

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

МАЗУР ЮЛІА ОЛЕКСАНДРІВНА

Допускається до захисту:
В.о. завідувача кафедри
прикладної математики та
кібербезпеки, доктор філософії з
математики

_____ Луценко А. В.
«__» _____ 20__ р.

**РОЗРОБКА МЕТОДИКИ КОНТРОЛЮ ОБІГУ ЦИФРОВИХ
ФІНАНСОВИХ АКТИВІВ НА ОСНОВІ КРИПТОВАЛЮТНИХ
ТРАНЗАКЦІЙ**

Спеціальність 105 Прикладна фізика та наноматеріали

Кваліфікаційна (магістерська) робота

Науковий керівник:
Л. В. Загоруйко,
к.т.н., доцент,
доцент кафедри прикладної
математики та кібербезпеки

(підпис)

Оцінка : ____ / ____ / ____

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця 2024

АНОТАЦІЯ

Мазур Ю.О. Розробка методики контролю обігу цифрових фінансових активів на основі криптовалютних транзакцій. Спеціальність 105 «Прикладна фізика та наноматеріали», Освітня програма «Технології інтернету речей». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (магістерській) роботі досліджено методики контролю обігу цифрових фінансових активів, зокрема на прикладі здійснення криптовалютних транзакцій. Проаналізовано відомі методи аналізу та передбачення цін на криптовалюти. Розроблено методику контролю обігу цифрових фінансових активів на основі криптовалютних транзакцій на базі простих рекурентних нейронних мереж (SRNN).

Ключові слова: криптовалютні транзакції, штучний інтелект, нейронні мережі, криптовалюта, методи контролю обігу.

76 с., 3 табл., 30 рис., 8 дод., 43 джерел.

Mazur Yu. Development of a methodology for monitoring the circulation of digital financial assets based on the analysis of cryptocurrency transactions. Specialty 105 «Applied physics and nanomaterials», Educational program «Internet of Things technologies». Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

The qualification (master's) work investigates methods of controlling the circulation of digital financial assets, in particular, on the example of cryptocurrency transactions. The author analyzes the known methods of analyzing and predicting cryptocurrency prices. A methodology for controlling the circulation of digital financial assets based on cryptocurrency transactions based on simple recurrent neural networks (SRNN) is developed.

Keywords: cryptocurrency transactions, artificial intelligence, neural networks, cryptocurrency, methods of circulation control.

76 pp., 3 tables, 30 figures, 43 sources.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1 МЕТОДИКИ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ НА ОСНОВІ АНАЛІЗУ КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ.....	7
1.1 Криптовалютні транзакції та їх нормативно-правове регулювання	7
1.2 Сучасні моделі машинного навчання для аналізу криптовалютних транзакцій	13
1.2.1 Гібридна модель машинного навчання.....	15
1.2.2 Модель Jang та Lee.....	17
1.2.3 Модель Mittal та Poongodi.....	19
1.2.4 Модель Amon та Kim.....	20
1.2.5 Модель Chang	22
1.3 Висновок до розділу 1	26
РОЗДІЛ 2 НЕЙРОННІ МЕРЕЖІ ЯК МОДЕЛЬ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ	27
2.1 Класифікація нейронних мереж та їх архітектура	27
2.1.1 Нейронні мережі прямого поширення	27
2.1.2 Згорткові нейронні мережі	28
2.1.3 Рекурентні нейронні мережі	30
2.2 Порівняння нейронних мереж, за їх архітектурою.....	32
2.3 Рекурентні нейронні мережі та їх види.....	34
2.3.1 Long Short Term Memory (LSTM).....	35
2.3.2 Gates Recurrent Units (GRU).....	36
2.3.3 Simple Recurrent Neural Network (SRNN)	38
2.4 Порівняння рекурентних нейронних мереж.....	39
2.5 Висновки до розділу 2	40
РОЗДІЛ 3 РОЗРОБКА МЕТОДИКИ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ.....	42
3.1 Криптовалюта, як різновид фінансових активів	42
3.2 Розробка методики контролю за обігом цифрових активів.....	44
3.3 Висновок до розділу 3	58
ВИСНОВОК	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	60



ВСТУП

Історія розвитку грошей – це певна історія еволюції. Від металевих копійок до паперових банкнот, від online-платежів до використання віртуальних грошей. Вперше про використання віртуальних валют, а саме криптовалют заговорили у 2008 році. Тоді невідома особа чи група осіб під псевдонімом Satoshi Nakamoto створили криптовалюту. Основною ідеєю цифрової валюти було створення незалежної та децентралізованої системи електронних платежів, які засновані на математичних доказах і криптографії.

Криптовалюта стала ще одним етапом розвитку еволюції, який несе в собі як величезні можливості, так і ризики. Віртуальна грошова одиниця не є фіатною валютою тому мови про її знецінення йти не може. Фіатна валюта – це та, яку уряд оголосив законним платіжним засобом, як-от гривні чи долари. До прикладу, біткоїн відносно стійкий до цензури та політичних тисків саме через це. Біткоїн та загалом операції з криптовалютами за замовчуванням мають бути довірені та перевірені усіма учасниками індустрії. Криптовалюта відкрита для усіх користувачів, тобто можна сказати, що криптовалюта є наддержавною централізованою системою якою можуть керувати усі. Однак існує чимало ризиків, пов'язаних із даною валютою. Одним з них є ризик волатильності біткоїна. Волатильність – це тенденція зміни ринкових цін і доходів впродовж певного часу.

З точки зору безпеки, криптовалюта вимагає використовувати верифікацію. Це є обов'язковою процедурою, що проводиться під час реєстрації користувача для створення криптовалютного гаманця. Але варто також наголосити на тому, що біржа яка отримує персональні дані користувача не має права надавати їх стороннім людям, чи використовувати без потрібної на те причини. Доступ до даної інформації, згідно з чинним законодавством може отримувати тільки кіберполіція України.

Актуальність роботи. Ринок криптовалют на сьогоднішній день є одним з найбільш динамічних та складних ринків у світі. Ціни на ньому сильно коливаються, що в свою чергу ускладнює інвесторам та трейдерам можливість

приймати обґрунтовані рішення. Криптовалюти мають багато переваг над традиційними валютами, наприклад, анонімність та децентралізацію. Аналіз криптовалютних транзакцій є важливим інструментом протидії цим ризикам. Він дозволяє виявляти потенційну злочинну діяльність, навіть якщо вона непомітна для громадськості. За останні роки було досягнуто значного прогресу в розробці моделей машинного навчання для аналізу криптовалютних транзакцій. Ці моделі успішно використовуються для виявлення ризиків та різних видів злочинної діяльності. Їх вдосконалення та інтеграція триває, адже це надзвичайно важлива сфера в контексті безпеки та ефективності фінансових операцій.

Мета роботи – розробка методики контролю за обігом цифрових фінансових активів на основі криптовалютних транзакцій з врахуванням їх інтеграцій в екосистему інтернету речей задля забезпечення безпеки, ефективності та децентралізованого управління фінансовими капіталами.

Для досягнення поставленої мети потрібно виконати наступні **завдання**:

1. Аналіз відомих моделей машинного навчання для контролю за обігом цифрових фінансових активів та моделювання зміни цін на криптовалюту.
2. Порівняння архітектур, структур або моделей нейронних мереж та вибір за їх класифікацією та вибір оптимальної структури, архітектури або моделі.
3. Розробка методики контролю за обігом цифрових фінансових активів на основі криптовалютних транзакцій з використанням нейронних мереж.

РОЗДІЛ 1

МЕТОДИКИ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ НА ОСНОВІ АНАЛІЗУ КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ

1.1 Криптовалютні транзакції та їх нормативно-правове регулювання

Історія проведення першої транзакції з-за допомогою віртуальних грошей розпочалась ще у травні 2009 році. Саме тоді Лазло Ганеш зробив першу комерційну транзакцію за біткоїн, купивши дві піци за 10 000 BTC. Bitcoin – став першою віртуальною валютою, на основі якої почали започатковувати інші криптовалюти. У 2008 році Satoshi Nakamoto [1] використав наступні припущення, які у подальшому призвели до виникнення цифрових грошей. Повністю однорангові електронні гроші дозволять здійснювати Online-платежі безпосередньо між двома сторонами без залучення фінансових установ. Цифрові підписи є частиною рішення, але основна перевага втрачається, якщо довірений посередник все ще необхідний для запобігання повторних витратів. Мережа фіксує процес, об'єднуючи його в безперервний рядок хешів, отриманих під час proof-of-work (PoW), і створює записи, які неможливо змінити без повторного запуску тесту. Proof-of work це алгоритм, який використовують для організації нових блоків у блокчейн криптовалюти. Найдовший ланцюжок служить не тільки доказом спостережуваної послідовності подій, але й доказом того, що він походить від найбільшого джерела обчислювальної потужності. Більша частина обчислювальної потужності контролюється вузлами, які не співпрацюють у атаці на мережу, але утворюють найдовші ланцюжки та завжди на крок попереду зловмисника. Сама мережа вимагає мінімальної структури. Повідомлення розповсюджується за принципом «максимальних зусиль», вузли можуть вільно залишати мережу та знову приєднуватися до мережі як доказ того, що сталося за їх відсутності. Хоча система працює досить добре для більшості транзакцій, вона все ще страждає від неминучих недоліків моделі, заснованої на довірі. Повністю незворотні транзакції практично неможливі, оскільки фінансові установи не можуть уникнути посередництва в суперечках. Посередницькі витрати збільшують транзакційні витрати, обмежуючи практичний мінімальний розмір

транзакцій та усуваючи можливість невеликих, випадкових транзакцій. Існують також більші витрати, пов'язані з несплатою неповернених платежів за послуги, що не підлягають відшкодуванню. З можливістю відшкодування поширюється потреба в довірі. Торговці повинні бути обережними, коли клієнти просять більше інформації, ніж їм потрібно. Певний рівень шахрайства вважається неминучим. Цих платіжних витрат і невизначеності можна уникнути безпосередньо за допомогою фізичної валюти, але немає механізму здійснення платежів через канал зв'язку без довіреної сторони. Потрібна електронна платіжна система, заснована на криптографічному доказі, а не на довірі, що дозволить двом бажаючим сторонам здійснювати транзакції безпосередньо одна з одною, не потребуючи довіреної третьої сторони [1]. Незворотні цифрові транзакції захищають продавців від шахрайства, а для захисту покупців можна легко застосовувати звичайні механізми рахунків умовного зберігання (ескроу).

Другою за капіталізацією криптовалюта, яка зробила можливими випуск сотень бо навіть тисяч інших криптовалют є Ethereum. Ethereum – керована технологія спільноти, що стоїть за криптовалютою Ether (ETH) і тисячами децентралізованих програм [2]. Дану криптовалюту було розроблено у листопаді 2013 року для створення більш універсальної платформи. Вона використовує механізм PoW, щоб уніфікувати концепцію суспільного економічного консенсусу [3]. Завдяки можливостям потужності віртуальних машин розробники додатків можуть легко створювати програми, які отримують переваги від децентралізованих і безпечних блокчейнів, без необхідності створювати новий блокчейн для кожної нової програми [2]. Ethereum – це проект, спрямований на створення загальної технології, на основі якої можна побудувати всі концепції транзакційних автоматів стану, а також надати розробникам «тісно інтегровані наскрізні системи для створення програмного забезпечення на раніше не досліджених основних обчислювальних моделях; надійна обчислювальна платформа обміну повідомлення від об'єкту до об'єкта» [4]. Подібно до Bitcoin, Ethereum також має цифрову валюту та є децентралізованим, без управління центральним банком [5]. На відміну від попередньої

криптовалюти, Ethereum був створений не тільки для обміну, але і для програмування. Розробники цієї криптовалюти створили притчу, аби показати головну різницю між цими двома цифровими валютами: «Хоча Bitcoin це просто платіжна мережа, Ethereum – це ринок фінансових послуг, ігор, соціальних мереж та інших програм, які поважають вашу конфіденційність, і не можуть вас цензурувати» [2]. Як вже зазначалось раніше Ethereum є програмованим, тому він дозволяє користувачам визначати власні правила, формати транзакцій і функції стану за допомогою свого загального класу. Це досягається за допомогою смарт контрактів [3]. Смарт-контракти це комп'ютерні програми, що знаходяться в блокчейні Ethereum. Коли транзакція походить від іншого користувача, смарт-контракт активується та виконується. Саме ця характеристика відрізняє дану криптовалюту від інших, оскільки робить його більш гнучким. Загалом технологія блокчейну працює на основі окремого вузла, і кожен вузол Ethereum запускає копію віртуальної машини EVM, яка виконує смарт-контракт та гарантує, що всі вузли досягають однакових результатів [3].

Останнім часом блокчейн привернув увагу представників різних сфер: від фінансів і охорони здоров'я до комунальних та державних послуг. Причиною такого інтересу полягає в тому, що блокчейн може створювати децентралізовані програми, які раніше могли працювати лише через ненадійного посередника. Тепер ж вони можуть працювати без центрального органу та забезпечувати той самий рівень функціональності та надійності. Можна сказати, що блокчейн створює ненадійні мережі, оскільки сторони можуть здійснювати транзакції, навіть якщо вони не довіряють одна одній. Відсутність надійного посередника означає, що розрахунок між сторонами угоди відбувається швидше. Блокчейн – це розподілена структура даних, яка копіюється та ділиться учасниками мережі [5]. Вперше він був представлений разом із біткоїнами для вирішення проблеми подвійних витрат. Оскільки вузли в мережі біткоїн додають до неї перевірені та узгоджені транзакції, блокчейн Bitcoin є авторитетною книгою транзакцій, яка встановлює, хто чим володіє [6]. Кожен блок ідентифікується за допомогою криптографічної хеш-функції. Кожен блок має містити посилання на хеш

функцію попереднього блоку в їхньому ланцюжку. Така схема створює зв'язок між блоками, утворюючи ланцюжок блоків або блокчейн. Будь-який вузол, який має доступ до цього впорядкованого спуску блоків відповідей, може прочитати його та зрозуміти стан даних, якими обмінюються в мережі. Мережею блокчейнів називають набір вузлів, які працюють з одним блокчейном, використовуючи копії своїх даних. Ці вузли утворюють однорангову мережу, де користувачі взаємодіють із блокчейнами за допомогою приватного та публічного ключів. До прикладу приватні ключі використовуються для підписання власних транзакцій, а IP-адреси користувачі ідентифікуються за допомогою публічного ключа. Використання асиметричної криптографії забезпечує автентичність, цілісність і неспростовність транзакцій. Кожна підписана транзакція передається вузлом користувача своїм однопрохідним сусідам. Сусідні вузли, в свою чергу, перевіряють дійсність вхідної транзакції перед її передачею, недійсні транзакції будуть відхилені. Транзакції, зібрані та перевірені мережею протягом узгодженого періоду часу, сортуються та групуються в блок-кандидат із міткою часу. Цей процес називається майнінгом. Майнінговий вузол передає даний блок у мережу. Вузол перевіряє чи запропонований блок містить транзакцію та чи хеш попереднього блоку в їх ланцюжкові має правильне посилання. Якщо ці дві характеристики наявні, то вони додають блок до свого ланцюжка та застосовують транзакції в нього, аби оновити стан своїх даних. В іншому ж випадку запропонований блок буде відхилено. Як зазначалось раніше, блокчейн це сукупність ненадійних учасників, які спільно використовують баку даних без довіреного посередника, та для того, аби уникати хаосу в розподіленому середовищі та допомогти мережі досягти спільного консенсусу, кожна мережа повинна встановлювати певні правила, яким повинні відповідати всі транзакції. Дані правила запрограмовані в кожному блокчейн-клієнті, який потім використовує їх, щоб вирішити, чи є вхідна транзакція дійсною.

