

$$\hat{\underline{Y}}_t = \operatorname{argmax}_{\underline{Y}_t} \underline{p}_t \quad \forall t \in \{1, \dots, T\}. \quad (6)$$

Для кращого функціонування моделі кодер-декодер запропоновано різні механізми уваги для цієї моделі. Введено додаткові зважені зв'язки для підключення декодера до прихованого стану кодера [14,15]. З іншого боку, система кодер-декодер уразлива до попередніх помилкових прогнозів на прямому шляху. Нещодавно Bi-RNN було введено в кодер Багданау та ін. [14], але їхня конструкція не вирішує проблему помилкового передбачення.

2.4.6 Extended LSTM

Вводимо коефіцієнт масштабування для компенсації швидкого згасання вхідної реакції. Це призводить до нового рішення під назвою

розширений LSTM (ELSTM). Комірку ELSTM зображено на рисунку 2.15, де $\underline{s}_i \in R^N, i = 1, \dots, t - 1$ – навчальні входні вектори масштабування.

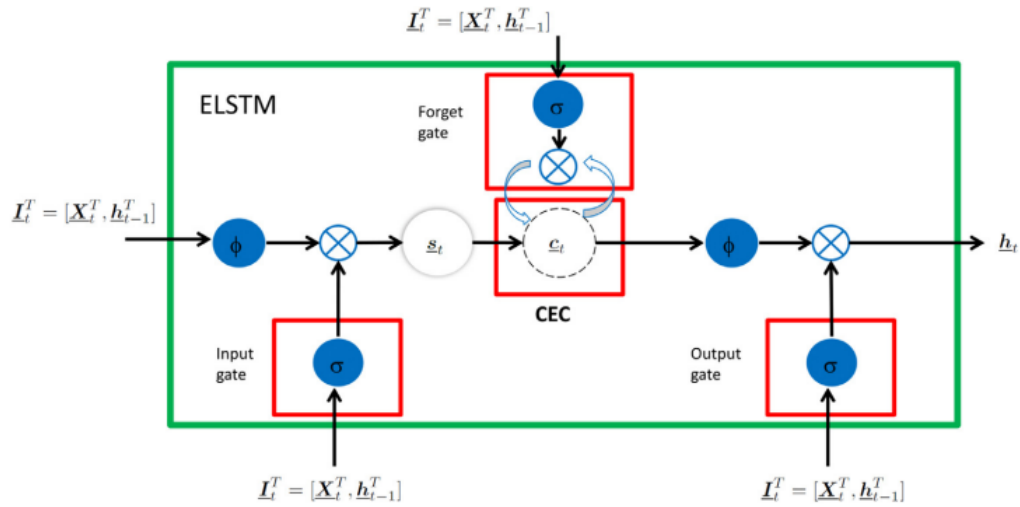


Рисунок 2.15 – Модуль ELSTM [12]

Модуль ELSTM можна описати за допомогою:

$$\underline{c}_t = \sigma(\underline{W}_f \underline{I}_t) \underline{c}_{t-1} + \underline{s}_i \sigma(\underline{W}_i \underline{I}_t) \phi(\underline{W}_{in} \underline{I}_t), \quad (7)$$

$$\underline{h}_t = \sigma(\underline{W}_o \underline{I}_t) \phi(\underline{c}_t) \quad (8)$$

член зміщення $\underline{b} \in R^N$ для $\underline{c}_t$ опущено в рівнянні (8). Як показано вище, до ELSTM вводиться коефіцієнт масштабування $\underline{s}_i$, $i = 1, \dots, t - 1$, щоб збільшити або зменшити вплив входу $\underline{I}_t$ у послідовності. Щоб показати, що ELSTM має більшу пам'ять, ніж LSTM, спочатку отримуємо вираз закритої форми $\underline{h}_t$:

$$\underline{h}_t = \sigma(\underline{W}_o \underline{I}_t) \phi \left(\sum_{k=1}^t \underline{s}_k [\prod_{j=k+1}^t \sigma(\underline{W}_f \underline{I}_j)] \sigma(\underline{W}_f \underline{I}_k) \phi(\underline{W}_{in} \underline{I}_k) \right) \quad (9)$$

Тоді можна знайти наступний окремий випадок:

$$\exists \underline{s}_k \text{ такий, що } |\underline{s}_k \prod_{j=k+1}^t \sigma(\underline{W}_f \underline{I}_j)| \geq |\prod_{j=k+1}^t \sigma(\underline{W}_f \underline{I}_j)| \quad \forall \underline{W}_f \quad (10)$$

Робимо висновок, що завжди існує ELSTM, чия пам'ять довша за всі варіанти LSTM. І навпаки, не можна знайти таку LSTM з аналогічною перевагою ELSTM. Це демонструє, що система ELSTM за побудовою має перевагу перед LSTM. Подібне скалярне рішення також існує для згорткових нейронних мереж (CNN) [12].

Кількість параметрів, які використовуються різними модулями RNN, порівняно в таблиці 2.1, де $\underline{X}_t \in \mathbb{R}^M$, $\underline{h}_t \in \mathbb{R}^N$ і $t = 1, \dots, T$.

Таблиця 2.1 – Кількість параметрів для кожного типу рекурентної мережі

Модуль	Кількість параметрів
LSTM	$4N(M + N + 1)$
GRU	$3N(M + N + 1)$
ELSTM	$4N(M + N + 1) + N(T + 1)$

Як показано в таблиці 2.1, кількість параметрів комірки ELSTM залежить від максимальної довжини T вхідних послідовностей, які робить розмір моделі неконтрольованим. Щоб вирішити цю проблему, вибирають фіксований T_s (з $T_s < T$) як верхню межу для кількості масштабних коефіцієнтів і встановлюють $\underline{s}_k = \underline{s}_{(k-1) \bmod T_s + 1}$, якщо $k > T_s$. Значення k починається з 1, де $\bmod$ позначає оператор за модулем.

Таким чином, послідовність масштабуючих коефіцієнтів є періодичною з періодом T_s , тому елементи в послідовності, що віддалені на довжину T_s , матимуть той самий коефіцієнт масштабування.

Можна описати комірку ELSTM з періодичними коефіцієнтами масштабування як:

$$\underline{c}_t = \sigma(\underline{W}_f \underline{I}_t) \underline{c}_{t-1} + \underline{s}_{t_s} \sigma(\underline{W}_i \underline{I}_t) \varphi(\underline{W}_{in} \underline{I}_t), \quad (11)$$

$$\underline{h}_t = \sigma(\underline{W}_o \underline{I}_t) \varphi(\underline{c}_t), \quad (12)$$

де $t_s = (t - 1) \bmod T_s + 1$. Вибір T_s впливає на продуктивність мережі. Так, невеликі значення T_s підходять для простих мовних завдань, які вимагають коротшої пам'яті, тоді як для складних завдань, які вимагають більш тривалої пам'яті, потрібні більші значення T_s . Для моделей РНМ типу «послідовність у послідовність», seq2seq, більша величина T_s вважається кращою [12].

2.4.7 Dependent Bi-RNN (DBRNN)

Одна РНМ рідко використовується на практиці через її обмежені можливості в моделюванні реальних проблем. Щоб побудувати більш потужну модель РНМ, часто потрібно інтегрувати кілька комірок з різними імовірнісними моделями. Наприклад, модель послідовності до послідовності (seq2seq) потребує передбачити вихідну послідовність, $\{\underline{Y}_t\}_{t=1}^{T'}$, де $\underline{Y}_i \in R^N$, на основі вхідної послідовності $\{\underline{X}_t\}_{t=1}^T$, де $\underline{X}_i \in R^M$, де T та T' довжини вхідної та вихідної послідовностей відповідно. Модель залежної двонаправленої рекурентної нейронної мережі (DBRNN) показана на рисунку 2.16.

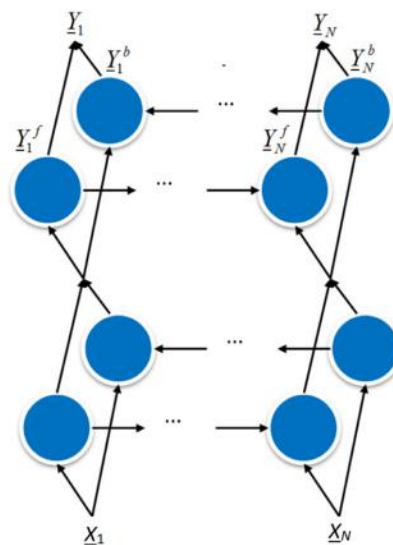


Рисунок 2.16 – Структура моделі DBRNN [14]

Вона складається з нижньої та верхньої гілок Bi-RNN. На кожному кроці часу вхід до прямої і зворотної частин верхнього Bi-RNN є об'єднаними прямими і зворотними виходами з нижньої гілки Bi-RNN. Остаточне двонаправлене передбачення — це об'єднання як прямих, так і зворотних прогнозів. Така конструкція робить DBRNN стійким до попередніх помилкових прогнозів.

Нехай F — функція комірки. Вхід подається в пряму і зворотну RNN нижньої гілки Bi-RNN.

$$\underline{h}_t^f = F_l^f(\underline{x}_t, \underline{c}_{l(t-1)}^f), \underline{h}_t^b = F_l^b(\underline{x}_t, \underline{c}_{l(t+1)}^b), \underline{h}_t = \begin{bmatrix} \underline{h}_t^f \\ \underline{h}_t^b \end{bmatrix}, \quad (13)$$

де $\underline{c}$ і l позначають прихований стан комірки і нижній Bi-RNN відповідно.

Кінцевий вихід, $\underline{h}_t$, нижнього Bi-RNN є конкатенацією виводу, $\underline{h}_t^f$, прямого RNN і виводу, $\underline{h}_t^b$, зворотної RNN. Аналогічно, верхній Bi-RNN генерує кінцевий вихідний $\underline{p}_t$

$$\underline{p}_t^f = F_u^f(\underline{h}_t, \underline{c}_{u(t-1)}^f), \underline{p}_t^b = F_u^b(\underline{h}_t, \underline{c}_{u(t+1)}^b), \underline{p}_t = W^f \underline{p}_t^f + W^b \underline{p}_t^b, \quad (14)$$

де u позначає верхній Bi-RNN. Для генерування прямого передбачення $\hat{\underline{Y}}_t^f$ і зворотного передбачення $\hat{\underline{Y}}_t^b$, прямий і зворотний шляхи верхньої гілки Bi-RNN окремо навчаються оригінальною та зворотною цільовими послідовностями відповідно. Результати прямих і зворотних прогнозів верхньої гілки RNN потім об'єднуються для отримання кінцевого результату.

2.4.8 Метод групового урахування аргументів

Метод групового врахування аргументів (МГУА) — сімейство індуктивних алгоритмів для математичного моделювання багатопараметричних даних. Метод заснований на рекурсивному селективному відборі моделей, на основі яких будуються складніші моделі. Точність моделювання на кожному наступному кроці рекурсії збільшується за рахунок ускладнення моделі[44].

Метод Групового Урахування Аргументів застосовується у самих різноманітних областях для аналізу даних та знаходження знань, прогнозування і моделювання систем, оптимізації і розпізнавання образів.

Індуктивні алгоритми МГУА дають унікальну можливість автоматично знаходити взаємозалежності у даних, вибрати оптимальну структуру моделі чи мережі, підвищувати точність існуючих алгоритмів.

Цей підхід самоорганізації моделей принципово відрізняється від дедуктивних методів, що звичайно використовуються. Він заснований на індуктивних принципах - вирішення задач засновано на переборі по зовнішньому критерію.

За допомогою перебору різних розв'язань методи індуктивного моделювання намагаються мінімізувати роль припущень автора у результатах моделювання. Комп'ютер сам знаходить структуру моделі і закони, що діють в об'єкті. Він може бути використаний при створенні штучного інтелекту як радник для вирішення суперечок та прийнятті рішень.