Майнінг є невід'ємною частиною операцій з криптовалютою, що забезпечує створення, передачу та перевірку транзакцій [7]. Це забезпечує стабільний, безпечний та надійний рух валюти від платника до одержувача. На

відміну від фіатних валют, де централізований орган контролює та регулює транзакції, криптовалюти децентралізовані та працюють на одноранговій основі. Для роботи банків, які створюють фізичну валюту та контролюють транзакції, потрібна величезна інфраструктура. Криптовалюти долають цю проблему за допомогою системи майнінгу, у якій учасники мережі, які називаються «майнерами» або «вузлами», відстежуються і підтверджують транзакції, за які вони отримують винагороду у формі створення нової валюти. Майнери відіграють важливу роль у майнінгу. Вони обробляють транзакції, перевіряючи право власності на валюту від джерела до місця призначення. Кожна транзакція містить хеш попередньої транзакції, здійсненої власником, який використовується для перевірки автентичності поточної транзакції, тим самим підтверджують її. Майнери також уникають подвійних витрат завдяки цьому процесу. Основною метою майнінгу є створення та випуск монет у криптовалютну економіку. Щоразу, коли транзакція відбувається та підтверджується, майнери збирають ці транзакції включають їх у блок, який вони вирішують. Кожен блок має бути розкрито, перш ніж його можна буде звільнити та додати до блокчейну. Розгадування блоків включає розв'язання складних і нерозв'язних математичних головоломок із обмеженнями на отримані результати. Лише розв'язавши математичну головоломку, можна додати новий блок до книги, щоб отримати винагороду у вигляді монет. Тож врешті решт майнінг перетворюється на змагання з вирішення математичних головоломок. Такий механізм не дозволяє майнерам легко здобути монети, тим самим зберігаючи справедливість системи.

З моменту створення цифрових валют, криптовалюти не мали певного державного органу чи емітенту (юридичної особи, що випускає певну валюту), який контролював би їх обіг. Загалом існують централізовані децентралізовані біржі, які мають певні правила користування криптовалютами. Централізованою біржою називають біржу, яка має голову команди, який контролює та відповідає за цілісність активів. Децентралізовані біржі – це біржі, які зазвичай побудовані на смарт-контрактах, тобто всі дії є автоматичними. Як вже зазначалось раніше

криптовалюти можуть використовувати для фінансування тероризму або відмивання грошей, саме тому було створено орган ФАТФ.

ФАТФ це Міжнародна група з протидії відмивання брудних грошей. Загалом дана група створює рекомендації та методології, щодо запобігання витрачання грошей на кримінал [8]. Одним з таких документів є: «Рекомендації FATF Міжнародні стандарти боротьби з відмиванням коштів, фінансуванням тероризму і розповсюдженням зброї масового знищення; Методологія з оцінки відповідності рекомендаціям FATF та ефективності систем протидії відмиванню коштів та боротьби з фінансуванням тероризму; Правила та процедури 5-го раунду взаємних оцінок комітетом Moneyval» [9]. Ці стандарти та рекомендації спрямовані на боротьбу з відмиванням грошей та фінансуванням тероризму, а також запобігання цим злочинам. FATF опублікували спеціальні рекомендації щодо боротьби з відмиванням грошей і фінансуванням тероризму в криптовалютному секторі. Ці рекомендації пропонують наступні дії:

1. Виявлення та моніторинг транзакцій з криптовалютою
2. Ідентифікація та верифікація користувачів, що використовують криптовалюту
3. Повідомляти про підозрілі операції з криптовалютою
4. Фінансові установи повинні повідомляти відповідні правоохоронні органи про підозрілі операції з криптовалютою

У вересні 2020 року до Верховної ради України був внесений доопрацьований проект закони про віртуальні активи, де криптовалюта описується як нематеріальне благо. У цьому законопроекті майнінг також буде дещо легалізований на території України. Щоправда у 2020 році даний закон не був прийнятий і тільки у лютому 2022 року, президент України, Володимир Зеленський підписав закон. Закон України «Про віртуальні активи» регулює правовідносини, пов'язані з обігом віртуальних активів в Україні [10]. Він передбачає, що віртуальними активами є нематеріальні блага, що є об'єктами цивільних прав, мають вартість та можуть бути використані в якості засобів платежу, заощаджень або інвестицій. Учасниками ринку віртуальних активів є

власники віртуальних активів, біржі, постачальники послуг збереження віртуальних активів. Закон зокрема передбачає, що власники віртуальної власності мають право вільно розпоряджатися своєю віртуальною власністю та вимагає від бірж віртуальних активів виконувати свої зобов'язання. Також передбачено, що віртуальні біржі мають бути зареєстровані в Національному банку України. Провайдери послуг зі збереження віртуальних активів мають мати ліцензію Національного банку України. Законом передбачено, що національне регулювання ринку крипто активів здійснює Національний банк України.

На території України, також існують закон України «Про запобігання та протидію легалізації (відмиванню) доходів, одержаних злочинним шляхом, фінансуванню тероризму та фінансуванню розповсюдження зброї масового знищення» та закон України «Про валюту і валютні операції».

1.2 Сучасні моделі машинного навчання для аналізу криптовалютних транзакцій

Сучасні моделі машинного навчання дозволяють аналізувати криптовалютні транзакції більш ефективно та точно, ніж традиційні методи. Вони можуть використовуватися для виявлення фінансових злочинів, боротьби з незаконним оборотом криптовалют та регулювання криптовалютної галузі. Загалом існує чимало моделей машинного навчання, що використовують у сфері криптовалютних транзакцій, але найчастіше використовують наступні [11]:

1. Гібридна модель машинного навчання в сфері криптовалютних транзакцій.

Гібридні моделі формуються шляхом поєднання різних методів машинного навчання (МН) [12]. Проте, ця модель ефективно прогнозує щоденні ціни та не демонструє таку ж ефективність на більш тривалих періодах, наприклад при прогнозування на тиждень.

2. Модель Jang та Lee.

Jang та *Lee* спрогнозували ціну біткоїна за допомогою байєсівської мережі (БМ). Перевагою даної моделі є те, що вона враховує інформацію про блокчейни

та мікроекономічні фактори для прогнозування цін [13]. Вхідні дані для моделі включили десять глобальних макроекономічних індексів та п'ять міжнародних валютних курсів. Результати дослідження показали, що БМ демонструє найкращу прогностичну ефективність порівняно з базовими моделями, такими як лінійна регресія та регресія опорних векторів.

3. Модель Mittal.

Mittal прогнозує ціни десяти криптовалют за допомогою багатовимірної лінійної регресії [14]. Щоб прогнозувати найвищу ціну в певний день, беруть до уваги ціну відкриття, ціну закриття та найнижчі ціни попередніх днів.

4. Модель Poongodi.

Poongodi для прогнозування ціни використовував дві моделі: лінійну регресію та машину опорних векторів [15]. Результати показали, що машина опорних векторів без додаткових функцій має більшу точність прогнозування, ніж лінійна регресія, приблизно на 10%.

5. Вплив емоцій інвесторів на ціни криптовалют.

Дослідники виявили, що настрої в засобах масової інформації, таких як тренди *Google* та обсяги твітів, корелюють з цінами криптовалют. Однак настрої твітів не є надійним джерелом прогнозування, оскільки вони часто залишаються позитивними, навіть коли ціни криптовалют падають.

6. Модель Amon.

Amon використовував щоденні заголовки новин і твіти для прогнозування цінових коливань криптовалют, застосовуючи чотири алгоритми класифікації [16]. Запропонована модель добре визначає загальні тенденції на ринку криптовалют, але вона не може передбачити коливання цін, які не відповідають цим тенденціям.

7. Модель Kim.

Kim вивчав, як кількість транзакцій та інформація про онлайн-спільноти криптовалют можуть бути використані для прогнозування цін. Він використовував метод усередненої однозалежності для аналізу даних про

транзакції та коментарі та відповіді користувачів [17]. Результати дослідження показали, що результати прогнозування відрізняються для різних криптовалют.

8. Модель Chang.

Chang пропонує новий підхід до деталізації даних інженерії признаков [18]. Вони виділяють низькочастотні та високочастотні признаи та використовують різні методи для їх обробки. Результати дослідження показують, що статистичні методи краще працюють з низькочастотними даними, тоді як інші підходи машинного навчання, такі як лінійний дискримінантний аналіз і машини опорних векторів, досягають кращих результатів для високочастотних даних.

Всі ці моделі були розроблені з ціллю використання різних методів, але всі вони мають за мету підвищення ефективності та точності аналізу щодо прогнозування та безпеки криптовалютних транзакцій.

1.2.1 Гібридна модель машинного навчання

Процедура побудови ансамблевих моделей. Оскільки ціни на криптовалюту демонструють нелінійні властивості, такі як аперіодичність та стохастичність, ціни слід прогнозувати за допомогою нелінійних моделей [19]. Порівняно з традиційними економетричними моделями, моделі глибокого навчання на основі даних більш ефективно справляються з нелінійною природою цін на криптовалюту. Тому використовують моделі глибинного навчання як еталонні моделі для прогнозування цін. При цьому використовується багатомасштабний аналіз при побудові моделі. Таким чином, гібридна модель виконує дві функції. По-перше, модель може розкласти початковий ряд цін на підкомпоненти з різною частотою, а по-друге, зіставлення підкомпонентів і різні моделі глибокого навчання можуть покращити продуктивність моделі. Першим кроком є розкладання початкової ціни криптовалюту на різні підкомпоненти. Другим кроком є об'єднання підкомпонентів на основі їх подібності та частоти за допомогою методу вибіркової ентропії. Таким чином, високочастотні, середньочастотні та низькочастотні компоненти відповідно реконструюються.

Декомпозиція та реконструкція ціни криптовалюти. Багатомасштабний аналіз ціни біткоіна вимагає застосування методів декомпозиції сигналів. Найпоширеніші методи декомпозиції сигналу включають Wavelet-аналіз, EMD, Ensemble EMD (EEMD) та VMD. Islam використовував метод EMD для декомпозиції фінансових часових рядів та виявили, що продуктивність декомпозиції EMD була кращою, ніж Wavelet декомпозиція [20]. Huang вважав, що VMD може краще подолати проблеми ефектів кінцевих точок та модального змішування, а експериментальні результати показують, що VMD може підвищити точність прогнозування порівняно з методом EMD [21]. Порівняно з іншими методами декомпозиції, метод VMD може подолати проблеми, які зазвичай виникають під час декомпозиції, такі як модальне змішування, погане контурування та нестабільність меж, тому він все ще має хорошу стабільність при роботі з нелінійними і нестабільними даними. Незважаючи на те, що метод VMD має кілька переваг, його придатність для прогнозування ціни біткоінів не була систематично перевірена. Зважаючи на це, необхідно порівняти декомпозицію та ефективність прогнозування EMD та VMD у прогнозуванні ціни біткоіна.

Побудова моделі прогнозування на основі алгоритму LSTM. Алгоритм LSTM використовує структуру воріт для обробки даних, ефективно використовуючи поточну та історичну інформацію в даних часових рядів [19]. Модель LSTM має унікальну мережеву структуру, тому цей метод може краще вивчати минулі дані часових рядів, досліджувати взаємозв'язок між часовими рядами та глибше досліджувати властиві правила часових рядів, використовуючи функцію пам'яті вибірки, таким чином отримавши хороший ефект прогнозування.

Побудова моделі прогнозування на основі алгоритму ELM. ELM – це алгоритм навчання, що складається з одношарової нейронної мережі, яка складається з вхідного, вихідного та прихованого прошарку [19]. Відповідні вихідні ваги обчислюються шляхом випадкового генерування вхідних ваг і набору прихованих прошарків.

Показники оцінки ансамблевої моделі прогнозування. Для вимірювання здатності прогнозування інтегрованої моделі використовується R^2 , середня абсолютна похибка (MAE) та середньоквадратична похибка (RMSE) [19]. Для оцінки точності прогнозування спочатку використовується R^2 .

$$R^2 = \frac{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^N (\hat{y}_i - \bar{y}_i)^2}, \quad (1.1)$$

де R^2 відображає ступінь, до якого незалежна змінна може пояснити залежну змінну, і чим вище значення, тим кращий прогнозний ефект мають ансамблеві моделі.

MAE – це середнє значення відхилення між ціною біткоіна та прогнозованою ціною.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (1.2)$$

RMSE – це арифметичний квадратний корінь, з квадратного очікуваного значення різниці між розрахунковим значенням параметра і фактичним значенням. Середньоквадратичне відхилення використовується для оцінки ступеня зміни даних. Чим менше значення RMSE, тим вища точність прогнозуючої моделі в описі експериментальних даних.

$$RMSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (1.3)$$

Чим менші MAE та RMSE, тим менша похибка моделі прогнозування та краще ефективність прогнозування.

1.2.2 Модель Jang та Lee

Байєсівська нейронна мережа (BNN) – це трансформований багатошарковий перцептрон, який є загальним терміном для штучних нейронних мереж у галузі машинного навчання [22]. Для представлення складних часових рядів набагато ефективніше використовувати класичні лінійні моделі, такі як авторегресія. Структура BNN побудована з використанням ряду блоків обробки, класифікованих за трьома типами: вхідних та вихідних шарів, та одного або декількох прихованих шарів. До прикладу, нейронні мережі, що містять більше одного прихованого шару, можуть

вирішити проблему виняткового АБО (XOR), яка не може бути розв'язана одношаровим перцептроном [23]. На відміну від одношарових перцептронів, які можна розділити лише лінійно, вони вирішують проблеми XOR, вводячи алгоритми зворотного поширення та приховані прошарки. Прихований прошарок відображає вихідні дані в новий простір, який перетворює лінійно нероздільні дані в лінійно роздільні дані. Ваги штучних нейронних мереж були вивчені між вхідним прихованим прошарком і прихованим вихідним прошарком. Поширенням називається процес, в якому ваги прихованих прошарків коригуються помилкою прихованих прошарків, що поширюється помилкою вихідного прошарку. Метод оптимізації різниці між цільовим значенням і вихідним значенням при виведенні алгоритму зворотного поширення. Загалом, штучні нейронні мережі мінімізують суму наступних помилок E_B , використовуючи алгоритм зворотного розповсюдження та правило дельта [12].

$$E_B = \frac{\alpha}{2} \sum_{n=1}^N \sum_{k=1}^K (t_{nk} - o_{nk})^2 + \frac{\beta_T}{2} B^T B, \quad (1.4)$$

де E_B – сума помилок,

N – кількість початкових змінних,

K – розмір вихідного прошарку,

t_{nk} – k – та змінна n – го цільового вектора,

o_{nk} – k – та вихідна змінна n – го початкового вектору,

α та β – гіперпараметр,

B – вектор ваг байєсівської нейронної мережі.

BNN – це нелінійна версія гребеневої регресії, що базується головним чином на теорії нейронних мереж Байєса. На відміну від звичайних нейронних мереж, які спрямовані на максимізацію граничної ймовірності, BNN – це машина, яка максимізує значення апостеріорної ймовірності шляхом застосування теорії Байєса [13]. Елементи, додані до терму помилки, змушують машину навчатися, обираючи вагу з високим значенням навіть тоді, коли кількість загальних ваг зменшується, а не розподіляється на велику кількість ваг.

1.2.3 Модель Mittal та Poongodi

Багатовимірна лінійна регресія використовується для прогнозування найвищих та найнижчих цін на криптовалюту [14]. У наведеній моделі декілька незалежних змінних впливають на залежну характеристику за допомогою кількох коефіцієнтів.

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2, \dots, \theta_n x_n \quad (1.5)$$

де $h(\theta)$ – це вихідна оцінка на основі n незалежних ознак і відповідних параметрів θ .

Функцію витрат можна обчислити наступним чином:

$$J(\theta_1, \theta_2, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2 \quad (1.6)$$

Враховуючи, що в навчальних даних є m точок, $h_{\theta}(x_i)$ є прогнозованим результатом, і є спостережуваними даними залежної змінної. Таким чином, модель лінійної регресії, яка включає ряд характеристик, стане багатовимірною моделлю лінійної регресії. Розрахунок параметрів є найважливішим фактором у цій моделі. Для обчислення параметрів усі незалежні змінні не використовуються одночасно, щоб мінімізувати функцію помилки. Необхідно використовувати найкращі можливі незалежні змінні, і це можна зробити шляхом розрахунку кореляції між незалежними та залежними змінними. Це дозволить нам вирішити, які змінні є значущими, а які ні. З огляду на n функцій і m навчальних прикладів, x має m рядків і $n+1$ стовпців, де перший доданок є 0-м доданком, доданим до кожного вектору зі значенням 1 (це коефіцієнт постійного числа θ_0). У даному наборі даних використовували наступні незалежні функції для визначення максимальної ціни криптовалюти. Це такі ознаки, як [13]:

1. High (початкова ціна криптовалюти);
2. Low (найнижча досягнута ціна);
3. Close (ціна закриття криптовалюти).

За допомогою даних формул знайдено кореляцію між найвищим параметром High ціни та цими характеристиками. На основі цього створюється багатовимірна лінійна регресійна модель, що передбачає максимальну ціну.

Регресійна модель. Лінійною регресією – є модель прогнозування, яка будує найкращу пряму через скалярні залежні та незалежні змінні. Логістичною регресією – є модель прогнозовної регресії, у якій змінні є категоричними, тому вона мінімізує середню квадратичну похибку між прогнозованим і фактичним результатами [24]. Для формування ймовірностей того, що логістична регресія прийме певний клас, використовується оцінка максимальної правдоподібності за допомогою ітеративного алгоритму.

Машина опорних векторів (SVM). Машиною опорних векторів – є контрольований алгоритм машинного навчання, який використовується як для задач класифікації, так і для регресії [25]. Лінійні межі рішення доповнюються більш складною формою границі з використанням реалізації ядра. SVM досягає межі рішення, утворюючи поле, яке максимізує функціональні та геометричні межі між класами. Вона також може бути доповнена для регресійних задач як регресія з опорними векторами, що оптимізує відстань змінної відгуку від границі рішення. Жоден із цих методів не використовує дані, які безпосередньо пов'язані з рухом цін, і не аналізує середній відсоток зростання та зниження ціни, передбачений їхніми моделями.

1.2.4 Модель Amon та Kim

Вчені використовували настроїв твітів для прогнозування руху цін на біткойн, тоді було повідомлено про 90 % точність прогнозування цінових коливань за допомогою подібних контрольованих алгоритмів навчання, але їхні дані були позначені за допомогою вбудованого текстового API настрою [16]. Таким чином, їх оцінка точності базується на тому, наскільки добре модель відповідає Sentiment Text API, а не тому наскільки вона точна з точки зору прогнозування руху цін. Подібним чином Stenqvist та Lonno використовували алгоритм глибокого навчання протягом набагато більш тривалого часу – кожні 30 хвилин, щоб досягти 79% точності в прогнозуванні руху ціни біткойна на основі 2,27 мільйона твітів [26].

Щоденні дані про курс монет використовуються для призначення кожному заголовку новин шести значень та кожному твіту двох значень. Для заголовка новин, опублікованого в день d_i , три мітки позначають зміну кожної монети в день d_i+1 і три мітки позначають зміну в день d_i+2 . Для твіту, пов'язаного з біткоїном, опублікованого в день d_i , одна мітка означає зміну ціни біткоїна в день d_i+1 і одна мітка означає зміну в день d_i+2 . Аналогічна логіка впливає для твітів, пов'язаних з криптовалютою Ethereum. Мітки є двійковими $\in \{0,1\}$, де «0», означає зниження ціни, а «1» - підвищення. Після маркування заголовки новин і твіти були сегментовані на тренувальні (60%), розробницькі (20%) та тестові (20%) набори [16].

Вилучення ознак. Кожен заголовок та твіт було токенизовано функціями з бібліотеки spaCy та додатковою логікою для перетворення всього тексту в малі літери, видалення пробілів, розділових знаків, символів зупинки та декількох специфічних типів рядків [27]. Ознаки, відібрані з заголовків та твітів, включають всі слова 1-ї та 2-ї градації, які з'являються в тексті.

Класифікація. Модель використовує класифікатор для визначення для визначення вагових коефіцієнтів, які використовуються для прогнозування міток даних. Початкова модель у дослідженні була побудована за допомогою простого класифікатора логістичної регресії. Даний метод був обраний, тому що він простий для розуміння та дозволяє швидко проводити аналіз помилок, щоб визначити як діяти далі [16]. Також у даному дослідженні спробували лінійну класифікацію опорних векторів, багаточленний Байєс та Бернуї, але логістична регресія дала найкращі результати на тестовій вибірці.

Предиктор. Щоб зробити унікальним прогноз щодо руху ціни кожної монети в певний день, результати класифікатора спочатку агрегуються за днями (наприклад, усі заголовки новин станом на 1 вересня 2023 року розглядаються разом) для кожного розділу. Потім остаточний прогноз буде призначено на основі мітки більшості, пов'язаної з кожною монеткою кожного дня [28]. Прогнози робляться окремо на основі тільки заголовків новин і тільки твітів.

1.2.5 Модель Chang

Метод неглибинного машинного навчання, які зазвичай використовуються в поточній моделі прогнозування запасів – це нейронні мережі, машини опорних векторів або моделі: які поєднують їх з іншими алгоритмами [18]. Hsu продемонстрував, що методи машинного навчання можуть бути використані для прогнозування фінансових ринків точніше, ніж економічні методи [29]. Це також демонструє, що на результати прогнозування фінансових активів впливають зрілість ринку, вхідні змінні, базовий період прогнозу та метод прогнозування.

Стоковий вектор. При обробці природної мови з'являється велика кількість нерозмічених текстів. Коли комп'ютер обробляє текст, він має бути перетворений у формат, зрозумілий машині. Тому дослідники запропонували використовувати термін «вектор слів», який використовує серію чисел для представлення слова. Найпростішим методом є єдиний вектор, що представляє позицію, де слово відображається як 1, а решта представлена як 0 [18]. Щоправда даний метод має два головних недоліки:

1. Розмірність вектору слів дорівнює розміру словника, але розмір словника занадто великий, тому розмірність вектора велика, що робить обчислення непрактичним.
2. Представлення не відображає подібність між словами, тому воно не підходить для обробки тексту та контекстно-семантичного аналізу.

Для того, аби повністю отримати інформацію з корпусу, дослідники впровадили нові методи машинного навчання, які в основному включили RBM, нейронну мережу та кореляцію між словом та контекстом [30]. Для конкретної мовної моделі, заданої послідовністю слів $(\omega_1, \omega_2, \dots, \omega_T)$, послідовністю прихованого прошарку $(h_1, h_2, \dots, h_T)$, та послідовність вихідного прошарку $(y_1, y_2, \dots, y_T)$, у RNN описується рівнянням

$$h_t = \tanh(W\omega_t + Uh_{t-1} + b_h) \quad (1.7)$$

$$y_t = Ah_t + b_y \quad (1.8)$$

,де ω_t -це вектор однієї гарячої точки слова t ,

$\theta = \{b_n, b_y, W, U, A\}$ це набір параметрів навчання.

Вкладеним вектором слова є матриця W .

Однак, штучні нейронні мережі в процесі навчання схильні до виникнення та зникнення градієнтів, було запропоновано розширену модель штучних нейронних мереж, в якій до початку структури додано прошарок ознак до початку структури.

$$h_t = \tanh(W\omega_t + Uh_{t-1} + F\mathcal{f}(t) + b_h). \quad (1.9)$$

При додаванні прошарку ознак $\mathcal{f}(t)$ інформація про втрати не згасає до 0, таким чином можна уникнути проблеми зникаючого градієнту. Раніше при прогнозуванні фондового ринку вхідними даними часто були одна акція або один індекс. Таким чином, розмірність вхідних даних не буде занадто великою. Однак, коли вхідними даними є історичні дані про акції, розмірність вхідних даних зростає до тисячі або навіть мільйонів [18]. Якщо використовувати дану вихідну інформацію для безпосереднього прогнозування фондового ринку, це може призвести до більших помилок через вплив інформаційної надлишковості та нерелевантної інформації. Тому вводиться концепція векторизації для фондового ринку, що названа фондовим вектором. Біржовий вектор відсилає до ідеї вектора слова. Перш за все, розмірність біржового вектора зменшується, а потім він виражається у просторі низької розмірності. Фондовий ринок прогнозується за допомогою фондового вектора. На основі вектора слів, запропоновано два типи моделей зменшення розмірності та прогнозування ціни акцій: нейронна мережа з довгою короткостроковою пам'яттю та вбудованим прошарком (ELSTM) та нейронна мережа з довгою короткочасною пам'яттю на основі автоматичного кодера (AELSTM).

LSTM алгоритм. У наведеній часовій моделі основним модулем мережі є LSTM [18]. Першим кроком є визначення того, яку інформацію збираються викинути зі стану комірки.

$$f_1 = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1.10)$$

Наступним кроком є задача, яку нову інформацію потрібно зберігати в комірці стану.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \quad (1.11)$$

$$C_t = \tanh(W_{\tilde{c}} \cdot [h_{t-1}, x_t] + b_c) \quad (1.12)$$

Залишилось оновити старий стан комірки на новий.

$$C_e = f_t * C_{t-1} + i_t + \tilde{C}_t. \quad (1.13)$$

Кінцевий результат буде базуватись на стані комірки, але вже відфільтрованої.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_j) \quad (1.14)$$

$$h_t = o_t * \tanh(C_t). \quad (1.15)$$

Час відкриття кожної акції на ринку не однаковий, що в свою чергу робить розрив у запасах занадто великим. Для того, аби вирівняти дані, відсутні значення заповнюються 0, тому в результаті будуть отримані розріджені підматриці. У той же час, щоб зменшити розмірність даних і відфільтрувати нерелевантну інформацію, додають вбудований прошарок перед прошарком LSTM для досягнення точного прогнозування запасів. Принцип роботи прошарків наступний [18]:

- Вбудований прошарок. Вбудований прошарок ініціалізується випадковою матрицею. Відбувається перетворення високорозмірних даних в низькорозмірні за допомогою матричного перетворення. У процесі навчання моделі, матриця у вбудованому прошаркові навчається як параметр мережі.
- LSTM прошарок. Після обробки зменшеної розмірності у вбудованому прошарку, отримують вектор реєстра. LSTM циклічно зчитує опорний вектор і далі витягує інформацію про особливості. Потім він прогнозує значення запасу і порівнює його зі стандартним значенням. Нарешті він оновлює параметри через процес зворотного поширення помилки в моделі LSTM.
- Вихідний прошарок. Для регресійного прогнозування вихідний прошарок має один нейрон. Для прогнозування напрямку підйомів і спадів вихідний прошарок має два нейрони.

Значення кожного вузла автоматичного кодера дорівнює значенню $\{0,1\}$, а вся мережа налаштовується за допомогою випадкового градієнтного спуску. Результат навчання залежить від вибору початкової ваги. Якщо початкове

значення занадто велике, легко впасти в локальну оптимізацію. Якщо вибір занадто малий, а градієнтний спуск дуже повільний, оновлена вага сповільнюється, що призводить до збільшення часу навчання. Тому він не підходить для зменшення розмірності безперервних даних про запаси [18]. У даному дослідженні використовується автоматичний кодер з безперервною обмеженою машиною Больцмана (CRBM) для даних про запаси. Кожні два прошарки мережі навчаються початкової вазі за допомогою CRBM, а верхній лінійний – RBM призначений для виведення даних про запаси після зменшення розмірності.

Для нижчого CRBM вхід кожного вузла у прихованому прошарку пов'язаний зі станами всіх вершин у візуальному прошарку. У дослідженні вважається, що $\{s_i\}$ це множина станів візуального прошарку, $\{s_j\}$ множина станів прихованого прошарку, а φ_j сигмоїдна функція активації.

$$s_j = \varphi_j(\sum_i w_{ij} s_i + b_i) \quad (1.16)$$

Стан вузла прихованого прошарку все ще $\{0,1\}$, а конкретне значення дорівнює наступному:

$$s_j = \begin{cases} 0, s_j < U(0,1) \\ 1, s_j > U(0,1) \end{cases} \quad (1.17)$$

$U(0,1)$ представляє рівномірний розподіл в $(0,1)$. Для верхнього лінійного RBM додається випадкова безперервна одиниця гауссівського шуму:

$$s_j = \varphi_j(\sum_i w_{ij} s_i + b_i + N_j(0,1)) \quad (1.18)$$

$N_j(0,1)$ це гауссівська випадкова величина із середнім значенням 0 та дисперсією 1, а функція активації є лінійною функцією. У дослідженні значення кожного візуального прошарку RBM подається ймовірністю активації у прихованому прошарку попереднього RBM. Крім верхнього RBM, інші RBM прихованого прошарку вузли є випадковими двійковими значеннями.

Модель AELSTM формується шляхом введення вектора запасу, зменшеного вектора, розмірність якого зменшена за допомогою автоматичного кодера до LSTM. Модуль LSTM має той самий алгоритм, що і LSTM в моделі

ELSTM. Зазвичай AELSTM використовують для вивчення характеристик даних, а потім прогнозують вихідні дані.

1.3 Висновок до розділу 1

У даному розділі було розглянуто поняття криптовалюти, а саме Bitcoin та Ethereum, а також поняття блокчейну та майнінгу. Було розглянуто нормативно-правове забезпечення, яким регулюється криптовалюта на території України.

Були розглянуті сучасні моделі машинного навчання для аналізу криптовалютних транзакцій, а саме:

- Гібридні моделі, що формуються шляхом поєднання різних методів машинного навчання (МН)
- Моделі Jang та Lee, які спрогнозували ціну біткоіна за допомогою байєсівської мережі (БМ).
- Модель Mittal, який прогнозує ціни десяти криптовалют за допомогою багатовимірної лінійної регресії [3].
- Модель Poongodi для прогнозування ціни використовував дві моделі: лінійну регресію та машину опорних векторів [4].
- Модель Ato, який використовував щоденні заголовки новин і твіти для прогнозування цінових коливань криптовалют, застосовуючи чотири алгоритми класифікації [5].
- Модель Chang, який пропонує новий підхід до деталізації даних інженерії признаков

РОЗДІЛ 2

НЕЙРОННІ МЕРЕЖІ ЯК МОДЕЛЬ КОТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ

2.1 Класифікація нейронних мереж та їх архітектура

Нейронні мережі стали одним з найпотужніших інструментів у сфері машинного навчання та штучного інтелекту. Їхні можливості охоплюють широкий спектр задач починаючи від розпізнавання зображень, і закінчуючи прогнозуванням аномалій. Для того, аби розробити систему виявлення та прогнозування аномалій під час криптовалютних транзакцій, класифікуємо нейронні мережі за їх архітектурою.