Метод Групового Урахування Аргументів складається з декількох алгоритмів для вирішення різних задач. В нього входять як параметричні, так і алгоритми кластеризації, комплексування аналогів, ребінаризації та ймовірностні алгоритми. Цей підхід самоорганізації заснований на переборі моделей, що поступово ускладнюються та на виборі найкращого розв'язку згідно з мінімумом зовнішнього критерія. В якості базисних моделей

використовуються не тільки поліноми, але й також нелінійні, ймовірнісні функції чи кластеризації[44].

Метод групового врахування аргументів (МГВА) базується на заданні правил ускладнення моделі, системі опорних функцій, критерія селекції та методу регуляризації згідно зовнішнім критеріям. ЕОМ проводить генерацію моделей-претендентів, селекцію згідно зовнішнім критеріям та відсів моделей, що не пройшли селекцію. В зв'язку з чим основну структуру алгоритму самоорганізації можна навести у такому вигляді:

- 1) попередня обробка спостережень з урахуванням системи обраних опорних функцій (скорочується кількість претендентів);
- 2) генерація множини моделей-претендентів;
- 3) обчислення критеріїв селекції, що є зовнішніми доповненнями, та пошук моделі оптимальної складності.

Головна ідея МГВА полягає в наступному: стверджується, що для задачі однократного прогнозу доцільно знизити точність визначення оцінок коефіцієнтів рівняння регресії, але за рахунок цього придати йому більшу регулярність. Тому нашою метою в цій задачі є не мінімізація помилок на вже відомих вузлах інтерполяції, а мінімізація помилок на нових точках, які ми в момент синтезу рівняння регресії ще не мали[44].

Розглянемо полиноміальні алгоритми методу групового врахування аргументу. Послідовність вихідних даних ділиться на перевірочну та навчальну. Як визначено вище – M – кількість вузлів інтерполяції; m — кількість членів повного поліному регресії. При $m > M$ розв'язок можливо отримати тільки за допомогою МГВА.

Нехай повний опис об'єкту задається деякою залежністю $y = \varphi(x_1, x_2, \dots, x_n)$. Замінімо цей вираз декількома рядами часткових описів.

Перший ряд селекції:

$$y_1 = f(x_1, x_2), y_2 = f(x_1, x_3), \dots, y_s = f(x_{n-1}, x_n), s = C_n^2.$$

Другий ряд селекції:

$$z_1 = f(y_1, y_2), z_2 = f(y_1, y_3), \dots, z_p = f(z_{n-1}, z_n), p = C_s^2 \dots$$

Різноманітні алгоритми МГВА розрізняються по виду функції часткового опису $f(\cdot)$. В даній роботі розглядаються три види часткових описів:

лінійний частковий опис —

$$f(x_i, x_j) = A_0 + A_1 x_i + A_2 x_j$$

неповний квадратичний частковий опис —

$$f(x_i, x_j) = A_0 + A_1 x_i + A_2 x_j + A_3 x_i x_j$$

квадратичний частковий опис —

$$f(x_i, x_j) = A_0 + A_1 x_i + A_2 x_j + A_3 x_i x_j + A_4 x_i^2 + A_5 x_j^2,$$

де A_i - нечітке число.

З ряду в ряд селекції за допомогою порогових відборів передається тільки деяка кількість самих регулярних або незміщених змінних. Звичайно ця кількість однакова на всіх рядах селекції, та дорівнює F .

Правило зупинки селекції: ряди селекції нарощуються до тих пір, доки критерій незміщеності розв'язків зменшується, така кількість називається свободою вибору. Для запобігання індуциту селекцію при досягненні мінімуму необхідно зупинити[44].

2.5 Висновки за розділом 2

Отже, у даному розділі були розглянуті головні методи, для навчання рекурентних нейронних мереж та метод групового урахування аргументів, а також їх найпопулярніші підходи до створення архітектури. Завдяки цим методам та моделям нейронні мережи зможуть вирішити конкретну задачу із доволі високою точністю та знайти оптимальний шлях для її вирішення. Також, в залежності від обраного методу, час навчання та точність передбачення можуть змінюватись. У наступному розділі розглянемо архітектури та оберемо відповідну до нашої задачі.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ЗА ДОПОМОГОЮ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

3.1 Загальний огляд програмного продукту

В процесі виконання роботи було розроблено та реалізовано систему прогнозування, яка допомагає виконати прогноз за допомогою різних моделей та обрати найкращу з найбільшою точністю.

Для написання програмного продукту було обрано JupyterLab. JupyterLab – гнучке, інтегроване та легко розширюване середовище, що підтримує одночасну роботу з декількома блокнотами Jupyter, текстовими файлами, датасетами, терміналами та іншими компонентами. Аналогічно класичним IDE у JupyterLab можна впорядковувати документи у робочій області у зручному порядку за допомогою вкладок та роздільників. Головні особливості JupyterLab:

- Використовується метод drag-and-drop для того щоб можна було змінювати осередки і копіювати їх між блокнотами.
- Високий рівень інтеграції між файлами з різним розширенням.
- Блоки коду виконуються інтерактивно прямо з текстових файлів (.py, .R, .md, .tex і т. Д.).
- Консоль коду можна пов'язати з ядром блокнота, щоб вивчати код в інтерактивному режимі, не захаращуючи блокнот тимчасовими правками.
- Можливість редагування популярних форматів файлів з попереднім переглядом в реальному часі, такі як Markdown, JSON, CSV, і інші.

Програмний продукт розроблений на мові програмування Python, тому може бути запущений на будь-якій операційній системі, що є дуже великим плюсом для усіх користувачів. Потрібно лише завантажити середовище та мову програмування на свій ПК.

3.2 Інтерфейс програми

Розглянемо основі функції нашої програми. Головне вікно містить кнопку для запису ІІІ. Користувач має натиснути клавішу «Restart & Run All», ви можете побачити її розташування в інтерфейсі на рисунку 3.1.

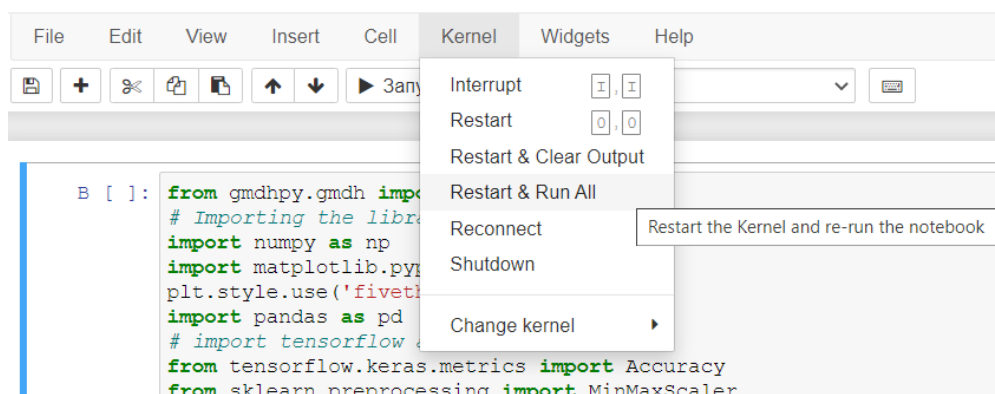


Рисунок 3.1 – Запуск програми

Надалі програма буде виконувати усі дії самостійно, та не буде потребувати додаткового втручання. Розглянемо детальніше, що саме передбачає даний програмний продукт.

3.3 Вхідні дані

У дослідженні використовувались дані за кожен день взяті з криптовалютної біржі. В ході роботи виконувалась побудова прогнозних моделей для визначення ціни закриття («Close»). Було побудовано декілька моделей різними методами. В таблиці 3.1 показано дані, що використовуються.

Таблиця 3.1 – Початкові дані.

Дата	Назва	Символ	High	Low	Open	Close
2013-04-29	Litecoin	LTC	4.573600	4.225640	4.366760	4.383900
2013-04-30	Litecoin	LTC	4.572380	4.168960	4.403520	4.296490
2013-05-01	Litecoin	LTC	4.356860	3.520290	4.289540	3.801010
2013-05-02	Litecoin	LTC	4.039300	3.007170	3.780020	3.371980
...	...	...	...	...	...	...
2021-07-05	Litecoin	LTC	144.849333	134.960263	144.849333	138.073246
2021-07-06	Litecoin	LTC	142.703568	135.924837	137.951668	138.985636

3.4 Експерименти

Отже, програма завантажує усі потрібні файли для використання їх у подальшому розв’язку задачі. Першим результатом аналізу є таблиця, що показує дані вибірки для навчання нейронної мережі. Дані представлені як значення.

Після цього програма виводить графік усієї вибірки на рисунку 3.2, зображуючи тренувальну частину синім кольором, а тестову червоним. За тренувальну частину було взято дані з 2017 до кінця 2020 року, а за тестувальну – 2021 рік. Об’єм усієї вибірки 2991 значень.



Рисунок 3.2 – Графік даних усієї вибірки

Наступний крок – нормалізація даних. Після підготовки усіх даних, програма готує вибірку навчальних та тестових даних та нейронна мережа починає навчання (рисунок 3.3).

```
Epoch 3/20
2708/2708 [=====] - 33s 12ms/step - loss: 0.0094
Epoch 4/20
2708/2708 [=====] - 31s 12ms/step - loss: 0.0069
Epoch 5/20
2708/2708 [=====] - 33s 12ms/step - loss: 0.0060
Epoch 6/20
2708/2708 [=====] - 32s 12ms/step - loss: 0.0052
Epoch 7/20
2708/2708 [=====] - 32s 12ms/step - loss: 0.0045
Epoch 8/20
2708/2708 [=====] - 31s 12ms/step - loss: 0.0041
Epoch 9/20
2708/2708 [=====] - 32s 12ms/step - loss: 0.0040
```

Рисунок 3.3 – Навчання нейронної мережі

Наприкінці роботи, програма показує графіки реального значення та передбаченого та значення помилки та точність усієї моделі кожного типу нейронної мережі (рисунок 3.4-3.6). Для LSTM було побудовано три моделі.

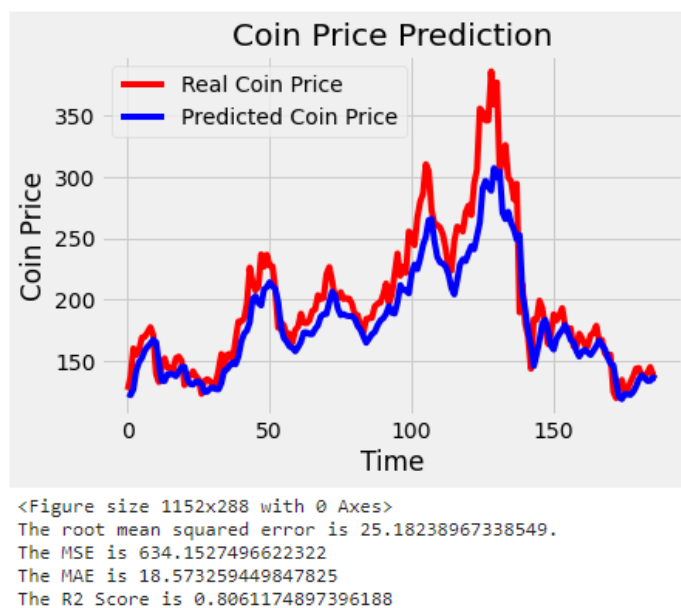


Рисунок 3.4 – LSTM-1

(Дана модель має 3 шари, кожен з яких має 50 нейронів, на кожному
непарному Dropout – 0.2)