2.1.1 Нейронні мережі прямого поширення

Багатошаровий перцептрон – це тип нейронної мережі прямого поширення, що складається з повністю з'єднаних нейронів з нелінійною функцією активації [31]. Вони широко використовуються для розрізнення даних, які не можуть бути лінійно розділені. Штучні нейронні мережі широко використовуються в різних галузях, таких як розпізнавання зображень, обробка природної мови та розпізнавання мови. Завдяки своїй архітектурній гнучкості та здатності апроксимувати будь-яку функцію за певних умов, вони стали фундаментальним будівельним блоком глибокого навчання та досліджень нейронних мереж. Архітектура багатошарового перцептрона наведено на рисунку 1.

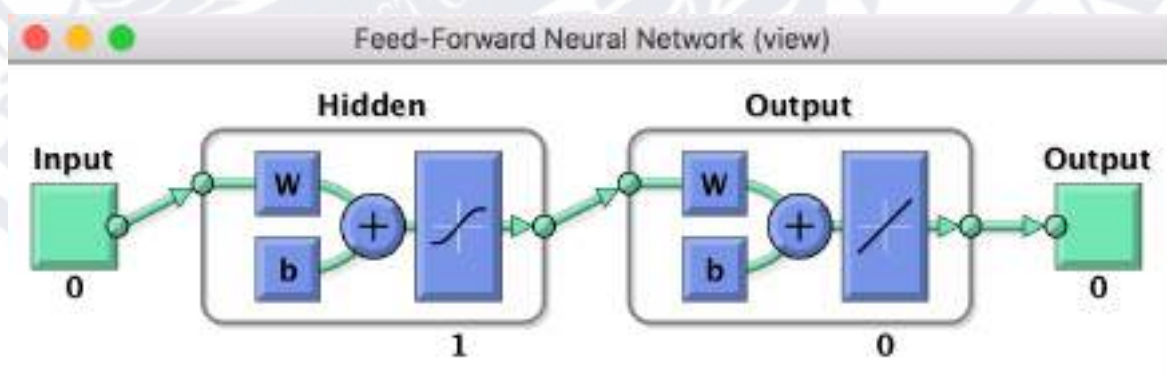


Рис.2.1. Архітектура багатошарового перцептрона (додаток А)

Вхідний прошарок (Input). Вхідний прошарок складається з вузлів або нейронів, які отримують початкові вхідні дані [32]. Кожен нейрон представляє характеристику або розмірність вхідних даних. Кількість нейронів у вхідному прошарку визначається кількістю вимірів вхідних даних.

Прихований прошарок (Hidden). Між вхідним та вихідним прошарками можна розмістити один або декілька прошарків нейронів [33]. Кожен нейрон у прихованому прошаркові отримує вхідні дані від усіх нейронів попереднього прошарку (вхідного прошарку або іншого прихованого прошарку) і виробляє вихідні дані, які передаються до наступного прошарку [34]. Кількість прихованих прошарків і кількість нейронів у кожному прихованому прошарку є гіперпараметрами, які необхідно визначити на етапі проектування моделі.

Вихідний прошарок (Output). Цей прошарок складається з нейронів, які виробляють кінцевий результат роботи мережі [34]. Кількість нейронів у вихідному прошаркові залежить від характеру завдань: у бінарній класифікації – один або два нейрони, що представляють ймовірність належності до одного класу, залежно від функції активації.

2.1.2 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) відрізняються від інших нейронних мереж своєю чудовою продуктивністю при обробці зображень, мови або аудіосигналів. Вони мають три основних прошарки [34]:

- згортковий прошарок;
- об'єднувальний прошарок;
- повнозв'язний прошарок.

Згортковий прошарок є першим прошарком згорткової мережі. За згортковим прошарком можуть слідувати інші згортковані або зв'язні прошарки, але повністю зв'язним є останній прошарок. З кожним наступним прошарком штучна нейронна мережа ускладнюється і описує більші частини зображення. Попередні прошарки фокусуються на простих елементах, таких як кольори та краї. Коли дані зображення проходять через кожен прошарок згорткової

нейронної мережі, вона розпізнає більші елементи і форми об'єктів та ідентифікує об'єкт, що цікавить користувача [35]. Структурна схема згорткової нейронної мережі представлена нижче.

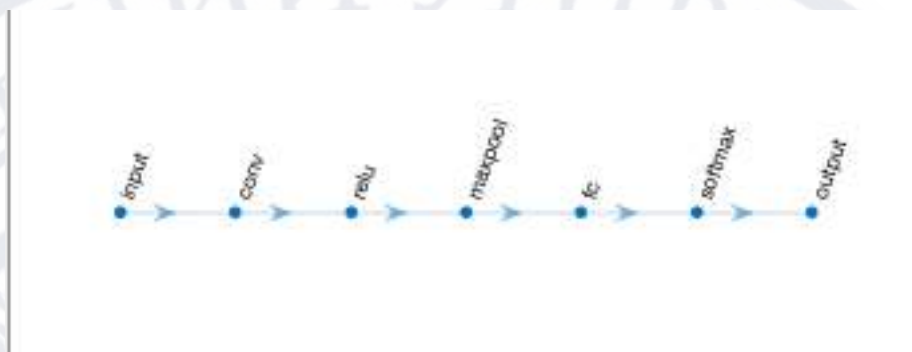


Рис 2.2. Структурна схема згорткової нейронної мережі (додаток Б)

Згортковий прошарок (Conv). Згортковий прошарок є основним будівельним блоком CNN, де виконується більшість обчислень [34]. Згортковий прошарок складається з декількох елементів:

- вхідних даних;
- фільтрів;
- карт ознак.

Певні параметри, як от значення ваг, змінюються під час навчання за допомогою зворотного розповсюдження та градієнтного спуску. Щоправда є три гіперпараметри, які впливають на об'єм вихідних даних, які необхідно встановити до початку навчання нейронної мережі [35]:

1. Кількість фільтрів впливає на глибину вихідних даних.
2. Крок – це відстань або кількість пікселів, на яку ядро переміщується по вхідній матриці
3. Нульове заповнення використовується, коли фільтри не підходять до вхідного зображення.

Додатковий прошарок згортки. Як вже згадувалося раніше, після першого згорткового прошарку можна додавати ще один. У таких випадках структура згорткової нейронної мережі є ієрархічною, і наступний прошарок може бачити пікселі в межах сприйнятливої області попереднього прошарку.

Об'єднувальний прошарок (maxpool). Об'єднування прошарків зменшує розмірність, зменшуючи кількість параметрів у вхідних даних [35]. Подібно до прошарку згортки, операція об'єднання накладає фільтр на всі вхідні дані, але різниця полягає в тому, що цей фільтр не має вагомих коефіцієнтів. Замість цього ядро застосовує функцію агрегування до значень в межах сприйнятливого поля, заповнюючи вихідний масив. Існує два основних типи об'єднання:

1. Максимальне об'єднання. Коли фільтр рухається по вхідному сигналу, він вибирає пікселі з максимальним значенням, щоб відправити його до вихідного масиву.
2. Усереднене об'єднання. Коли фільтр рухається по вхідному сигналу, він обчислює середнє значення в межах сприйнятливого поля, щоб відправити його у вихідний масив.

Повнозв'язний прошарок (fc). У частково зв'язних прошарках значення пікселів вхідного зображення не пов'язані безпосередньо з вершиною попереднього прошарку [34]. Цей прошарок виконує завдання класифікації на основі ознак, отриманих за допомогою попередніх прошарків та їхніх різних фільтрів.

2.1.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) – це нейронні мережі зі зворотним зв'язком, що спеціалізується на обробці послідовних даних і часових рядів, де на вихід подається новий вхід на кожному часовому кроці [37]. Петля зворотного зв'язку дозволяє нейронній мережі запам'ятовувати минулі дані при обробці наступного входу. Цей процес можна описати як рекурентну обробку, тому таку архітектуру також називають рекурентною нейронною мережею.

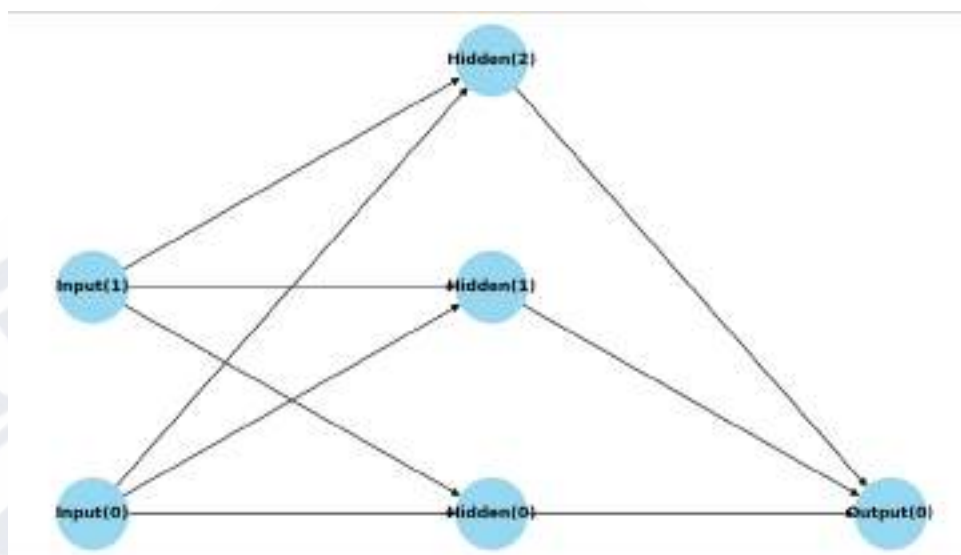


Рис 2.3. Архітектура рекурентної нейронної мережі (додаток В)

Інформація до рекурентної нейронної мережі циклічно надсилається до проміжного прихованого прошарку (Hidden). Вхідний прошарок отримує дані від входу нейронної мережі (Input) та передає його проміжному прошарку. Проміжний прошарок містить кілька прихованих прошарків, кожен з яких зі своєю функцією активації, вагами та зміщеннями [38]. Різні функції активації, ваги та зміщення стандартизуються рекурентною нейронною мережею таким чином, щоб кожен прихований прошарок мав однакові властивості. Замість того, аби створювати багато прихованих прошарків, створюється лише один та повторюється це стільки разів, скільки потрібно.

Рекурентна нейронна мережа – це тип нейронної мережі, що має приховані стан і може використовувати минулі виходи як вхідні дані [39]. Архітектура RNN залежить від проблеми, яку він намагається вирішити. Вона може варіюватися як [39]:

- «Один до одного». Традиційні нейронні мережі використовують дану архітектуру.
- «Один до багатьох». Один вхід може призвести до багатьох виходів.
- «Багато до одного». Багато входів з різних часових кроків об'єднуються, щоб сформувати один вхід.
- «Багато до багатьох». Існує багато варіацій даної архітектури, до прикладу три виходи створюються із двох входів.

2.2 Порівняння нейронних мереж, за їх архітектурою

Нейронні мережі демонструють вражаючі результати в широкому спектрі завдань. Здатність ефективно вчитися на великих обсягах даних робить їх незамінним інструментом для вирішення завдань. У дослідженні, що проведено у даній роботі розглядається три поширені типи нейронних мереж: мережі прямого поширення, згорткові нейронні мережі та рекурентні нейронні мережі. Кожна з них володіє унікальними властивостями, що робить їх придатними для різних задач.

Таблиця 2.1 – Переваги та недоліки нейронних мереж

Нейронні мережі прямого поширення	
<i>Переваги</i>	<i>Недоліки</i>
○ MLP легше зрозуміти, ніж більш складні моделі глибокого навчання, завдяки їхній простій архітектурі та принципам роботи ○ Кількість прошарків і нейронів, функції активації та алгоритми навчання можна налаштовувати для досягнення оптимальних результатів під конкретну задачу. ○ Розуміння того, як працює MLP є ключом для розуміння та розробки більш складних моделей.	○ Алгоритми навчання, що використовуються в MLP можуть не справлятися із обробкою великих обсягів даних, що у свою чергу призводить до повільного навчання та низької точності ○ Навчання MLP займає багато часу і вимагає значних обчислювальних ресурсів, особливо при роботі з великими масивами даних ○ Відсутність чіткого уявлення про внутрішні процеси MLP може ускладнити аналіз їхньої діяльності та виявлення можливих помилок і упереджень.

Продовження таблиці 2.1

Згорткові нейронні мережі	
<i>Переваги</i>	<i>Недоліки</i>
○ CNN можуть автоматично вивчати відповідні функції з необроблених вхідних даних, усуваючи необхідність ручної розробки функцій. ○ CNN можуть досягати найсучаснішої продуктивності в різноманітних завданнях розпізнавання зображення та відео. ○ CNN мають високу стійкість до шуму і спотворень вхідних даних, що робить їх високоефективними в реальних умовах	○ CNN потребують великих обчислювальних ресурсів, включаючи високопродуктивні графічні процесори для навчання та розгортання. ○ Для досягнення високої точності CNN потребують великих обсягів маркованих навчальних даних, які може бути важко отримати в певних областях науки. ○ CNN схильні до перенавчання, коли навчальні дані обмежені або зашумлені.
Рекурентні нейронні мережі	
<i>Переваги</i>	<i>Недоліки</i>
○ RNN може запам'ятовувати попередні входи і виходи. Підвищення запам'ятовування відбувається за рахунок мережі LSTM ○ RNN може вчитися на минулому досвіді, на основі цього вона може робити точні прогнози	○ Алгоритм присвоює вагам необґрунтовано високі значення без жодної конкретної причини ○ Коли значення градієнта занадто мале, модель перестає навчатися або навчається занадто довго ○ RNN обробляє дані послідовно, що може зробити процес навчання нудним

Продовження таблиці 2.1

<i>Переваги</i>	<i>Недоліки</i>
- О RNN ідеально підходить для обробки послідовних даних. оскільки вони розуміють часові аспекти даних. - О RNN мають широкий спектр застосувань, оскільки вони можуть обробляти послідовні дані, такі як часові ряди, аудіо та мовлення.	- О Чим довша послідовність даних, тим складніше для RNN запам'ятовувати попередню інформацію - О RNN обробляє дані послідовно, що може бути трудомістким і неефективним методом

Зважаючи на переваги та недоліки представлених нейронних мереж, можна зрозуміти, що кожна з них має свої переваги у різних сферах. До прикладу згоркові нейронні мережі краще використовувати для обробки аудіо та відео матеріалів; багатопрошарковий персептрон краще використовувати у завданнях, де потрібна класифікація невеликий об'ємів даних; рекурентні мережі краще використовувати у роботі з даними, де важлива часова послідовність та певна динаміка часу.

2.3 Рекурентні нейронні мережі та їх види

Традиційні рекурентні нейронні мережі (RNN) мають обмеження, такі як проблеми довготривалої залежності через проблему зникаючого градієнта. Для подолання цих обмежень було розроблені вдосконалені архітектури, такі як:

- LSTM (Long Short Term Memory) – це тип нейронної мережі, яка має спеціальний блок пам'яті, що дозволяє моделі зберігати та використовувати інформацію протягом тривалого періоду часу, уникаючи проблеми зникаючого градієнта [40].