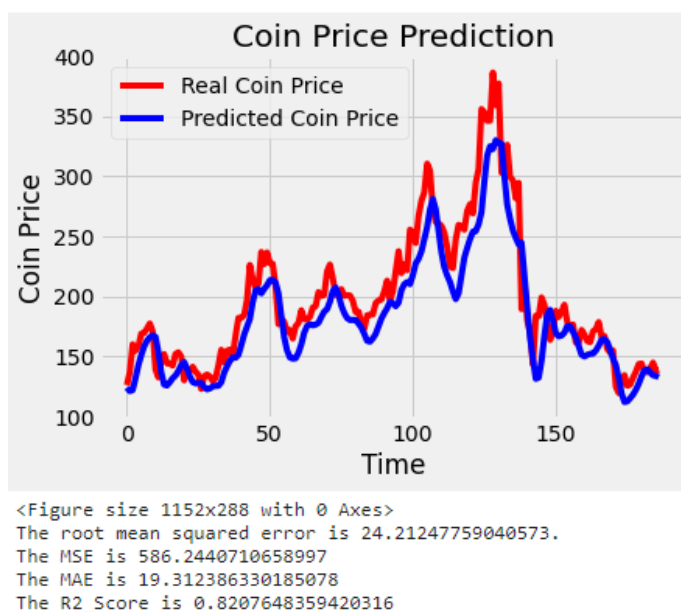


Рисунок 3.5 – LSTM-2

(Дана модель має 4 шари, кожен з яких має 100 нейронів, на кожному
непарному Dropout – 0.2)

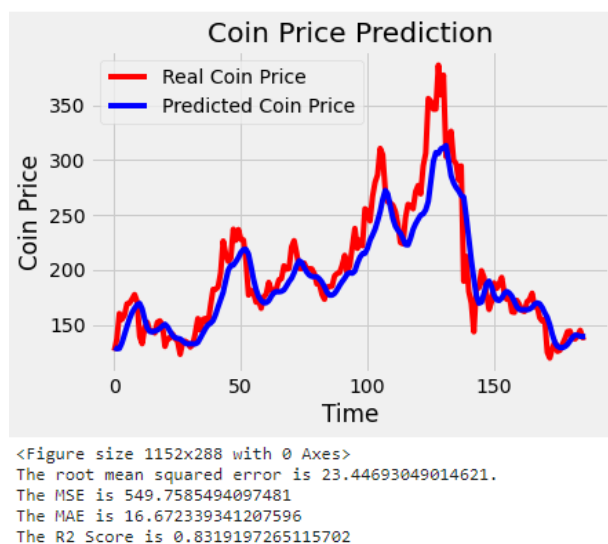


Рисунок 3.6 – LSTM-3

(Дана модель має 4 шари, кожен з яких має 30 нейронів, на кожному непарному Dropout – 0.1)

Найкращою з цих трьох моделей є остання модель, що складається з чотирьох LSTM шарів(які мають по 30 нейронів та кожен шар має регуляризацію Dropout – 0.1) та одного вихідного. Час навчання займав приблизно 7 хвилин.

Для GRU було побудовано також три моделі. Помилка та точність моделі представлена нижче на рисунку 3.7-3.9.

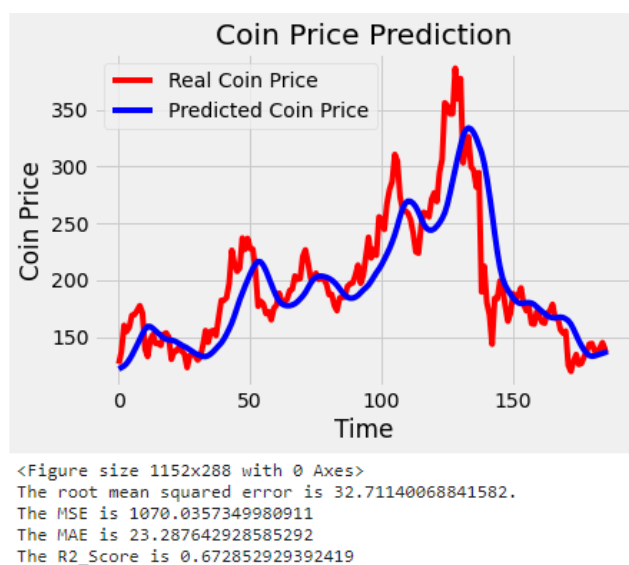


Рисунок 3.7 – GRU-1

(Дана модель має 4 шари, кожен з яких має 150 нейронів та Dropout – 0.2 на кожному шарі)

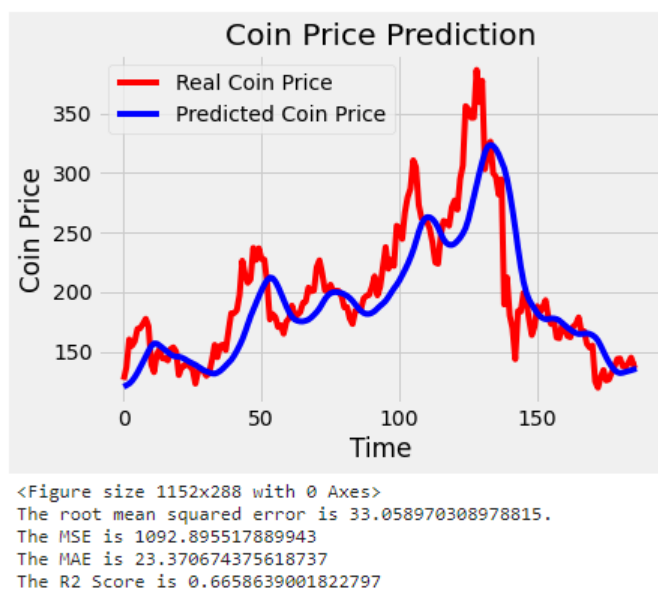


Рисунок 3.8 – GRU-2

(Дана модель має 4 шари, кожен з яких має 80 нейронів та Dropout – 0.5 на кожному шарі)

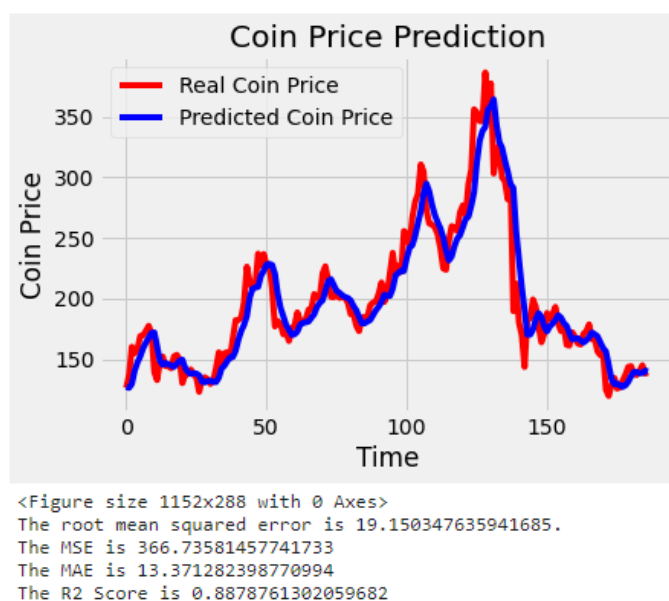


Рисунок 3.9 – GRU-3

(Дана модель має 2 шари, кожен з яких має 120 нейронів та Dropout – 0.2 на кожному шарі)

Найкраще вдалася третя мережа, що складається з двох GRU шарів(які складаються кожен з 120 нейронів та Dropout 0.2 на кожному шарі) та один вихідний шар. Також використовувався оптимізатор SGD. Час навчання займав приблизно 5 хвилин.

Для звичайної RNN було побудовано також три моделі, значення помилки та точність наведені нижче на рисунку 3.10-3.12.

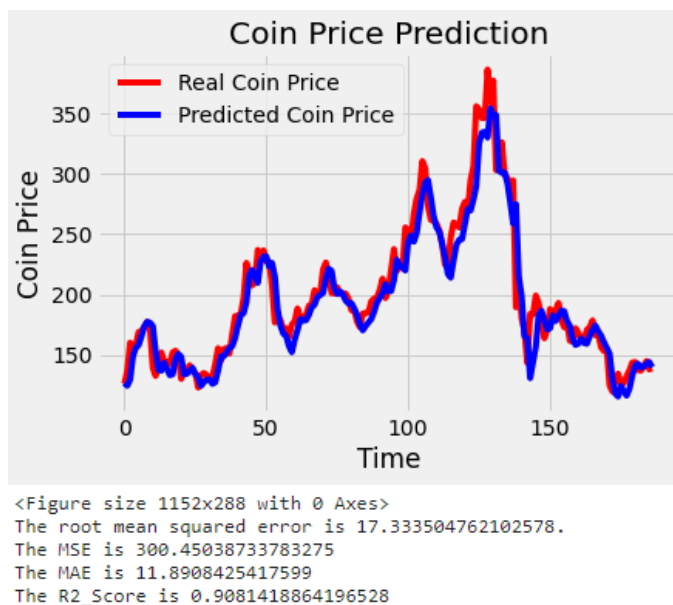


Рисунок 3.10 – SimpleRNN-1

(Дана модель має 4 шари, кожен з яких має 30 нейронів та Dropout – 0.1 на кожному шарі)

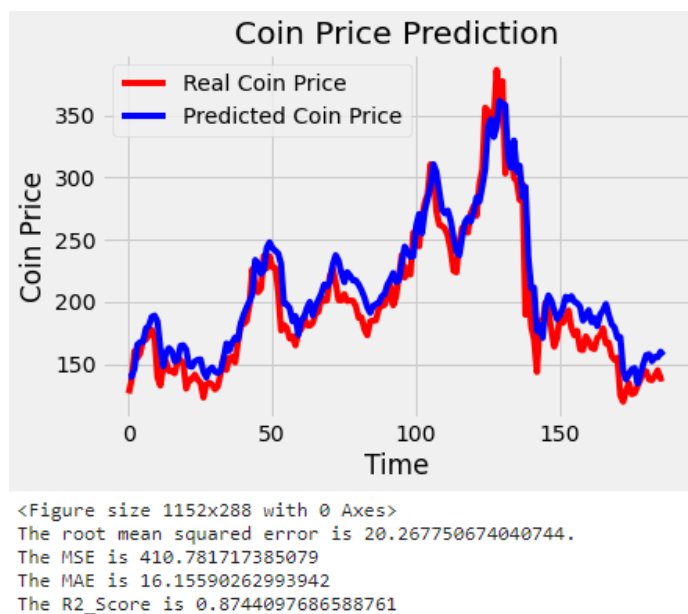


Рисунок 3.11 – SimpleRNN-2

(Дана модель має 4 шари, кожен з яких має 50 нейронів та Dropout – 0.15 на кожному шарі)

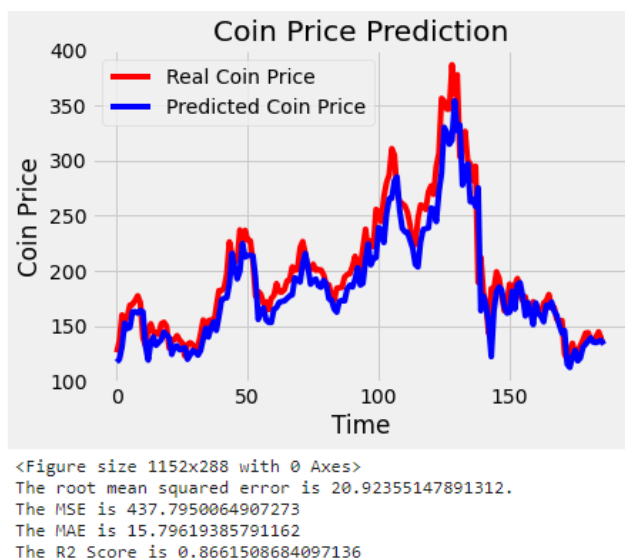


Рисунок 3.12 – SimpleRNN-3

(Дана модель має 3 шари, кожен з яких має 100 нейронів та Dropout – 0.3 на кожному шарі)

Найкращою з трьох моделей є перша, яка складається з 4 шарів по 30 нейронів та вихідного. Кожен шар має регуляризацію Dropout – 0.1. Оптимізатор використовувався RMSprop. Час навчання – приблизно 18 хвилин.

Зараз побудуємо моделі Bidirectional LSTM моделі, їх також буде три, значення помилки та точність наведені нижче на рисунку 3.13-3.15.

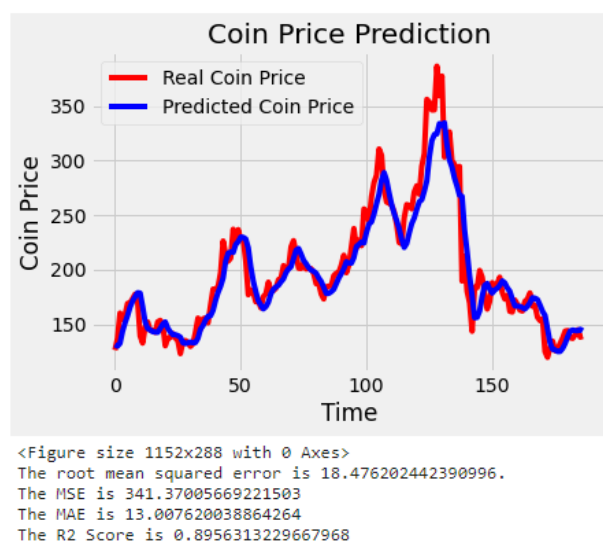


Рисунок 3.13 – BiLSTM-1

(Дана модель має 3 Bidirectional LSTM шари по 50 нейронів та один шар LSTM з 50 нейронами, Dropout – 0.2 на кожному шарі)

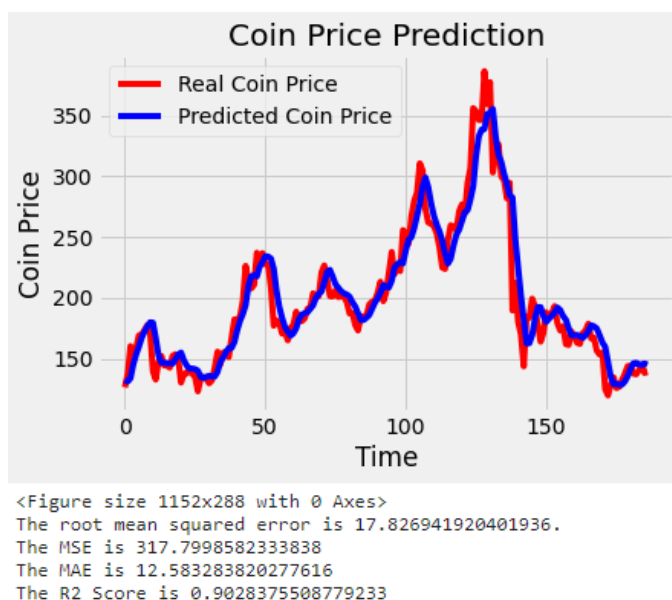


Рисунок 3.14 – BiLSTM-2

(Дана модель має 3 Bidirectional LSTM шари по 100 нейронів та один шар LSTM з 100 нейронами, Dropout – 0.5 на кожному шарі)

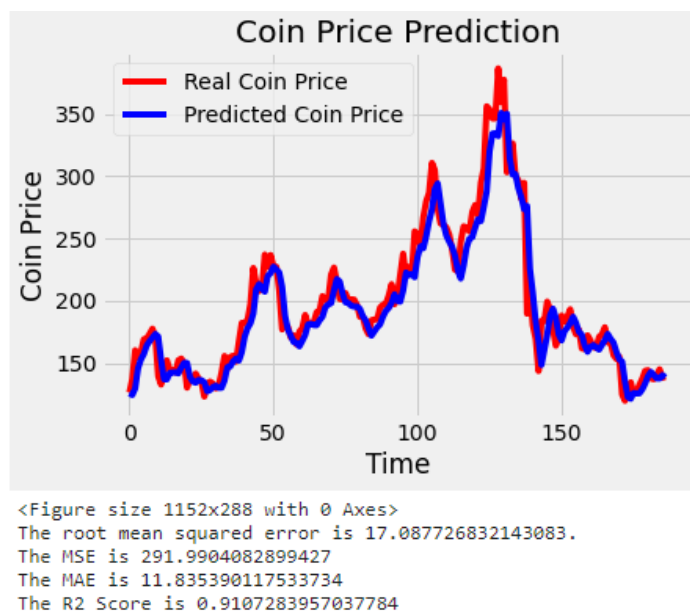


Рисунок 3.15 – BiLSTM-3

(Дана модель має 2 Bidirectional LSTM шари по 150 нейронів та один шар LSTM з 150 нейронами, Dropout – 0.3 на кожному шарі)

Найкращою з трьох моделей є остання, яка складається з 2 Bidirectional LSTM шарів по 150 нейронів, одного шару LSTM з 150 нейронами та вихідного. Кожен шар має регуляризацію Dropout – 0.3. Оптимізатор використовувався Adam. Час навчання – приблизно 6 хвилин.

Для Bidirectional GRU було побудовано також три моделі, значення помилки та точність наведені нижче на рисунку 3.16-3.18.

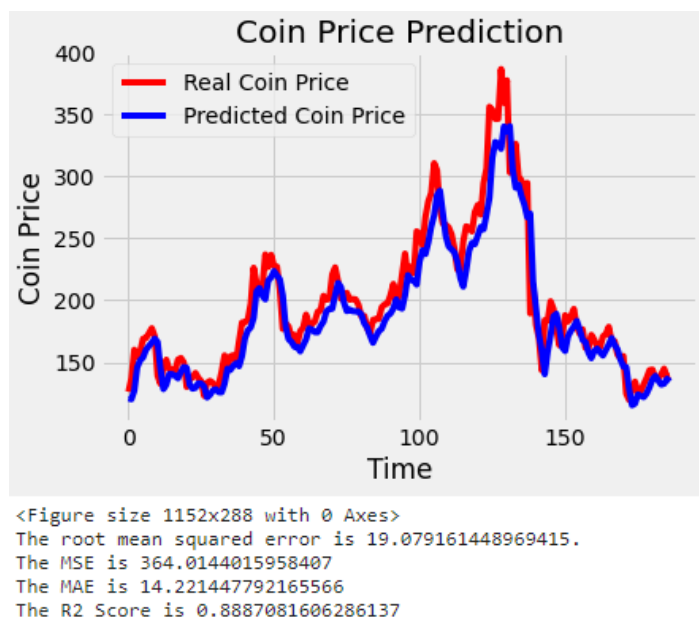


Рисунок 3.16 – BiGRU-1

(Дана модель має 3 Bidirectional GRU шари по 50 нейронів та один шар GRU з 50 нейронами, Dropout – 0.4 на кожному шарі)

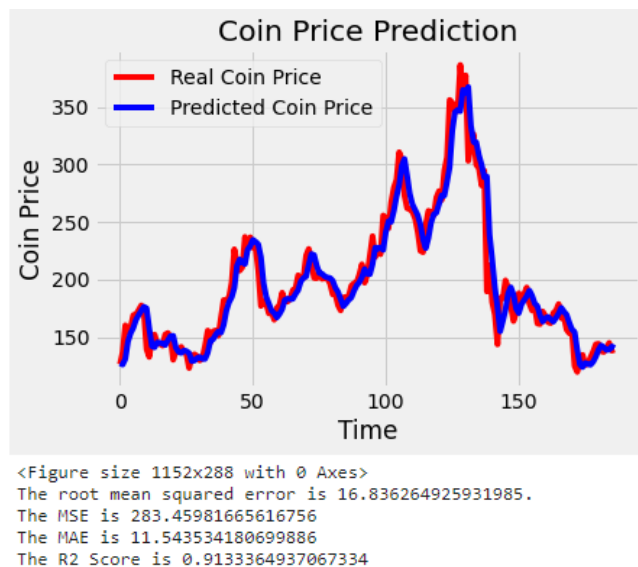


Рисунок 3.17 – BiGRU-2

(Дана модель має 3 Bidirectional GRU шари по 100 нейронів та один шар GRU з 100 нейронами, Dropout – 0.4 на кожному шарі)

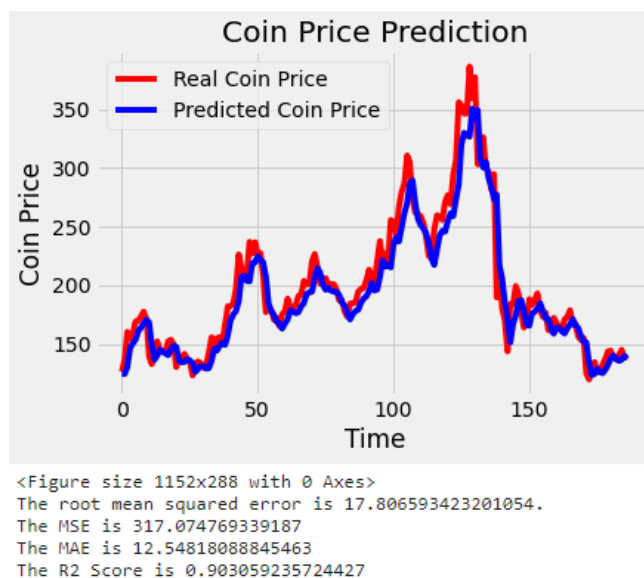


Рисунок 3.18 – BiGRU-3

(Дана модель має 2 Bidirectional GRU шари по 30 нейронів та один шар GRU з 30 нейронами, Dropout – 0.15 на кожному шарі)

Найкращою з трьох моделей є друга, яка складається з 3 Bidirectional GRU шарів по 100 нейронів, одного шару GRU з 100 нейронами та вихідного. Кожен шар має регуляризацію Dropout – 0.4. Оптимізатор використовувався RMSprop. Час навчання – приблизно 9 хвилин.

В наступному експерименті додамо до моделей механізм уваги(Attention Mechanism). Таких моделей буде три, значення помилки та точність наведені нижче на рисунку 3.19-3.21.

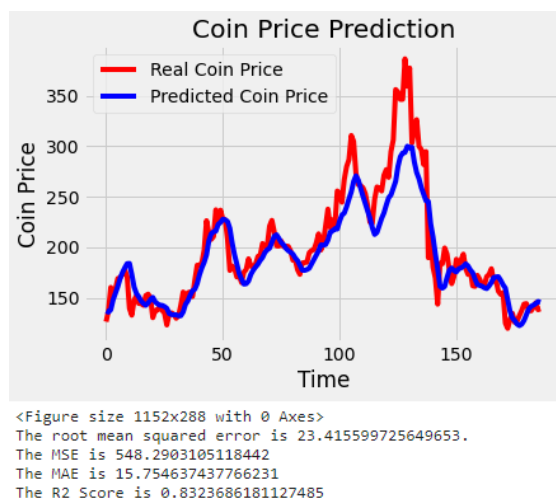


Рисунок 3.19 – BiLSTM with Attention – 1

(Дана модель має 3 Bidirectional LSTM шари по 32 нейрони, Attention шар та один шар LSTM з 32 нейронами, Dropout – 0.2 на кожному шарі)

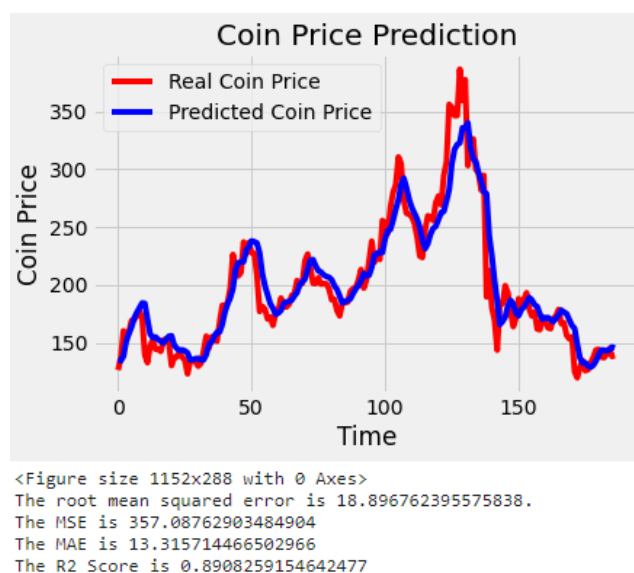


Рисунок 3.20 – BiLSTM with Attention – 2

(Дана модель має один Bidirectional LSTM шар з 50 нейронами, Attention шар та один шар LSTM з 50 нейронами, Dropout – 0.2 на кожному шарі)