- GRU (Gated Recurrent Units) використовують менше параметрів, ніж LSTM, але мають схожу здатність моделювати довгострокові залежності [41].
- SRNN (Simple Recurrent Neural Network) – це тип штучної нейронної мережі, яка характеризується на обробці даних, що змінюються з часом [42].

2.3.1 Long Short Term Memory (LSTM)

Довга короткочасна пам'ять (LSTM) є оновленою версією рекурентних нейронних мереж. Вони спеціально розроблені для вирішення проблеми втрати градієнта та уникнення проблем довготривалої залежності шляхом додавання механізму координації інформації, що дозволяє зберігати інформацію протягом тривалих періодів часу. Архітектура LSTM складається з декількох блоків пам'яті, які є рекурсивно з'єднаними підмережами. Блоки пам'яті виконують двійкову функцію, тобто вони зберігають стан мережі в хронологічному порядку і координують потік інформації між комірками [41]. Крок вхідної воронки відповідає за визначення інформації, яка зберігається в стані комірки, в той час як вихід відповідає за обчислення інформації, яка передається зі стану комірки.

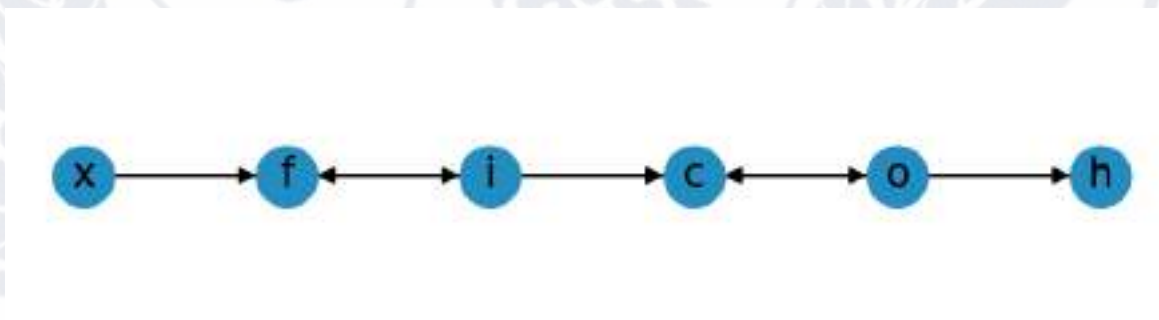


Рис. 2.3. Схема типової моделі LSTM (додаток Г)

Процес прямого навчання мережі LSTM можна описати наступними рівняннями:

На першому етапі прошарок воріт визначає, яка інформація видаляється з пам'яті комірки [14]. Цей рівень має наступну формулу:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

На другому етапі прошарок входу (input gate) і прошарок $\tanh$ визначають, яка нова інформація буде зберігатися в пам'яті комірки. Прошарок входу

визначає, які значення оновлювати, а рівень TANH створює вектор нової кандидатської пам'яті.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \quad (2.2)$$

$$\tilde{C}_t = \tanh((W_c \cdot [h_{t-1}, x_t] + b_c)) \quad (2.3)$$

На третьому етапі відбувається оновлення пам'яті клітин. Це використовується для того, щоб забути те, що раніше було вирішено відкинути, і масштабує значення нових кандидатів відповідно до того, наскільки було змінено значення кожного стану.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.4)$$

На четвертому етапі відбувається оновлення прихованих комірок. Сигмоїдний прошарок вирішує, які частини стану клітини відобразити. Відображаються лише ті частини, які він вирішив приховати.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o); \quad (2.5)$$

$$h_t = o_t * \tanh(C_t), \quad (2.6)$$

Де відповідно:

- x_t – вхід на кроку часу t
- h_t – прихований стан кроку часу t
- C_t – стан клітини на кроку часу t
- i_t, f_t, o_t – вхідна воронка, воронка забуття і вихідна воронка відповідно на крок часу t
- W та b – це матриці ваг і вектори зміщення
- Функція сигмоїда та гіперболічний тангенс ($\tanh$) використовується для обмеження виходу між 0 та 1, а також -1 та 1 відповідно.

2.3.2 Gates Recurrent Units (GRU)

Gates Recurrent Units (GRU) – це тип рекурентної нейронної мережі, який є покращеним варіантом мережі LSTM. Так само, як і довга короткочасна пам'ять, вентильний рекурентний вузол обробляє вхідні послідовності довільної довжини та кодує минулу інформацію, зберігаючи закодований стан [42]. На відміну

від LSTM, які використовують кілька вентилів і комірок внутрішньої пам'яті для управління потоком інформації, GRU використовують один вентиль оновлення, щоб визначити, яку інформацію зберігати та вентиль відкидання, щоб визначити, яку інформацію відкинути. Саме через це вентильний рекурентний вузол простіше і легше піддається навчанню.

Однією з переваг GRU є те, що його можна використовувати для виявлення довгострокових залежностей у послідовних даних більш ефективно, ніж використання простої рекурентної мережі. Це пов'язано з тим, що вентиля оновлення та скидання дозволяють вибірково зберігати або забувати минулу інформацію залежно від поточних вихідних даних та стану мережі. Це робить GRU особливо придатним для завдань, які вимагають здатності зберігати і використовувати довгі послідовності інформації.

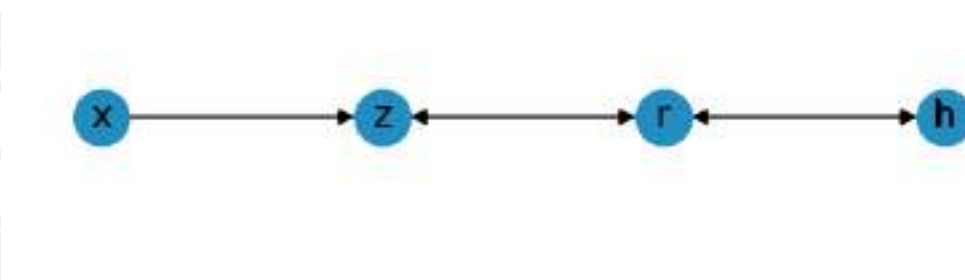


Рис. 2.4. Схема GRU (додаток Д)

Формування GRU можна описати наступними рівняннями [44]:

$$r_t = \text{sigm}(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (2.7)$$

$$z_t = \text{sigm}(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (2.8)$$

$$\tilde{h}_t = \text{tanh}(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (2.9)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \quad (2.10)$$

де відповідно:

r_t – відповідає воротовим блокам скидання

z_t – відповідає воротовим блокам оновлення

x_t – вхідний вектор

h_t – вихідний вектор

W – матриця ваг

b – зміщення

sigm – активаційна функція сигмоїду

tanh – гіперболічна тангенс активаційна функція

2.3.3 Simple Recurrent Neural Network (SRNN)

Проста рекурентна нейронна мережа (SRNN) – це тип штучної нейронної мережі, яка характеризується на обробці даних, що змінюються з часом [43]. Дані мережі «пам'ятають» попередні результати, дозволяючи їм виявляти закономірності та робити прогнози. На відміну від звичайних нейронних мереж, що працюють лише з поточними даними, SRN використовує попередню інформацію.

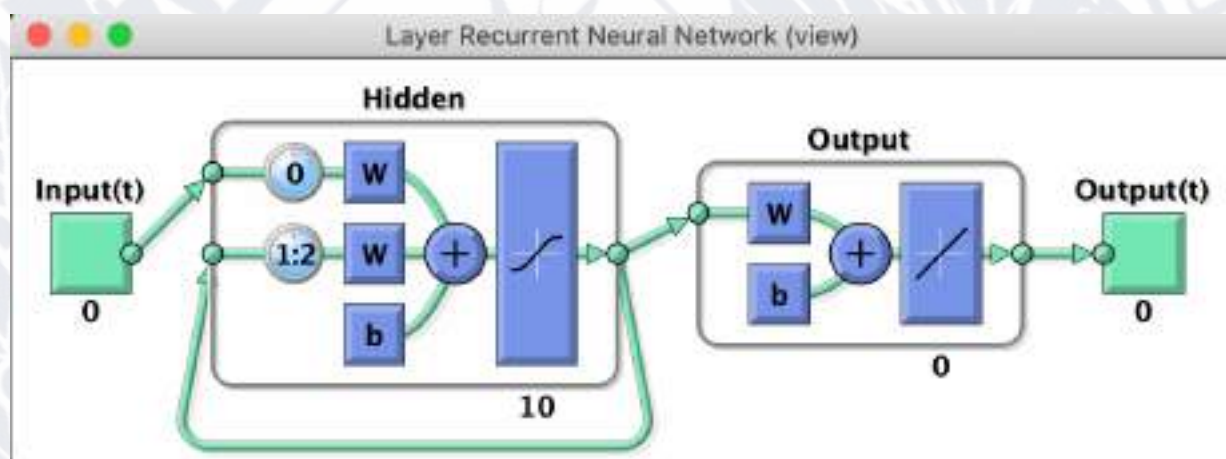


Рис 2.5. Архітектура простої рекурентної нейронної мережі (додаток Е)

На рисунку 1.2 представлено архітектуру простої рекурентної нейронної мережі, де: [45]

- Вхідний прошарок (input) отримує послідовність даних;
- Прихований прошарок (hidden) включає в себе петлі зворотного зв'язку, що дозволяють посылатись на інформацію отриману раніше;
- Вихідний прошарок (output) генерують вихідні дані, що залежать як від поточних, так і від попередніх значень.

У часових рядах поточне значення (ht) зазвичай залужить від попереднього значення ($ht-1$). Саме ця характеристика дозволяє простій рекурентній нейронній мережі (SRN) ефективно аналізувати дані та робити прогнози на майбутнє [46]. Щоправда, алгоритми SRN мають обмеження при застосуванні до довгих

часових рядів, оскільки пам'ять для зберігання даних в них дуже мала. Крім того, при використанні кількох прихованих прошарків SRN для вирішення складних задач із часовими рядами часто виникають проблеми зникнення градієнту.

2.4 Порівняння рекурентних нейронних мереж

Рекурентні нейронні мережі демонструють потужні результати під час аналізу послідовних даних. У дослідженні, що проведено у даній роботі розглядається три поширені типи рекурентних нейронних мереж: LSTM мережа, GRU мережа та SRNN мережа. Кожна з них володіє унікальними властивостями, що робить їх придатними для задач у роботі із рядами послідовних даних. Таблиця, що представлена нижче містить більше детальних огляд характеристик кожної із моделей мережі та допоможе обрати найбільш оптимальну для даного завдання рекурентну нейронну мережу.

Таблиця 2.2 – Порівняння рекурентних нейронних мереж за їх властивостями

	LSTM	GRU	SRNN
Архітектура	Найскладніша архітектура, містить багато параметрів	Складна архітектура, містить багато параметрів	Найпростіша архітектура, містить найменше параметрів
Пам'ять	Висока здатність запам'ятовувати довгострокові залежності	Висока здатність запам'ятовувати довгострокові залежності	Середня здатність запам'ятовувати довгострокові залежності
Швидкість	Повільна швидкість навчання	Середня швидкість навчання	Висока швидкість навчання

Продовження таблиці 2.2

Переваги	Гарно працює із довгостроковими залежностями	Гарно та швидко працює із довгостроковими залежностями	Проста у навчанні та швидка за рахунок малої кількості параметрів
Недоліки	Повільна у навчанні за рахунок великої кількості параметрів	Не результативна у випадку складних завдань (прогнозування)	Обмежена у роботі із довгостроковими залежностями.
Застосування	Обробка та розпізнавання мови, прогнозування часових рядів.	Обробка відео та аудіо, розпізнавання мови та генерація музики.	Прогнозування короткострокових часових рядів та поведінки системи, виявлення аномалій.

2.5 Висновки до розділу 2

В результаті огляду літературних джерел в курсовій роботі були класифіковані нейронні мережі за їх архітектурою та для аналізу обрано наступні три види архітектур:

- нейронні мережі прямого поширення;
- згорткові нейронні мережі;
- рекурентні нейронні мережі.

Базуючись на проведеній класифікації були визначені переваги та недоліки кожної з мереж. Для роботи із аналізом криптовалютних транзакцій було обрані рекурентні нейронні мережі.

Вибір рекурентних нейронних мереж для обробки даних на основі криптовалютних транзакцій зумовлений їхньою здатністю ефективно моделювати послідовності даних. Оскільки дані про транзакції змінюються з часом і можуть містити складні закономірності рекурентні нейронні мережі виявилися потужним інструментом для виявлення аномалій та прогнозування руху цін на криптовалюту. На основі цього було розглянуто три типи рекурентних нейронних мереж, а саме:

- мережа довгої короткочасної пам'яті (LSTM);
- вентильний рекурентний вузол (GRU);
- просту рекурентну нейронну мережу (SRNN).

На основі проведеного аналізу для роботи із криптовалютними транзакціями було обрано оптимальну мережу – SRNN мережу.

РОЗДІЛ 3

РОЗРОБКА МЕТОДИКИ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ

3.1 Криптовалюта, як різновид фінансових активів

Із розвитком цифрової індустрії громадяни різних країн почали активно переносити своє життя у комп'ютери. До прикладу, Україна стала першою країною у світі, яка використовує електронні паспорти, розробила портал ДІЯ, де люди можуть отримувати різні послуги та документи. Чи можна це назвати цифровими активами? Цифрові цінні папери, електронні бази даних, облікові записи, цифрові документи – все це відноситься до цифрових активів. У даній роботі ми будемо розглядати криптовалюту, як один з основних різновидів поняття «цифровий актив».

Криптовалюти все більше закріплюються на фінансових ринках, де щодня відбувається значна кількість транзакцій та угод. Подібно до інших фінансових систем, передбачення цінових змін, є однією з ключових проблем у сфері криптовалютної торгівлі. Ідея використання штучного інтелекту, як одного з інструментів прогнозування, стало популярною темою досліджень у криптовалютній індустрії. Моделі машинного навчання здаються ідеальними для вирішення проблеми прогнозування цін на криптовалютному ринку [47].

Для аналізу динаміки цін на цифрові активи, включаючи біткойн, будуть використані відкриті дані про ціни на криптовалюту за останнє десятиліття. Цей набір даних дозволить промодельовати та проаналізувати фундаментальні тенденції на ринку криптовалют. Це перший крок для розробки ефективних інструментів для моніторингу та контролю за обігом цифрових активів.

Для отримання кращого результату, ми будемо використовувати дані в період з 01.08.2023 рік по 01.09.2023 рік, що наведені у таблиці нижче.