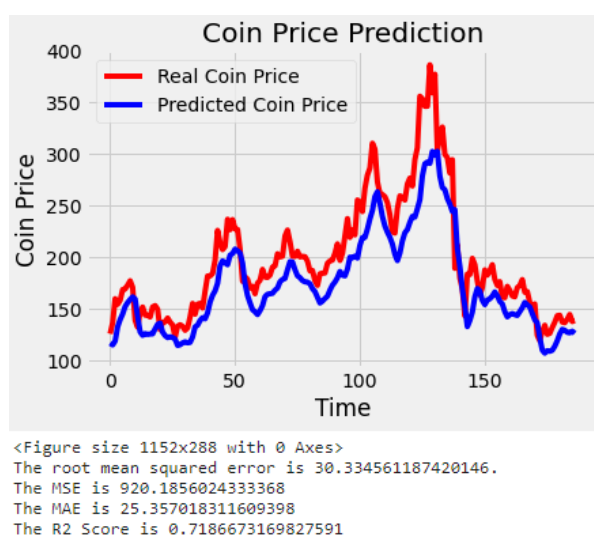


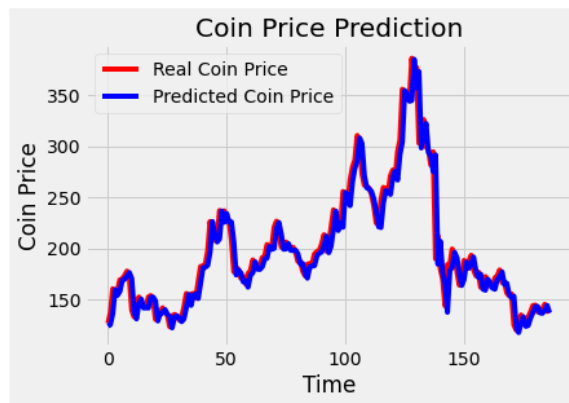
Рисунок 3.21 – BiLSTM with Attention – 3

(Дана модель має 2 Bidirectional LSTM шари по 100 нейронів, Attention шар та один шар LSTM з 100 нейронами, Dropout – 0.1 на кожному шарі)

Найкращою з трьох моделей є друга, яка складається з одного Bidirectional LSTM шарів по 50 нейронів, Attention шару, одного шару LSTM з 50 нейронами та вихідного. Кожен шар має регуляризацію Dropout – 0.2. Оптимізатор використовувався Adam. Час навчання – приблизно 6 хвилин.

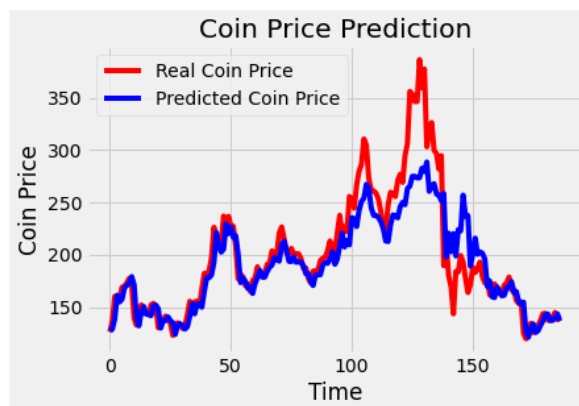
Наступним кроком виконується навчання мережі на основі МГУА.

Було побудовано три моделі, значення помилок та точність зображені нижче на рисунку 3.22, 3.23 та 3.24.



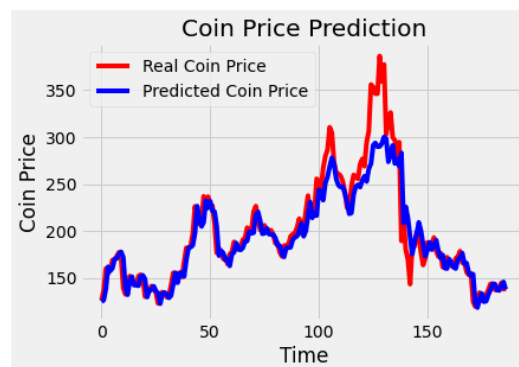
<Figure size 1152x288 with 0 Axes>
 The root mean squared error is 15.98896984109779.
 The MSE is 255.64715657953468
 The MAE is 10.456081474272741
 The R2_Score is 0.9218397894119849

Рисунок 3.22 – GMDH-1



<Figure size 1152x288 with 0 Axes>
 The root mean squared error is 26.85617024136722.
 The MSE is 721.2538800332982
 The MAE is 17.232901961603986
 The R2_Score is 0.7794876504590138

Рисунок 3.23 – GMDH-2



<Figure size 1152x288 with 0 Axes>
 The root mean squared error is 21.324906444849073.
 The MSE is 454.7516348815656
 The MAE is 13.09046394826319
 The R2_Score is 0.8609666384592494

Рисунок 3.24 – GMDH-2

Дані моделі відрізняється один від одного розміром вікна спостережень. У першій це 7, у другій 30 а у третій 14. Перша модель показала себе краще за інші. Вікно моделі складалося з 7 спостережень, кількість передаваних моделей на наступний рівень дорівнює 7, значення регуляризації L2 – 0,5.

Повний список параметрів зображено нижче на рисунку 3.25. Час навчання зайняв приблизно 4 хвилини.

```
ref_functions=('cubic','quadratic','linear_cov','linear'),
criterion_type='validate',
criterion_minimum_width=7,
stop_train_epsilon_condition=0.0001,
layer_err_criterion='top',
l2=0.5,
n_jobs='max')
```

Рисунок 3.25 – Список параметрів моделі заснований на МГУА

Проведемо порівняння найкращих моделей. Спочатку порівняємо змінення значення помилки на кожній епосі(рисунки 3.26 та 3.27).

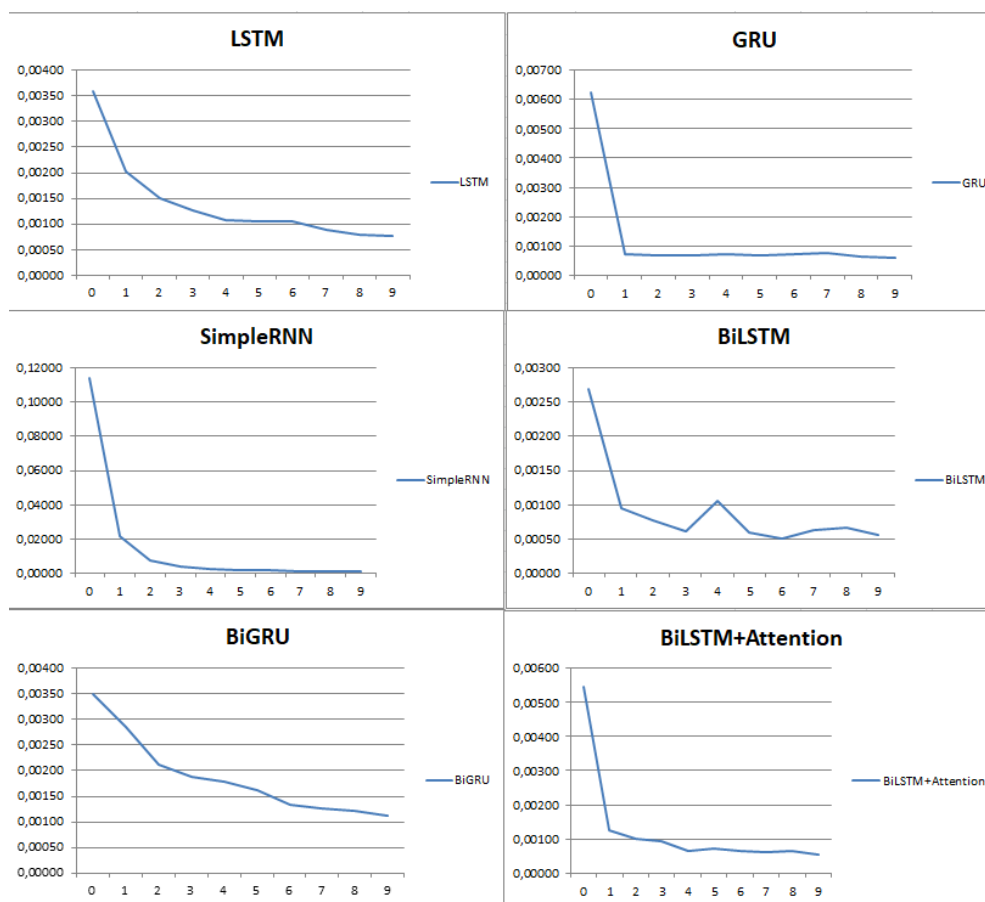


Рисунок 3.26 – Значення помилки(MSE) на кожній епосі для кожної моделі

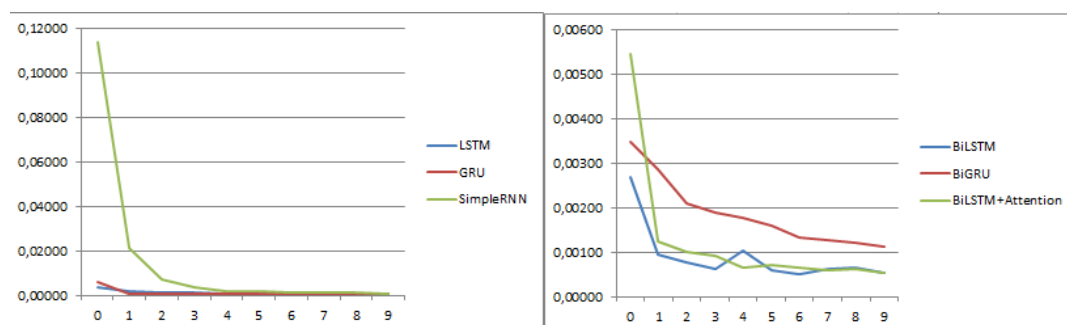


Рисунок 3.27 – Значення помилки(MSE) на кожній епосі для кожної моделі

В таблиці 3.2 наведено результати навчання найкращих мереж.

Таблиця 3.2 – Результати навчання мережі із власною архітектурою

	Оцінки якості моделей		Оцінки якості прогнозів			Learn_Time
	R2_Score	MSE	RMSE	MAE	MAPE	
LSTM	86%	457.48	21.3	14.89	7.74	7 minutes
GRU	88%	563.09	23.72	16.95	8.30	5 minutes
SimpleRNN	90,8%	266.55	16.32	11.34	5.77	18 minutes
BiLSTM	91%	291.99	17.08	11.83	6.91	6 minutes
BiGRU	91%	283.45	16.83	11.54	6.98	9 minutes
BiLSTM+Attention	89%	410.12	20.25	14.99	6.70	6 minutes
GMDH	92,18%	255.56	15.98	10.44	5.24	4 minutes

Нижче в таблиці 3.3 наведено частково вхідні дані та прогнозовані.

Таблиця 3.3 – Порівняння прогнозованих та реальних даних

Real	Predicted
126.23	123.55
136.94	125.05
160.19	135.9
154.8	159.2
158.59	153.74
169.016	157.57
...	...

3.5 Висновок до третього розділу

У ході виконання даної роботи, були досліджені різноманітні типи рекурентних мереж та метод заснований на алгоритмі МГУА для виконання прогнозування на ринку криптовалют. Після виконання порівняння цих мереж, було вирішено, що найкращою з усіх навчених у даній роботі мереж є мережа заснована на алгоритмі МГУА. Тому, що вона показала кращий результат у порівнянні з іншими моделями та була швидшою у навчанні.

У результаті дослідження був розроблений програмний продукт, що виконує прогноз із точністю 92,18%.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Вступ та постановка задачі стартап проекту

На сьогодні великої популярності набуває такий вид підприємництва як стартап. Стартап-проект — є комерційним проектом, який знаходиться в стані розробки, або нещодавно вийшов на ринок. Характерною особливістю стартапу, що відрізняє його від малого бізнесу, є оригінальність та інновації, він не може бути копією вже реалізованих ідей. При цьому проект не обов'язково повинен бути масштабного характеру, головне, щоби він був креативним, а його завдання — спрощувати людям будь-які дії в їх повсякденному житті.

Наразі, з появою Інтернету та сучасних технологій, стало простіше заходити на ринок, знаходити інвесторів та споживачів. З'явилося набагато більше можливостей для розвитку свого проекту за кордоном, ніж раніше. Проте розробка стартапу є досить ризикованим завданням. Не всім вдається довести свій стартап-проект до ринкового впровадження. За статистикою успіху досягає лише 10–20 % від усіх стартап-проектів.

Запуск стартапу передбачає цілий ряд обов'язкових дій, у межах яких визначають ринкові перспективи стартапу, графік розробки, принципи організації виробництва, заходи з залучення інвесторів та аналіз ризиків.

Різно постає питання у створенні диверсифікованих інвестиційних портфелів. Одним зі способів інвестування можна розглядати криптовалюту. Робота дозволяє побудувати відповідний портфель, взявши за основу описану систему прогнозування.

Для розробки стартап проекту та виведення його на ринок необхідно провести детальне дослідження, яке передбачає виконання наведених нижче чотирьох кроків.

Здійснити маркетинговий аналіз стартап-проекту, в рамках якого:

- 1) розробити опис ідеї проекту, визначити основні напрямки використання товару чи послуги та сформулювати основні відмінності від

товарів/послуг конкурентів;

- 2) проаналізувати ринкові можливості для його реалізації;
- 3) розробити стратегію виведення товару на ринок базуючись на аналізі ринкового середовища.

Організація стартап-проекту, яка включає такі кроки:

- 1) скласти календарний план реалізації та запуску стартап-проекту;
- 2) визначити плановий обсяг виробництва потенційного товару та н його основі розрахувати потребу у матеріальних ресурсах і персоналі;
- 3) розрахувати витрати, необхідні для реалізації проекту, та витрати на запуск проекту.

Виконати фінансово-економічний аналіз та оцінити ризики стартап проекту, в межах якого:

- 1) визначити обсяг інвестиційних витрат;
- 2) 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);

3) визначити основні ризики проекту та способи для їх запобігання. Розробити заходи з комерціалізації проекту. Цей етап націлений на пошук фінансування проекту та просування інвестиційної пропозиції.

Для його досягнення необхідно:

- 1) визначити цільову групу інвесторів та описати їх бізнес інтереси;
- 2) скласти інвестиційну пропозицію: стислий опис проекту для ознайомлення інвестора із стартап-проектом;

3) визначити основні канали та заходи для просування офerti інвесторам.

4.2 Карта стартап проекту

Стартап проект полягає у створенні системи прийняття рішень для автоматичного проведення торговельних операцій на ринку криптовалют. Така програма аналізує масиви даних, після чого формує біржовий приказ оптимальний, обґрунтований історичними даними. На основі даної системи пропонується створити інвестиційний фонд у відповідні криптовалютні активи.

В таблиці 4.1 представлено основна інформація проекту, розкрито основну ідею та рішення поставленої проблеми.

Таблиця 4.1 – Інформаційна карта проекту

1. Назва проекту	Система інвестування на ринку криптовалют на основі методів машинного навчання
2. Коротка анотація	Система дозволяє керувати активами користувача і приймає статистично обґрунтоване рішення спрямована на збільшення об'єму капіталу інвестиційного портфеля користувача.
3. Термін реалізації проекту	6 місяців

Продовження таблиці 4.1

4. Необхідні ресурси	Обладнання – комп'ютер. Програмне забезпечення, операційна система, антивірусне обладнання. Електрика, газ, водопостачання, Інтернет. Фінансові ресурси – заробітна плата працівникам на 6 місяців роботи, гроші на на оплату комунальних послуг, оренди, реклами тощо. Приміщення з усіма необхідними комунікаціями.
5. Опис проблеми, які вирішує проект	Дана комплексна система дозволяє забезпечити прийняття інвестиційних рішень. Система позбавлена суб'єктивності та забезпечує спосіб прийняття рішень аналогічний до поглядів інвестора. Система працює безпосередньо з акаунтом користувача і не має повного доступу до фінансів, що забезпечує захист даних користувача.
6. Головні цілі та завдання проекту	Основна мета проекту – доведення ефективності обраної системи та розробленого продукту. Додаткові завдання – новий досвід, розробка комплексної системи, робота із реальними даними та створення комерційно успішного продукту.

Продовження таблиці 4.1

7. Очікувані результати	Автоматична система прийняття рішень, здатна працювати на необмеженій кількості користувачів в автономному режимі з допоміжними інструментами побудованих на статистиці, для спрощення аналізу та прийняття рішень.
-------------------------	---

4.3 Технологічний аудит ідеї проекту

Далі буде виділено основу ідею стартапу та зведено її в наступній таблиці 4.2.

Таблиця 4.2 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Дана комплексна система дозволяє побудувати власну інвестиційну стратегію що є легальним шляхом для будь-якого українця забезпечити стабільні накопичення та їх зростання.	1. Робота в якості компанії по управлінню активами, що надаються користувачем	Система дозволяє повністю автоматизувати дохід користувача, компанія несе відповідальність за кошти користувачів.
	2. Робота в якості управляючої компанії за обмеженим доступом до акаунтів користувачів.	Система дозволяє повністю автоматизувати дохід користувача. Моніторинг дохідності покладається на користувача. Користувач в будь-який момент може вивести кошти або відмовитись від послуг компанії.

Далі було визначено конкурентів на ринку і зроблено порівняльний аналіз програмних продуктів конкурентів, виявлено їх переваги та недоліки. Також було представлено перелік переваг над існуючими програмними

рішеннями (таблиця 4.3, 4.4). Варто відмітити відсутність конкурентів на ринку України тому порівняння було зроблене з американськими конкурентами.

Таблиця 4.3 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п / п	Техніко Економі чні характер истики ідеї	(потенційні) товари/концепції конкурентів				W (слаб касто рона)	N (нейт рала ьна)	S(с иль на)
		Мій проект	PhiLA bAn dreess en Horow itz	Fortr ess	Flint Capit al			
1.	Точність прогнозу вання	Застосування моделі, що безпосередньо оцінює ймовірність його потенційної привабливості	Власн ий алгори тм	Власн ий алгор итм	Власн ий алгор итм			+
2.	Простота управлін ня капітало м інвестора (швидкіс ть зняття грошей інвесторо м тощо)	Все зводиться до отримання сигналу на укладання угоди та аналізу її доцільності з допомогою інструментів	Проце ду ра складн а	Спро щена проце дура залуч ення капіта лу	Проц еду ра склад на			+

Продовження таблиці 4.3

3.	Ризики невірного прогнозу	Існують, через велику кількість факторів.	Невідомо (комерційна таємниця)	Невідомо (комерційна таємниця)	Невідомо (комерційна таємниця)		+	
4.	Доступність /Зручність	Зручний веб- застосунок, який потребує обчислювальних мінімум ресурсів	Власний інтерфейс	Власний інтерфейс	Власний інтерфейс		+	

Таблиця 4.4 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Створення комплексної системи керування активами	Використання мови програмування C#	Потрібні доопрацювання	Не доступні
2.	користувача	Використання мови програмування Python	Наявна	Не доступні

Продовження таблиці 4.4

3.		Використання мови програмуванн я Pine	Наявна	Доступні
Обрана технологія реалізації ідеї проекту: мова програмування Pine				

4.4 Аналіз ринкових можливостей запуску стартап-проекту

У таблиці 4.5 зображено попередню характеристику потенційного ринку стартап-проекту.

Таблиця 4.5 – Попередня характеристика потенційного ринку стартап-проекту.

№ п/ п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	0
2	Загальний обсяг продаж	3 млн \$
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	12%

Після аналізу ринку можна зробити висновок, що він є сприятливим для створення програмного продукту, оскільки динаміка ринку позитивна, а конкуренти відсутні.

Наступним кроком необхідно охарактеризувати основні групи потенційних користувачів продукту і скласти опис вимог кожної такої групи (таблиця 4.6).

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту.

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Необхідність пенсійних накопичень	громадяни України в віці від 18 до 50 років	Незначна кількість початкового капіталу.	Простота використання, простота моніторингу.
2	Збереження бюджету установи та його примноження	Малі підприємства, селищні ради, громадські бюджети	Цікавить короткострокове інвестування.	Великий обсяг, гнучкість капіталу.
3	Довгострокове інвестування.	Великі інвестори	Цікавить передусім відсутність ризику	Високі вимоги до контролю капіталу.

Далі необхідно проаналізувати можливі загрози, що можуть виникнути на етапі виведення продукту на споживацький ринок і перешкодити успішному запуску проекту. Результати представлені у таблиці 4.7.

Таблиця 4.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Конкуренція	Доволі незначний рівень виходу на ринок, можливий вихід нових компаній протягом 3 наступних років	Пришвидшити вихід нових програмних продуктів
2.	Збут	Ускладнення збуту через недовіру користувачів	Розміщення додаткових рекламних банерів в інтернеті, розширена наукова база.

Також необхідно розглянути можливі фактори, що навпаки сприятимуть запуску проекту. Результати представлені у таблиці 4.8.

Таблиця 4.8 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Відсутність конкуренції	Бути першим на ринку	Розширення можливостей продукту, концентрація на якості
2	Створення позитивного іміджу компанії.	Надання послуг на найвищому рівні, забезпечення задоволення клієнтів.	Створення якісної рекламної кампанії, техпідтримка існуючих клієнтів.

Далі необхідно охарактеризувати конкурентне середовище, а саме визначити тип та рівень конкуренції. Результати аналізу наведено у таблиці 4.9.

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1.Вказати тип конкуренції - Досконала конкуренція	Багато систем/команд аналітиків	Розробити впізнаваний продукт, якість, що вирізнятиме нас від конкурентів
2. За рівнем конкурентної боротьби: міжнародний	На ринку присутні системи, розроблені за кордоном.	Розширення аудиторії, розширення списку мов, які підтримуються системою
3. За галузевою ознакою - внутрішньогалузева	_____	_____
4. Конкуренція за видами товарів: товарно-родова	Конкуренція між прогнозами інформаційних систем та команд аналітиків.	Збільшення точності та швидкості відправки біржових приказів
5.За характером Конкурентних переваг: Нецінова	Різні способи прогнозування дають різну точність	Розробка кращих(точних) алгоритмів
6. За інтенсивністю: марочна	Впізнаваний бренд надає великих переваг	Велику увагу приділити розвитку бренду

Далі необхідно виконати детальний аналіз конкуренції за моделлю 5 сил конкуренції Майкла Портера, яка використовується для розуміння структури галузі, аналізу її привабливості з точки зору отримання прибутку, оцінки конкуренції і розробки стратегії бізнесу. Результати аналізу зведено в таблицю 4.10.

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером.

	Прямі конкурен ти в галузі	Потенційні конкуренти	Постачальн ик и	Клієнти	Товари- замінник и
Складо ві аналізу	Інші комплекс ні системи	Гнучкі ціни, розмір капіталовкла день	Фактори сили постачальни ків	Контрол ь якості, система інформа ц ії	Ціна, лояльніс т ь спожива ч ів
Висновк и:	Інтенсивн а конкурен ція можлива в майбутнь о му	Є як можливості входження на ринок, так і нові потенційні конкуренти	Постачальни к и відсутні	Клієнти не диктуют ь умови роботи на ринку	Товари Замінник и відсутні

Результати аналізу конкурентного середовища підтверджують, що на ринку сприятлива ситуація для створення і запуску даного стартап-проекту. Ґрунтуючись на проведеному аналізі конкуренції (таблиця 4.10), а також враховуючи 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 та 4.8) було сформульовано та обґрунтовано перелік факторів конкурентоспроможності. Аналіз оформлено в таблицю 4.11.