Таблиця 3.1 - Ціни на біткойн в період від 01.08.2023 по 01.09.2023

Data	Closing Price (USD)	24h Open (USD)	24h High (USD)	24h Low (USD)
01.08.2023	29675.732422	29230.873047	29675.732422	28657.023438
02.08.2023	29151.958984	29704.146484	29987.998047	28946.509766
03.08.2023	29178.679688	29161.8125	29375.707031	28959.488281
04.08.2023	29074.091797	29174.382813	29302.078125	28885.335938
05.08.2023	29042.126953	29075.388672	29102.464844	28957.796875
06.08.2023	29041.855469	29043.701172	29160.822266	28963.833984
07.08.2023	29180.578125	29038.513672	29244.28125	28724.140625
08.08.2023	29765.492188	29180.019531	30176.796875	29113.814453
09.08.2023	29561.494141	29766.695313	30093.435547	29376.800781
10.08.2023	29429.591797	29563.972656	29688.564453	29354.447266
11.08.2023	29397.714844	29424.902344	29517.773438	29253.517578
12.08.2023	29415.964844	29399.787109	29465.113281	29357.587891
13.08.2023	29282.914063	29416.59375	29441.433594	29265.806641
14.08.2023	29408.443359	29283.263672	29660.253906	29124.105469
15.08.2023	29170.347656	29408.048828	29439.121094	29088.853516
16.08.2023	28701.779297	29169.074219	29221.976563	28701.779297
17.08.2023	26664.550781	28699.802734	28745.947266	25409.111328
18.08.2023	26049.556641	26636.078125	26808.195313	25668.921875
19.08.2023	26096.205078	26047.832031	26249.449219	25802.408203
20.08.2023	26189.583984	26096.861328	26260.681641	26004.314453
21.08.2023	26124.140625	26188.691406	26220.201172	25846.087891
22.08.2023	26031.65625	26130.748047	26135.507813	25520.728516
23.08.2023	26431.640625	26040.474609	26786.898438	25804.998047
24.08.2023	26162.373047	26431.519531	26554.910156	25914.925781
25.08.2023	26047.667969	26163.679688	26248.103516	25786.8125
26.08.2023	26008.462891	26047.234375	26107.384766	25983.878906
27.08.2023	26089.693359	26008.242188	26165.373047	25965.097656
28.08.2023	26106.150391	26089.615234	26198.578125	25880.599609
29.08.2023	27727.392578	26102.486328	28089.337891	25912.628906
30.08.2023	27297.265625	27726.083984	27760.160156	27069.207031
31.08.2023	25931.472656	27301.929688	27456.078125	25752.929688
01.09.2023	25800.724609	25934.021484	26125.869141	25362.609375

З таблиці 3.1 бачимо, що ми маємо чотири представлених ціни з якими надалі будемо працювати:

- кінцева ціна (closing price);
- стартова ціна на час початку доби (24h open);

- найвища ціна за час доби (24h high);
- найнижча ціна за час доби (24h low).

3.2 Розробка методики контролю за обігом цифрових активів

Для розробки методики контролю за обігом цифрових активів ми будемо дотримуватись певного плану. Для отримання результату ми будемо використовувати програму MATLAB R2017b [48].

Перш за все нам потрібно завантажити та відфільтрувати діапазон обраних даних від пустих та неналежних значень вибраних даних у програмне середовище. Код представлено на рисунку 3.1

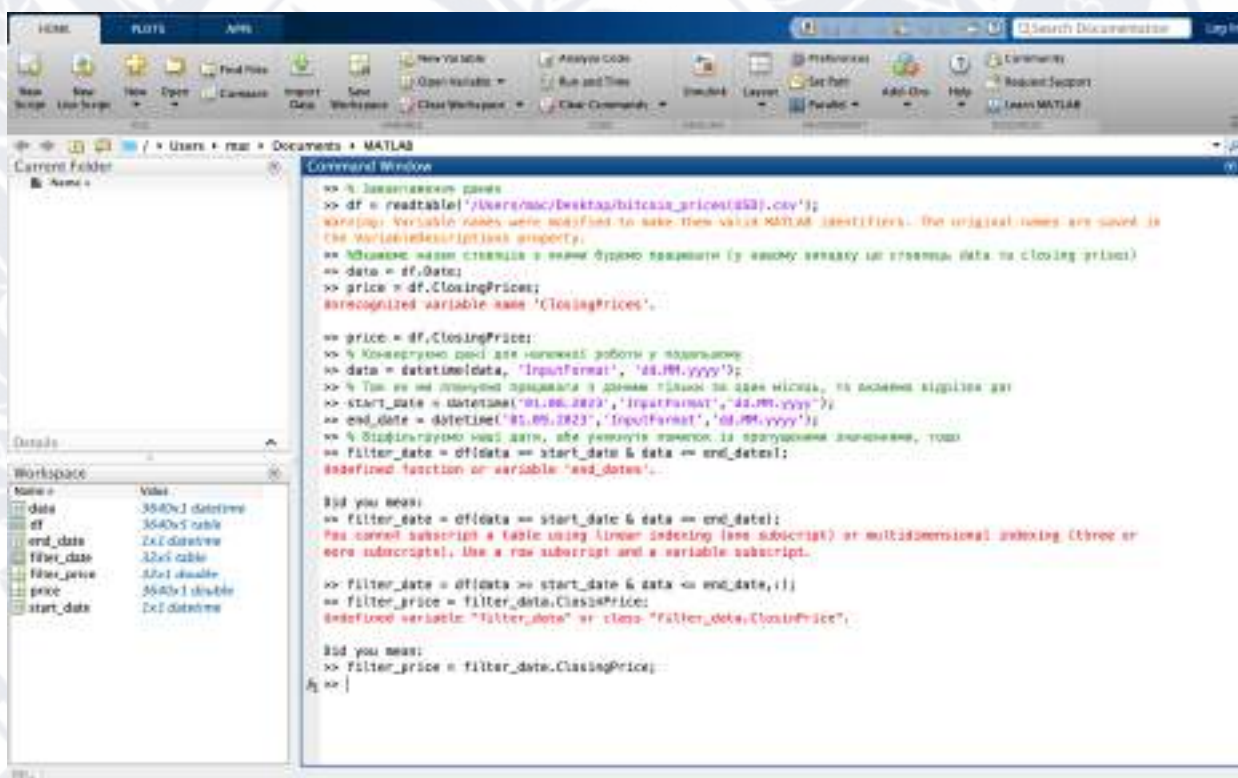


Рис 3.1 Завантаження та фільтрування даних

Другим етапом роботи є розбиття набору даних на навчальний набір та тестовий. Зазвичай використовують схему 80 на 20, тобто 80 % даних – це дані з яких буде навчатись наша мережа, а 20 % - це дані які будуть тестуватися для отримання результату. Також важливим фактором на даному етапі є

нормалізація даних, яка дозволить прискорити навчання та отримати краще результат.

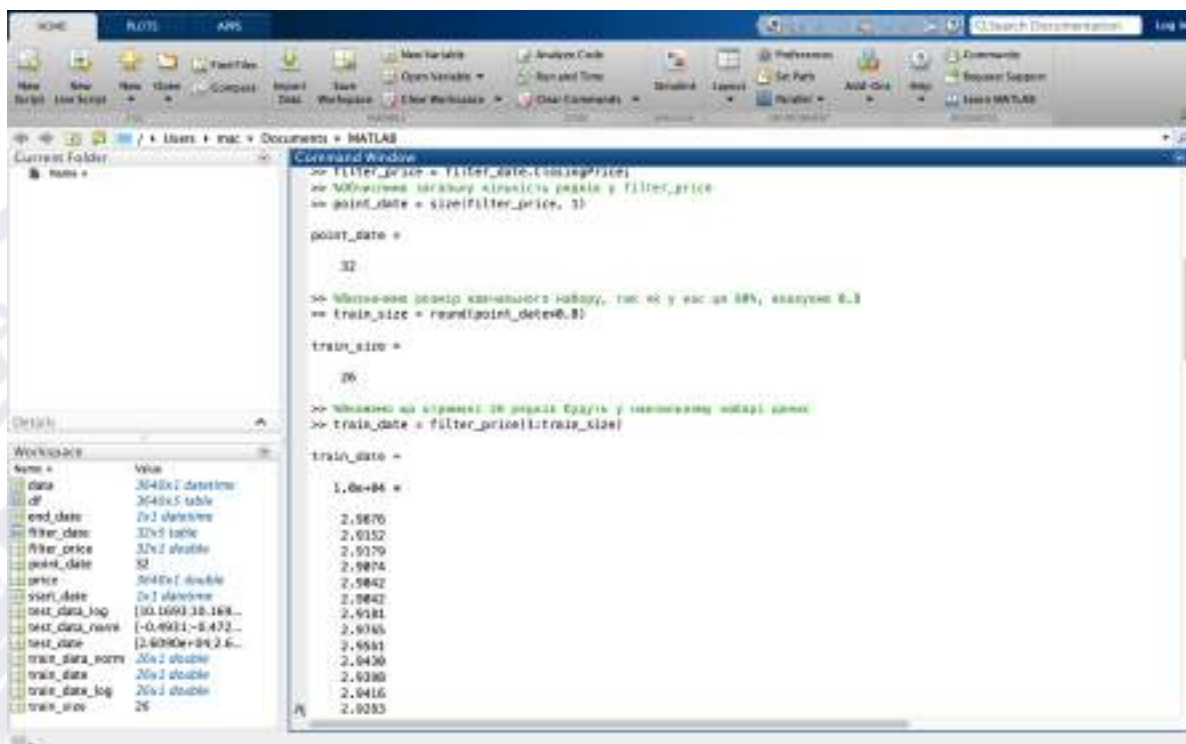


Рис 3.2 Розподіл даних на навчальний та тестовий набори

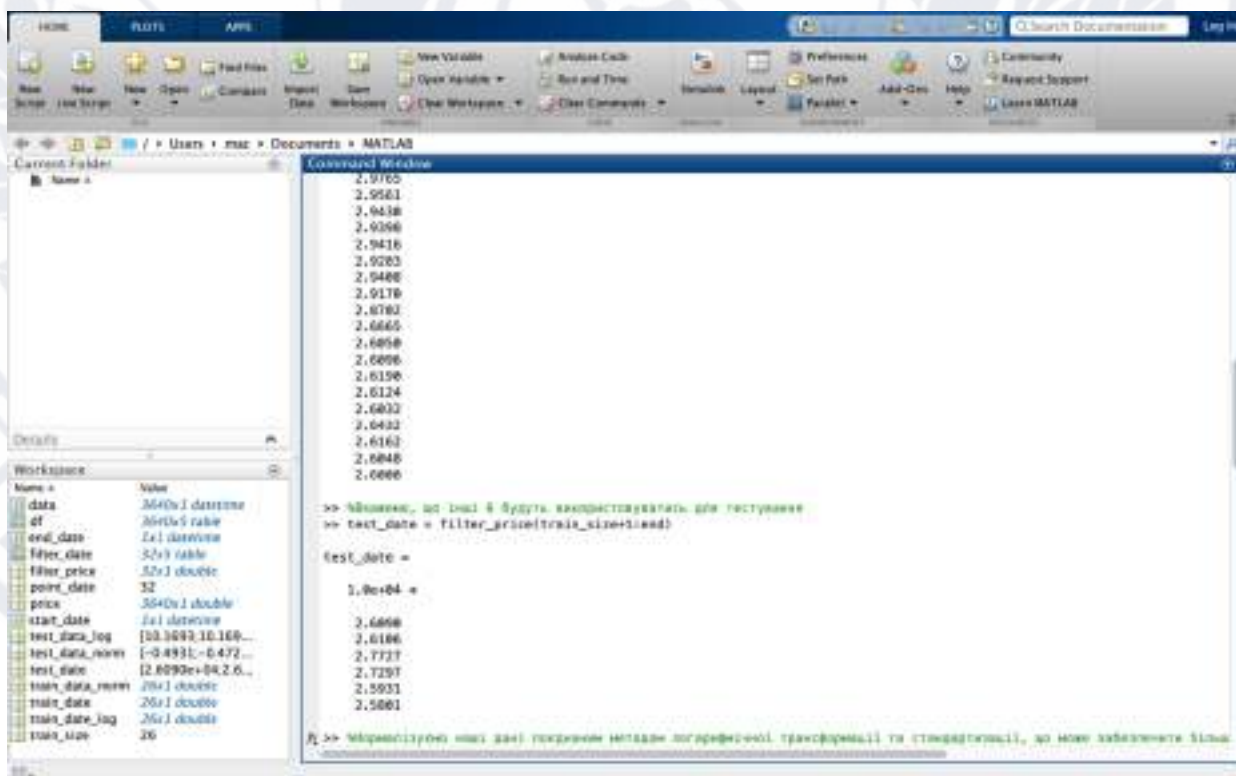


Рис 3.3 Розподіл даних на навчальний та тестовий набори (продовження)

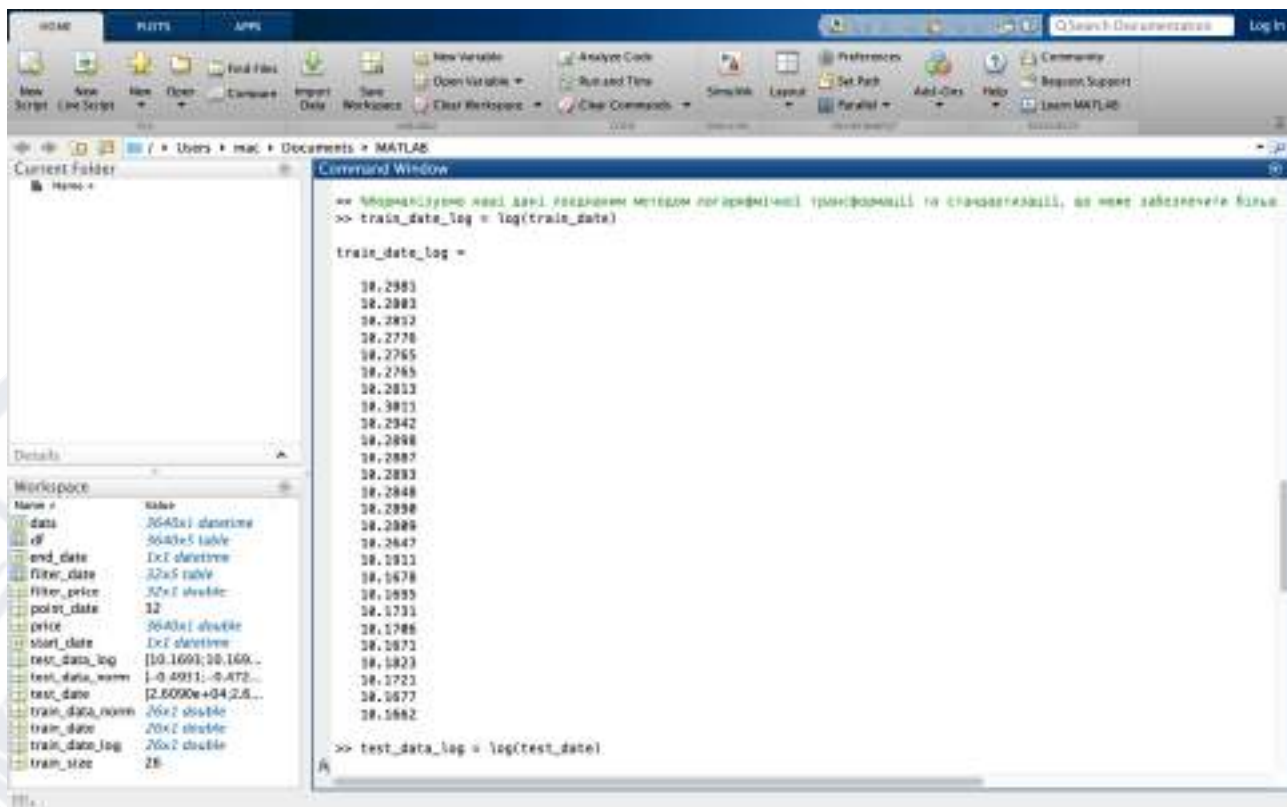


Рис 3.4 Нормалізація даних (частина 1)