Таблиця 4.11 – Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
Низька конкуренція	В нашій країні на час розробки стартапу не було виявлено конкурентів
Доступність програмного продукту	Розроблений продукт є загальнодоступним і кросплатформним. Для доступу необхідне підключення до мережі Інтернет.
Зручність використання	Інтерфейс надається власне біржею користувача, який вже знайомий користувачу.

Після проведення аналізу можна виділити сильні та слабкі (які потребують вдосконалення) сторони 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	Низька конкуренція	20				+			
2	Доступність програмного продукту	20				+			
3	Зручність використання	18		+					

Завершальним етапом аналізу ринкових можливостей для запуску проекту є складання SWOT-аналізу. Він дозволяє оцінити можливості та

загрози бізнесу, а також сильні і слабкі сторони продукту. Результати наведені у таблиці 4.13.

Таблиця 4.13 – SWOT- аналіз стартап-проекту

Сильні сторони: відсутність конкурентів; дружній інтерфейс; не вимагає спеціальних навичок/знань для використання.	Слабкі сторони: немає налагодженої клієнтської бази.
Можливості: розширення списку мов для аналізу; додавання нових предметних областей; інтеграція з іншими програмними системами.	Загрози: поява конкурентів; збут .

На основі SWOT-аналізу було спроектовано 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 (таблиця 4.14).

Таблиця 4.14 – Альтернативи ринкового впровадження стартап проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Швидкий вихід на ринок із «сирим» продуктом, можливі проблеми із точністю прогнозу та універсальністю	10%	3 місяці
2	Поступовий вихід з готовим продуктом. Висока якість та конкурентоспроможна ціна.	25%	6 місяців

Отже, в результаті детального аналізу ринкового та конкурентного середовища, факторів загроз та можливостей, сильних та слабких сторін продукту можна зробити висновок, що на ринку склалися сприятливі умови для впровадження товару і, що даний товар відповідає вимогам користувачів.

4.5 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії перш за все передбачає визначення стратегії охоплення ринку. Для цього було охарактеризовано цільові групи потенційних споживачів (таблиця 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційн их клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Громадяни, що прагнуть створити накопиченн я	Низька готовність	5%	Низька	Середня
2	Малі та середні підприємст ва	Висока	15%	Низька	Висока
3	Великі компанії з власними інвесторам и	Висока	30%	Висока	Висока
Які цільові групи обрано: 1,2					

Стратегією охоплення ринку було обрано недиференційований (масовий) маркетинг — компанія концентрує свої зусилля одразу на всіх сегментах споживачів. Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиці 4.16, 4.17, 4.18).

Таблиця 4.16 – Визначення базової стратегії розвитку

п/п	Альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
	1 та 2	Стратегія недиференційованого маркетингу	Висока універсальність, багатогалузевість, висока якість, ціна.	Масовий маркетинг

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*
Так	Так	Ні	Стратегія виклику лідера

Таблиця 4.18 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають формувати комплексну позицію власного проекту (три ключових)
Легкість у використанні, висока точність аналізу	Позиціонування як продукту, що використовує передові технології для аналізу	Якість прогнозу, універсальність.	По іміджу Позиціонування на низькій ціні Легкість у використанні

4.6 Висновки до 4 розділу

В даному розділі було повністю виконано перший етап розроблення стартап проекту, а саме, виконано маркетинговий аналіз стартап проекту. За допомогою нього можна сказати, що існує можливість ринкової комерціалізації проекту, адже на ринку відсутні конкуренти, до того ж рентабельність роботи є досить високою. З огляду на потенційну групу клієнтів, а саме, громадян, малий бізнес, та інноваційність технології є великі перспективи впровадження даного програмного забезпечення у вигляді колючого продукту фонду заощаджень.

Отже, робота може бути розглянута як стартап що спрямований на стратегії голубих океанів (не зайнятих ринків). Про це свідчить відсутність конкурентів та величезна потенційна аудиторія. На цій основі вважаю необхідним продовжувати досліджувати ринок криптовалют з позиції інвестиційної привабливості та прогнозування.

ВИСНОВОК

В ході виконання дисертаційної роботи було досліджено результати застосування рекурентних нейронних мереж до прогнозування цін на ринку криптовалют на основі даних криптовалютної біржі за 2017-2021 рік, вивчено сучасні архітектури рекурентних нейромереж, такі як LSTM, GRU, SimpleRNN, Bidirectional LSTM, Bidirectional GRU, використання механізму уваги у мережі та МГУА.

Проведено порівняльний аналіз сучасних модифікацій рекурентних нейронних мереж LSTM та GRU, таких як двонаправлені (bidirectional) LSTM та GRU, вказаних моделей з додаванням механізму уваги (attention-based mechanism), методів машинного навчання, а також методу групового врахування аргументів (МГУА).

Було встановлено, що найкращі результати демонструє метод МГУА до вхідних даних за даний період. До того ж застосування різних видів РНМ до тих самих даних дає також гарні результати проте демонструє меншу точність.

Практичним результатом роботи є прогноз ціни криптовалюти Litecoin на кінець липня 2021 року.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мащенко П.Л., Пилипенко М.О. Технология Блокчейн и ее практическое применение. Иваново: Олимп, 2017. 64с.
- 2.Greenberg A. Crypto Currency. 20.04.2011. URL: <https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html?sh=2ffb0356353e> (last accessed: 16.11.2021)
3. Miers I., Garman C., Green M., Rubin. A., et. al. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. Berkley: The Johns Hopkins University Department of Computer Science, 2013. 411p.
4. Хажиахметова Е.Ш. Криптовалюта - деньги XXI века. Челябинск: Агентство международных исследований, 2016. 179с.
5. Archer P. The Genesis Block "What is Bitcoin Mining?". 26.05.2013. URL: <http://thegenesisblock.com/mining-for-dummies/> (last accessed: 01.12.2021)
6. Щербик Е.Е. Феномен криптовалют: опыт системного описания. Нижневартонск: Концепт, 2017. 64с.
7. Ахметов А. Как коронавирус повлиял на блокчейн и криптовалюту. 26.06.2020. URL:<http://kabar.kg/news/kak-koronavirus-povliial-na-blokchein-i-kriptovaliutu/> (дата звернення: 01.12.2021)
8. Гафаров Ф.М. Искусственные нейронные сети и приложения: Казань: Изд-во Казан. ун-та, 2018. 121 с.
9. Будыльский, Д. В. GRU и LSTM: современные рекуррентные нейронные сети. Казань: Молодой ученый, 2015. 54с.
- 10.Aggarwal R. Bi-LSTM. 04.07.2019. URL: <https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0> (last accessed: 22.11.2012)
11. Schuster M., Kuldip K. P. Bidirectional recurrent neural networks. Chicago: IEEE Transactions, 1997. 309p.

12. J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks. *Computer Vision and Pattern Recognition: Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, USA, 18 June - 23 June, 2018. P. 7132-7141.
13. K. Cho et. al., Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014: Proceedings of the Empirical Methods in Natural Language Processing*, Doha, Qatar, 25 October – 29 October, 2014. P.123-126.
14. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate. *ICLR 2015: Proceedings of the International Conference on Learning Representations*, San Diego, California, USA, 7 May – 9 May, 2015. P.201-206.
15. O. Vinyals, L. Kaiser, T. Koo, S. Petrov, et. al. Grammar as a foreign language. *Neural Information Processing Systems 2015 – Volume 2: Advances in Neural Information Processing Systems*, 7 December – 12 December, 2015. P. 2773-2781.
16. Yuanhang Su, C.-C. Jay Kuo. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing*. 2019. Vol. 356, No. 1. P. 151-161.
17. Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, et. al. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *IJCAI-17: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 19 August – 25 August, 2017. P.2627-2633.
18. Kyunghyun Cho, et. al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. Montreal: Association for Computational Linguistics, 2014. 412p.
19. Kalchbrenner N., Blunsom P. Recurrent continuous translation models. Berlin: EMNLP, 2013. 422p.
20. Sutskever I., Vinyals O., Le. Q. Sequence to sequence learning with neural networks. San Jose: NIPS, 2014.388p.

21. Bahdanau D., Cho K., Bengio Y. Neural machine translation by jointly learning to align and translate. Bremen: arXiv, 2014. 195p.
22. Дуброва Т.А., Статистические методы прогнозирования. Ростов: ЮНИТИ, 2003. 206 с.
23. Айвазян С. А. Прикладная статистика. Основы эконометрики. Москва: Юнити-Дана, 2001. 432 с.
24. Магнус Я. Р., Катышев П. К., Пересецкий А.А. Эконометрика. Начальный курс. Москва: Дело, 2007. 504 с.
25. Елисеева И. И , Эконометрика. Москва: Финансы и статистика, 2006. 576с.
26. Росси Э. Одномерные GARCH-модели: обзор. Москва: Квантиль, 2010. 67с.
- 27.Зайченко Ю.П.. Нечеткие модели и методы в интеллектуальных системах. Киев: Дом «Слово», 2008, 354с.
- 28.Зайченко Ю.П. Основи проектування інтелектуальних систем. Київ: Видавничий Дім «Слово», 2004. 352с
29. Тарик Р. Создаем нейронную сеть. Санкт-Петербург: Альфа-книга, 2017. 272 с.
30. Суворова Є. Ю. Аналіз основних технологій обчислюваного інтелекту. Львів: Левко, 2013. 165 с.
31. Згуровський М. З., Зайченко Ю. П. Основы вычислительного интеллекта. Киев: Наукова думка, 2013. 406 с.
32. Николенко С. Глубокое обучение. Погружение в мир нейронных сетей. Санкт-Петербург: Питер, 2018. 480 с.
33. Blockchain.info. Total Circulating Bitcoin. URL: <https://www.blockchain.com/charts/total-bitcoins> (дата звернення: 21.09.2021)
34. Bitcoin.org “FAQ”, 2009. URL: <https://bitcoin.org/en/faq> (дата звернення: 21.09.2021)

35. De Vries A. Bitcoin's growing energy problem. URL: <https://www.sciencedirect.com/science/article/pii/S2542435118301776> (last accessed 26.09.2021)
36. Library of Congress. Our New Reports on Regulation of Cryptocurrency Around the World. URL: <https://blogs.loc.gov/law/2018/07/our-new-reports-on-regulation-of-cryptocurrency-around-the-world/> (last accessed: 21.09.2021)
37. Hayes A. Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0736585315301118> (last accessed: 09.11.2021)
38. Euny H. How Does Bitcoin Mining Work?. 30.11.2021. URL: <https://www.investopedia.com/tech/how-does-bitcoin-mining-work/> (last accessed: 10.12.2021)
39. Анна Акопян, Что такое Ethereum: объясняем простыми словами. 08.07.2021. URL: <https://currency.com/ru/chto-takoe-ethereum-prostymi-slovami> (дата звернения 10.12.2021)
40. Иван Гидаспов, Биткоин vs лайткоин: что выбрать. 27.11.2021. URL: <https://currency.com/ru/bitcoin-vs-litecoin-chto-vybrat> (дата звернения: 10.12.2021)
41. Иван Гидаспов, Георгий Истигечев, Криптовалюта Cardano (ADA): причины роста и прогноз на 2021. 30.11.2021. URL: <https://currency.com/ru/kriptovalyuta-cardano-ada-stoit-li-investirovat> (дата звернения: 10.12.2021)
42. Анна Акопян, Криптовалюта PolkaDot: что это, как работает, прогноз. 02.11.2021. URL: <https://currency.com/ru/obzor-kriptovalyuty-polkadot-ubijca-ehfira-ili-blokchejn-budushchego> (дата звернения: 10.12.2021)
43. Владислав Можаров, Что такое Solana?. Масштабируемая децентрализованная сеть для DApps. 24.05.2021. URL: <https://vc.ru/crypto/250113-chto-takoe-solana-masshtabiruemaya-decentralizovannaya-set-dlya-dapps> (дата звернения: 10.12.2021)