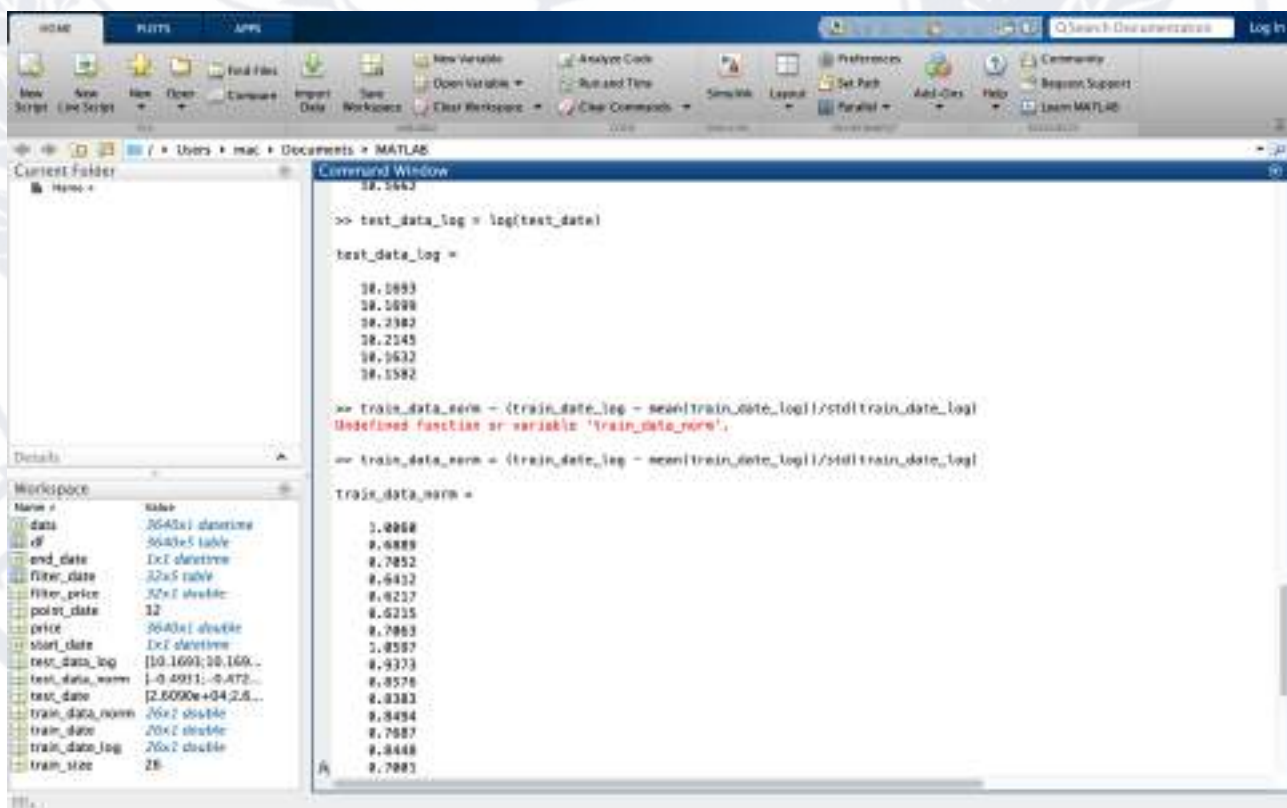


Рис 3.5 Нормалізація даних (частина 2)

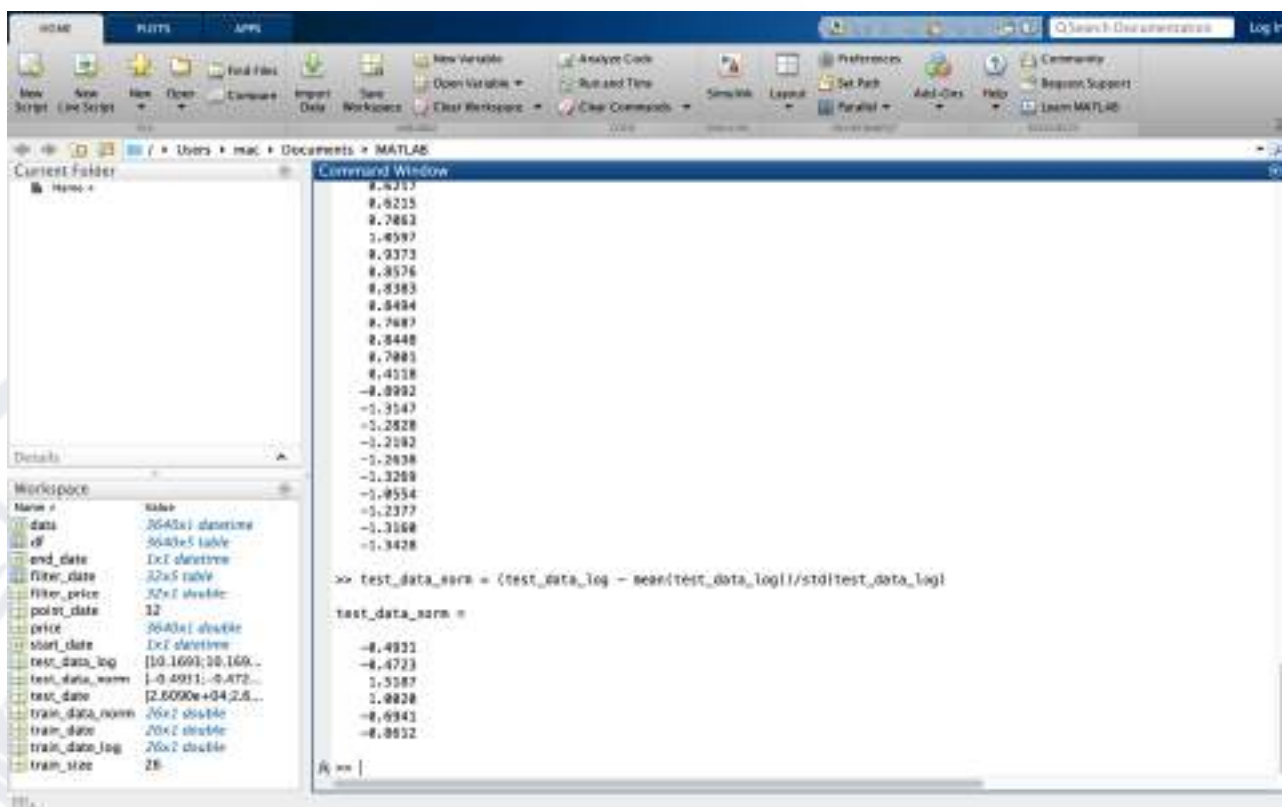


Рис 3.6 Нормалізація даних (частина 3)

Третім кроком є створення простої рекурентної нейронної мережі. На даному етапі повинні визначити параметри мережі, яку маємо навчати. Перш за все визначимо кількість нейронів, так як у роботі використовується не надто велика база даних, ми використаємо 10 нейронів для прихованого (hidden) прошарку. Також експериментально задаємо 50 епох, тобто у нашому дослідженні пройде 50 повних ітерацій під час навчання. Це потрібно для того, аби отримати хороший результат. Також визначимо значення цільової похибки, до прикладу використаємо 10^{-5} .

Щоправда у процесі роботи, було помічено, що для даного часового ряду краще буде застосовувати не 50 епох, а всього лише 8. Це спричинено тим, що при проходженні 50 ітерацій проста рекурентна нейронна мережа перенавчається, що у свою чергу призводить до хибних результатів.

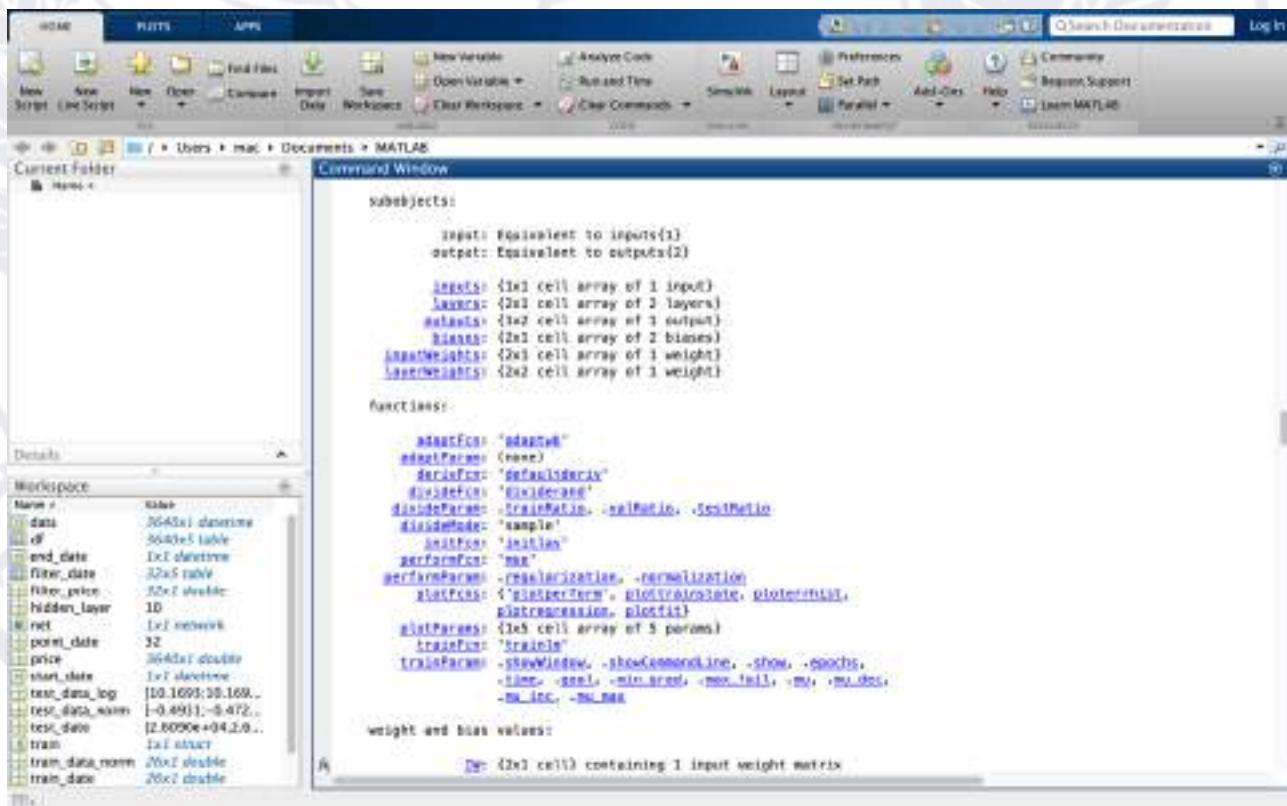


Рис 3.8 Створення простої рекурентної нейронної мережі (частина 2)