44. Зайченко Ю.П. Нечеткие модели и методы в интеллектуальных системах. Дом «Слово», Киев: 2008. 352с.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

!pip install git+git://github.com/kvoyager/GmdhPy.git
from gmdhpy.gmdh import Regressor
# 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, BatchNormalization, SimpleRNN,
RepeatVector, TimeDistributed
from keras.optimizers import SGD
import math
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error, r2_score
# Some functions to help out with
def plot_predictions(test,predicted):
    plt.plot(test, color='red',label='Real Coin Price')
    plt.plot(predicted, color='blue',label='Predicted Coin Price')
    plt.title('Coin Price Prediction')
    plt.xlabel('Time')
    plt.ylabel('Coin Price')
    plt.legend()
    plt.figure(figsize=(16,4))
    plt.show()

def return_rmse(test,predicted):
    rmse = math.sqrt(mean_squared_error(test, predicted))
    print("The root mean squared error is {}".format(rmse))
    print("The MSE is {}".format(mean_squared_error(test, predicted)))
    print("The MAE is {}".format(mean_absolute_error(test, predicted)))
    print("The R2_Score is {}".format(r2_score(test, predicted)))
    print("The MAPE is {}".format(mean(MAPE(test, predicted))))
# First, we get the data
dataset = pd.read_csv('../input/crypto-data1/coin_Litecoin.csv', index_col='Date', parse_dates=['Date'])
dataset
# Checking for missing values
training_set = dataset[:'2020'].iloc[:,6:7].values
test_set = dataset['2021:'].iloc[:,6:7].values

# print(training_set)
# print(test_set)
# We have chosen 'Close' attribute for prices. Let's see what it looks like
dataset["Close"].loc['2017':'2020'].plot(figsize=(16,4),legend=True)
dataset["Close"].loc['2021:'].plot(figsize=(16,4),legend=True)
plt.legend(['Training set (From 2017 till 2021)','Test set (2021 and beyond)'])

```

```

plt.title('LTC coin')
plt.show()
# Scaling the training set
sc = MinMaxScaler(feature_range=(0,1))
training_set_scaled = sc.fit_transform(training_set)
# print(training_set_scaled.shape)
print(training_set.shape)
print(test_set.shape)
window = 7
X_train = []
y_train = []
for i in range(window,training_set.shape[0]):
    X_train.append(training_set[i-window:i,0])
    y_train.append(training_set[i,0])
X_train, y_train = np.array(X_train), np.array(y_train)
print(X_train.shape)
print(y_train.shape)
gmdh = Regressor(ref_functions=('cubic','quadratic','linear','linear_cov'),
                  criterion_type='validate',
                  criterion_minimum_width=7,
                  stop_train_epsilon_condition=0.0001,
                  layer_err_criterion='top',
                  l2=0.5,
                  verbose=1,
                  n_jobs='max')
# Now to get the test set ready in a similar way as the training set.
# The following has been done so first 60 entries 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["Close"]['2017':'2020'],dataset["Close"]['2021:']),axis=0)
inputs = dataset_total[len(dataset_total)-len(test_set) - window:].values
# print(inputs.shape)
inputs = inputs.reshape(-1,1)
print(inputs.shape)

# print(inputs.shape)
# inputs = sc.transform(inputs)
# print(inputs.shape)
print(test_set.shape)
# Preparing X_test and predicting the prices
X_test = []
y_test = []
for i in range(window,inputs.shape[0]):
    X_test.append(inputs[i-window:i,0])
    y_test.append(inputs[i,0])
X_test, y_test = np.array(X_test), np.array(y_test)
print(X_test.shape)
print(y_test.shape)
# X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
gmdh.fit(X_train,y_train)

```

```

predicted_stock_price = gmdh.predict(X_test)
print(predicted_stock_price.shape)
# predicted_stock_price = regressor.predict(X_test)
# predicted_stock_price.reshape(-1,1)
# predicted_stock_price = sc.inverse_transform(predicted_stock_price)
plot_predictions(test_set,predicted_stock_price)
return_rmse(y_test,predicted_stock_price)
from sklearn.metrics import r2_score
y_test, predicted_stock_price = list(y_test),list(predicted_stock_price)
# gmdh.score(y_test, predicted_stock_price)
r2_score(y_test, predicted_stock_price)
# The LSTM architecture
regressor = Sequential()
# First LSTM layer with Dropout regularisation
regressor.add(LSTM(units=30, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressor.add(Dropout(0.1))
# Second LSTM layer
regressor.add(LSTM(units=30, return_sequences=True))
regressor.add(Dropout(0.1))
# Third LSTM layer
regressor.add(LSTM(units=30, return_sequences=True))
regressor.add(Dropout(0.1))
# Fourth LSTM layer
regressor.add(LSTM(units=30))
regressor.add(Dropout(0.1))
# The output layer
regressor.add(Dense(units=1))

# Compiling the RNN
regressor.compile(optimizer='rmsprop',loss='mean_squared_error')
# Fitting to the training set
csv_logger = CSVLogger('trainingLSTM.log')
regressor.fit(X_train,y_train,epochs=10,batch_size=8,callbacks=[csv_logger])
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=120, return_sequences=True, input_shape=(X_train.shape[1],1),
activation='tanh'))
regressorGRU.add(Dropout(0.2))
# Second GRU layer
# regressorGRU.add(GRU(units=120, return_sequences=True, input_shape=(X_train.shape[1],1),
activation='tanh'))

```

```

# regressorGRU.add(Dropout(0.2))
# Third GRU layer
# regressorGRU.add(GRU(units=120, return_sequences=True, input_shape=(X_train.shape[1],1),
activation='tanh'))
# regressorGRU.add(Dropout(0.2))
# Fourth GRU layer
regressorGRU.add(GRU(units=120, 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',metrics=['accuracy'])
# Fitting to the training set
csv_logger = CSVLogger('trainingGRU.log')
regressorGRU.fit(X_train,y_train,epochs=10,batch_size=16,callbacks=[csv_logger])
# Visualizing the results for GRU
plot_predictions(test_set,GRU_predicted_stock_price)

# Evaluating GRU
return_rmse(test_set,GRU_predicted_stock_price)
# The LSTM architecture
regressorRNN = Sequential()
# First LSTM layer with Dropout regularisation
regressorRNN.add(SimpleRNN(units=30, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressorRNN.add(Dropout(0.1))
# Second LSTM layer
regressorRNN.add(SimpleRNN(units=30, return_sequences=True))
regressorRNN.add(Dropout(0.1))
# Third LSTM layer
regressorRNN.add(SimpleRNN(units=30, return_sequences=True))
regressorRNN.add(Dropout(0.1))
# Fourth LSTM layer
regressorRNN.add(SimpleRNN(units=30))
regressorRNN.add(Dropout(0.1))
# The output layer
regressorRNN.add(Dense(units=1))

# Compiling the RNN
regressorRNN.compile(optimizer='rmsprop',loss='mean_squared_error', metrics=['accuracy'])
# Fitting to the training set
csv_logger = CSVLogger('trainingRNN.log')
regressorRNN.fit(X_train,y_train,epochs=10,batch_size=32,callbacks=[csv_logger])
predicted_stock_price = regressorRNN.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 LSTM architecture
Bi_regressor = Sequential()
# First LSTM layer with Dropout regularisation
Bi_regressor.add(Bidirectional(LSTM(units=150, return_sequences=True),
input_shape=(X_train.shape[1],1)))
Bi_regressor.add(Dropout(0.3))
# Second LSTM layer
Bi_regressor.add(Bidirectional(LSTM(units=150, return_sequences=True)))
Bi_regressor.add(Dropout(0.3))
# Third LSTM layer
# Bi_regressor.add(Bidirectional(LSTM(units=100, return_sequences=True)))
# Bi_regressor.add(Dropout(0.5))
# Fourth LSTM layer
Bi_regressor.add(LSTM(units=150))
Bi_regressor.add(Dropout(0.3))
# The output layer
Bi_regressor.add(Dense(units=1))

# Compiling the RNN
Bi_regressor.compile(optimizer='adam',loss='mean_squared_error', metrics=['accuracy'])
# Fitting to the training set

csv_logger = CSVLogger('trainingBiLSTM.log')
Bi_regressor.fit(X_train,y_train,epochs=10,batch_size=32,callbacks=[csv_logger])
predicted_stock_price = Bi_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 LSTM architecture
Bi_regressorGRU = Sequential()
# First LSTM layer with Dropout regularisation
Bi_regressorGRU.add(Bidirectional(GRU(units=100, return_sequences=True),
input_shape=(X_train.shape[1],1)))
Bi_regressorGRU.add(Dropout(0.4))
# Second LSTM layer
Bi_regressorGRU.add(Bidirectional(GRU(units=100, return_sequences=True)))
Bi_regressorGRU.add(Dropout(0.4))
# Third LSTM layer
Bi_regressorGRU.add(Bidirectional(GRU(units=100, return_sequences=True)))
Bi_regressorGRU.add(Dropout(0.4))
# Fourth LSTM layer
Bi_regressorGRU.add(GRU(units=100))
Bi_regressorGRU.add(Dropout(0.4))
# The output layer

```

```

Bi_regressorGRU.add(Dense(units=1))

# Compiling the RNN
Bi_regressorGRU.compile(optimizer='rmsprop',loss='mean_squared_error', metrics=['accuracy'])
# Fitting to the training set
csv_logger = CSVLogger('trainingBiGRU.log')
Bi_regressorGRU.fit(X_train,y_train,epochs=10,batch_size=16,callbacks=[csv_logger])
predicted_stock_price = Bi_regressorGRU.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)
from tensorflow.keras.layers import Layer
from tensorflow.keras import backend as K

class Attention(Layer):

    def __init__(self, return_sequences=True):
        self.return_sequences = return_sequences
        super(Attention,self).__init__()

    def build(self, input_shape):

        self.W=self.add_weight(name="att_weight", shape=(input_shape[-1],1),
                                initializer="normal")
        self.b=self.add_weight(name="att_bias", shape=(input_shape[1],1),
                                initializer="zeros")

        super(Attention,self).build(input_shape)

    def call(self, x):

        e = K.tanh(K.dot(x,self.W)+self.b)
        a = K.softmax(e, axis=1)
        output = x*a

        if self.return_sequences:
            return output

        return K.sum(output, axis=1)
# The LSTM architecture
Bi_regressorAtt = Sequential()
# First LSTM layer with Dropout regularisation
Bi_regressorAtt.add(Bidirectional(LSTM(units=50, return_sequences=True),
input_shape=(X_train.shape[1],1)))
Bi_regressor.add(Dropout(0.2))
# Second LSTM layer

```

```

# Bi_regressorAtt.add(Bidirectional(LSTM(units=100, return_sequences=True)))
# Bi_regressorAtt.add(Dropout(0.1))

Bi_regressorAtt.add(Attention(return_sequences=True))

# Third LSTM layer
# Bi_regressorAtt.add(Bidirectional(LSTM(units=32, return_sequences=True)))
# Bi_regressorAtt.add(Dropout(0.2))
# Fourth LSTM layer

Bi_regressorAtt.add(LSTM(units=50))
Bi_regressorAtt.add(Dropout(0.2))
# The output layer
Bi_regressorAtt.add(Dense(units=1))

# Compiling the RNN
Bi_regressorAtt.compile(optimizer='adam',loss='mean_squared_error', metrics=['accuracy'])
# Fitting to the training set
csv_logger = CSVLogger('trainingAtt.log')
Bi_regressorAtt.fit(X_train,y_train,epochs=10,batch_size=16,callbacks=[csv_logger])

predicted_stock_price = Bi_regressorAtt.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)

```