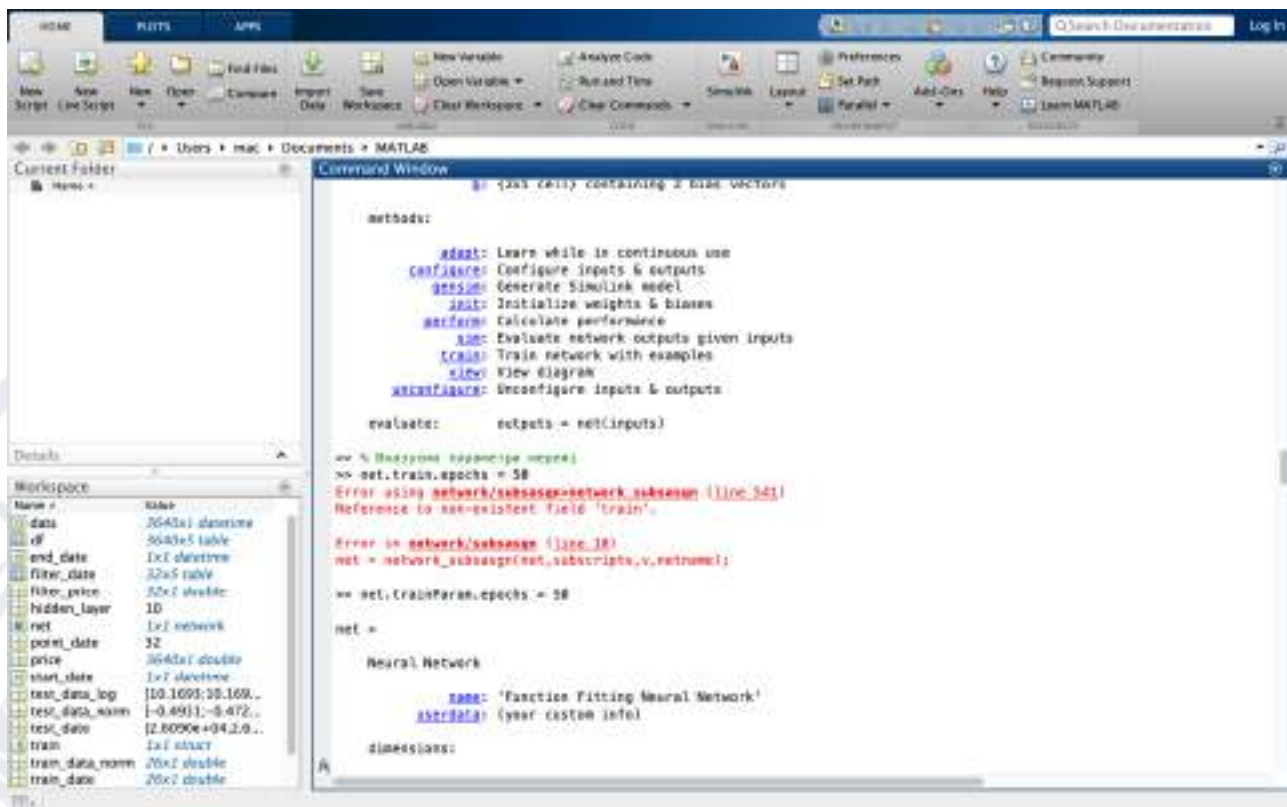


Рис 3.9 Задаємо параметри мережі епохи

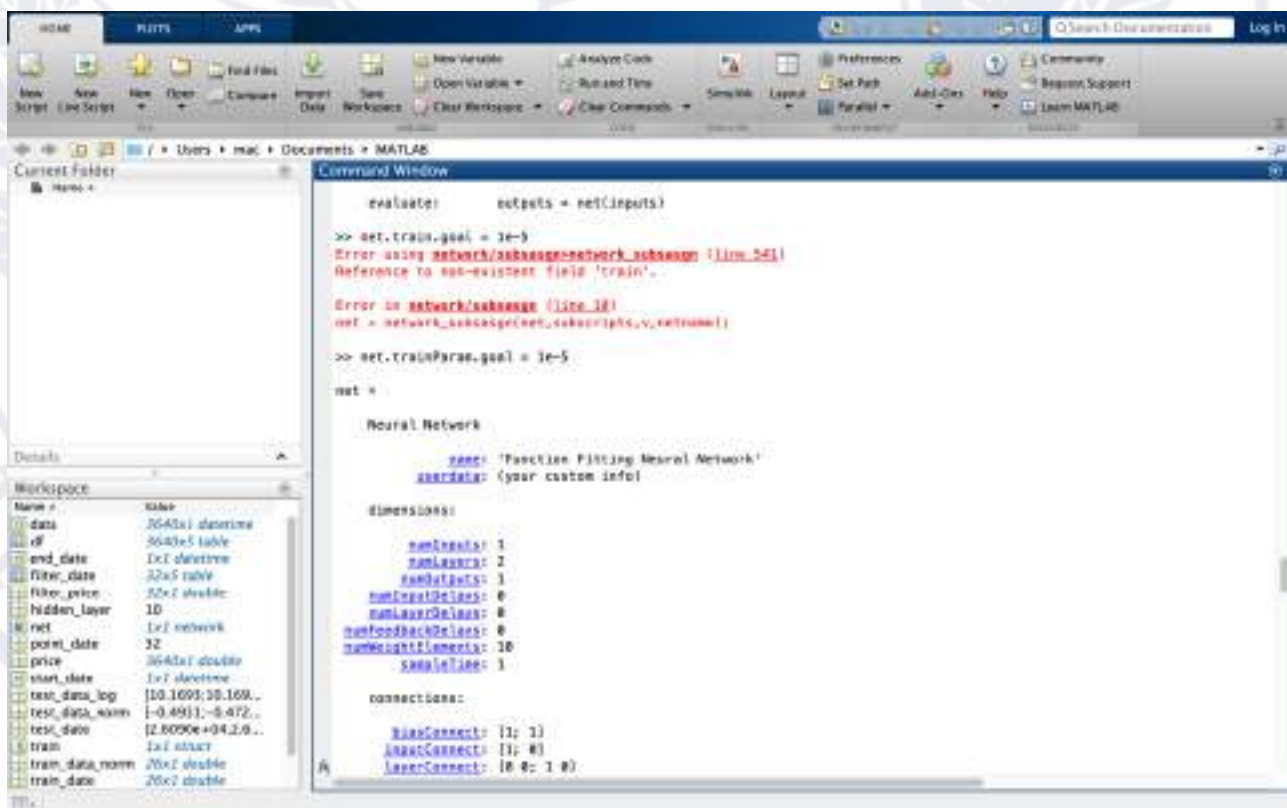


Рис 3.10 Задаємо параметри мережі цільової похибки

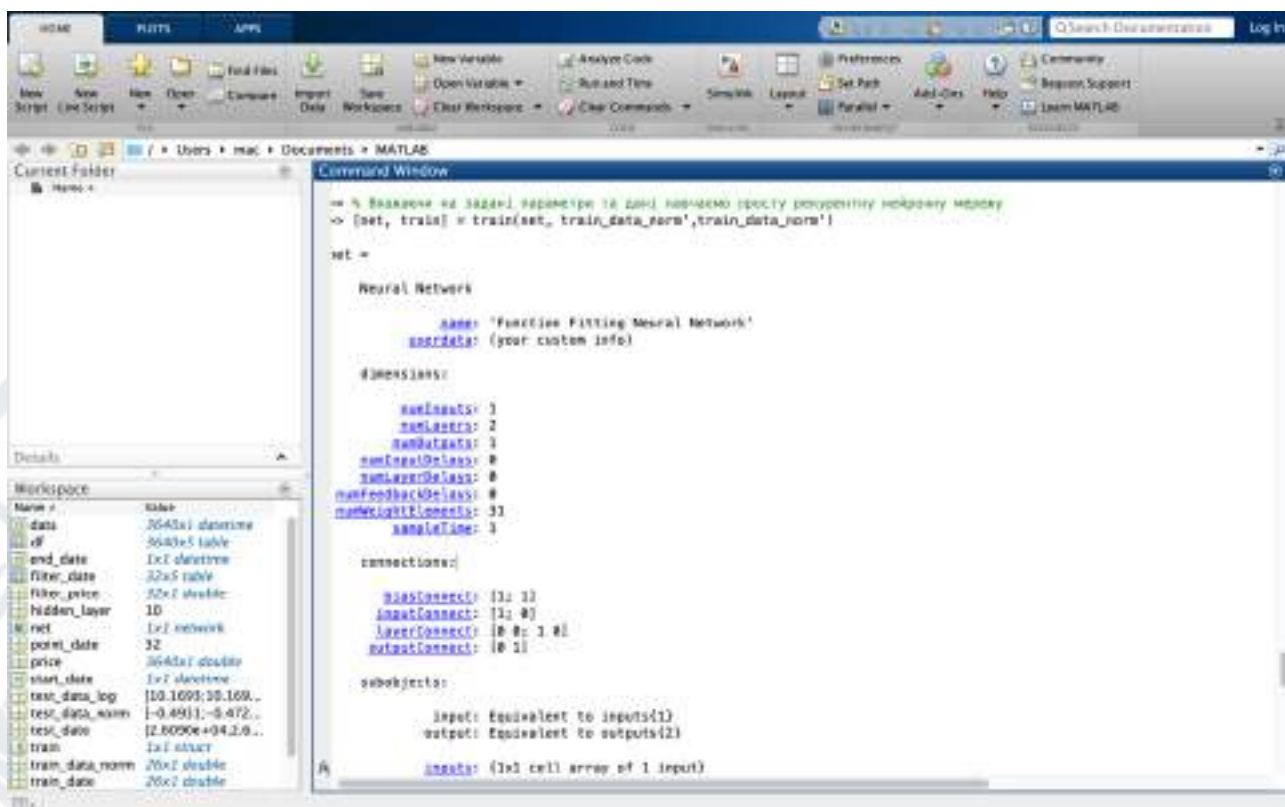


Рис 3.11 Навчання простої рекурентної нейронної мережі (частина 1)

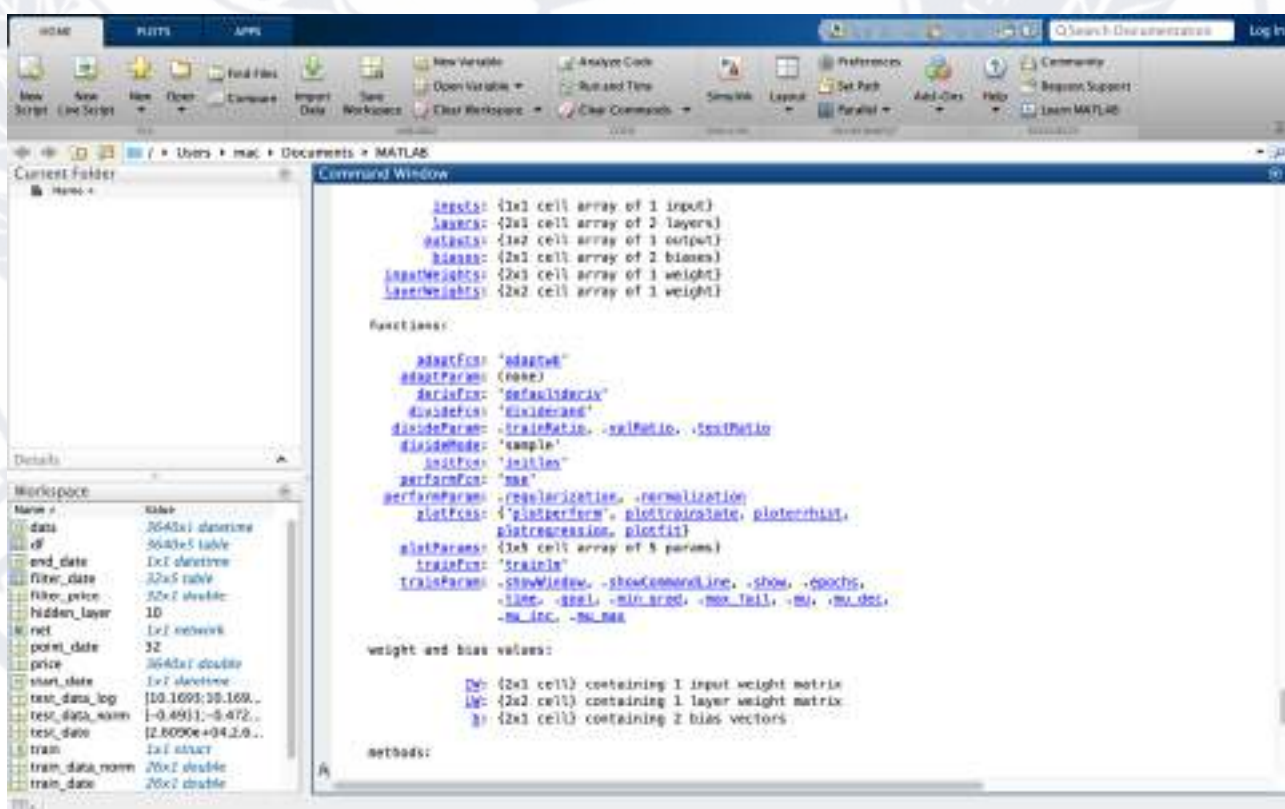


Рис 3.12 Навчання простої рекурентної нейронної мережі (частина 2)

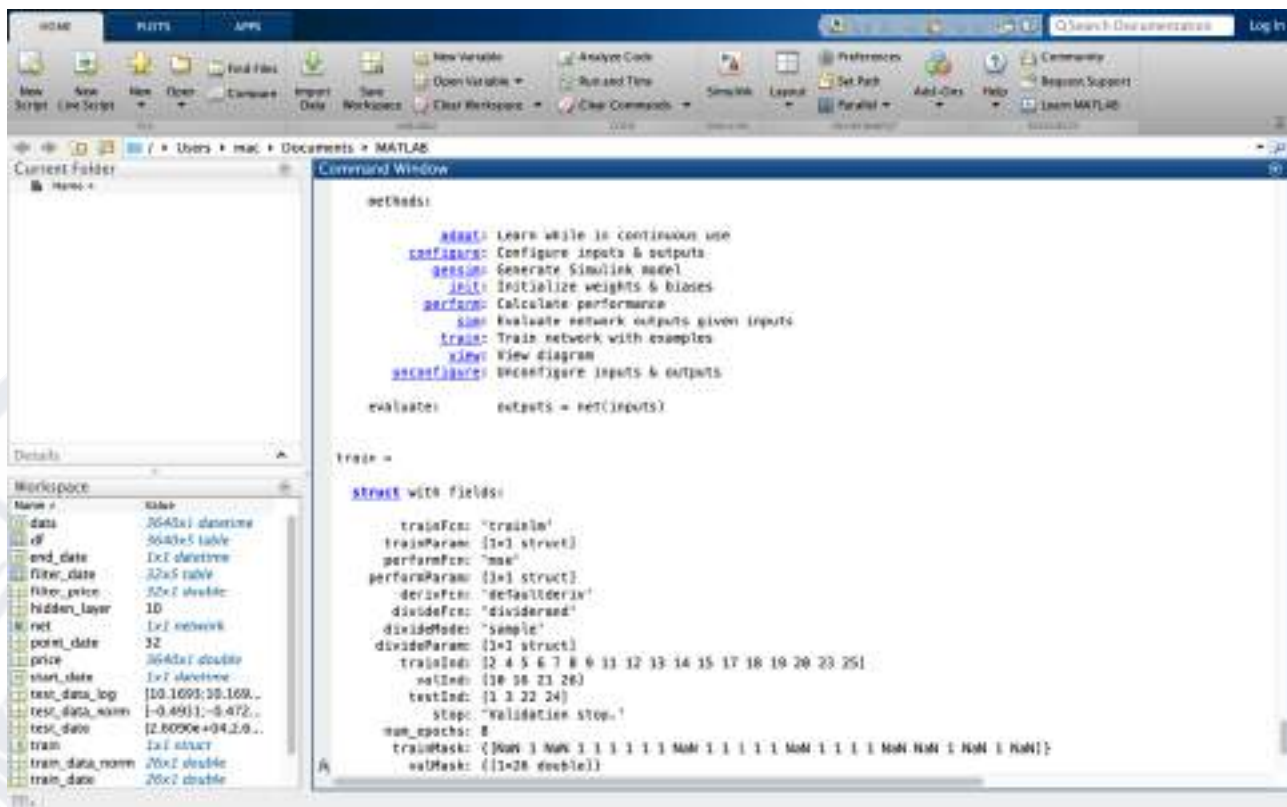


Рис 3.13 Навчання простої рекурентної нейронної мережі (частина 3)

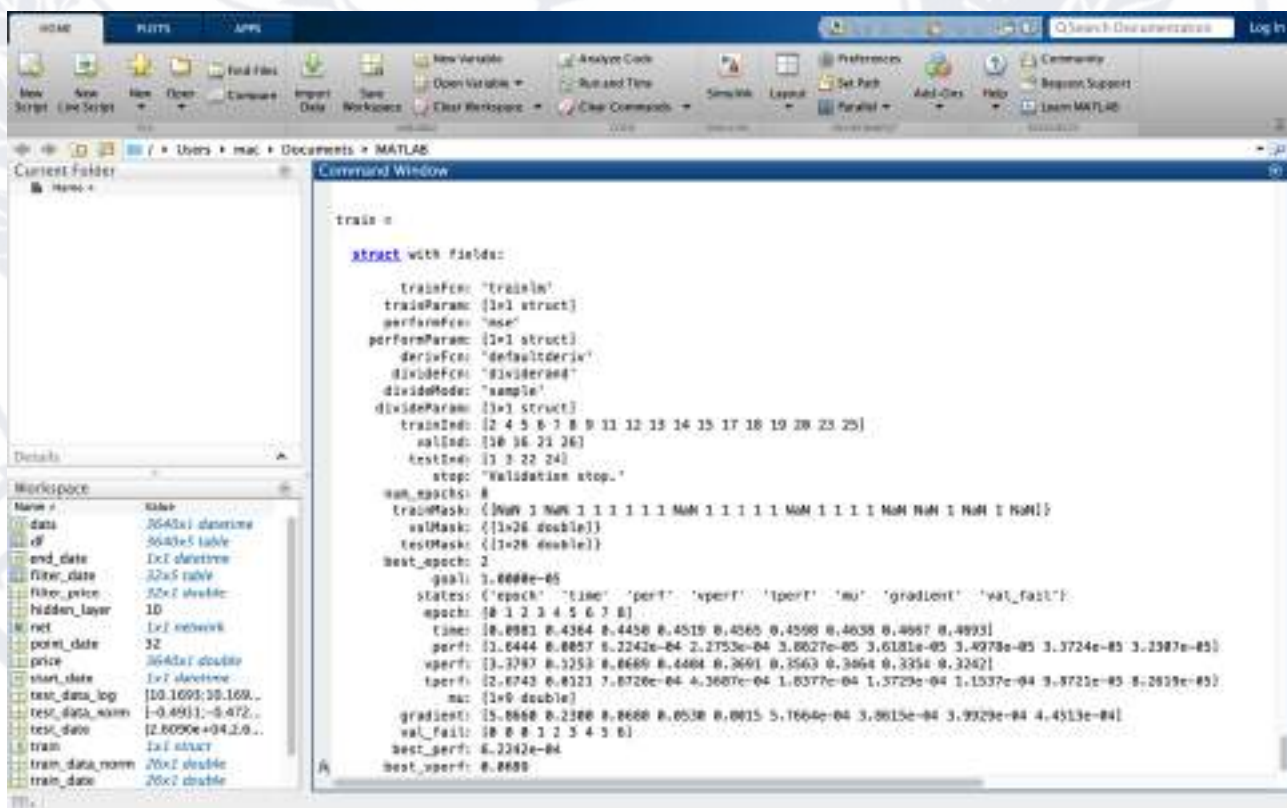


Рис 3.14 Навчання простої рекурентної нейронної мережі (частина 4)



Рис 3.15 Представлення схеми навчання простої рекурентної нейронної мережі

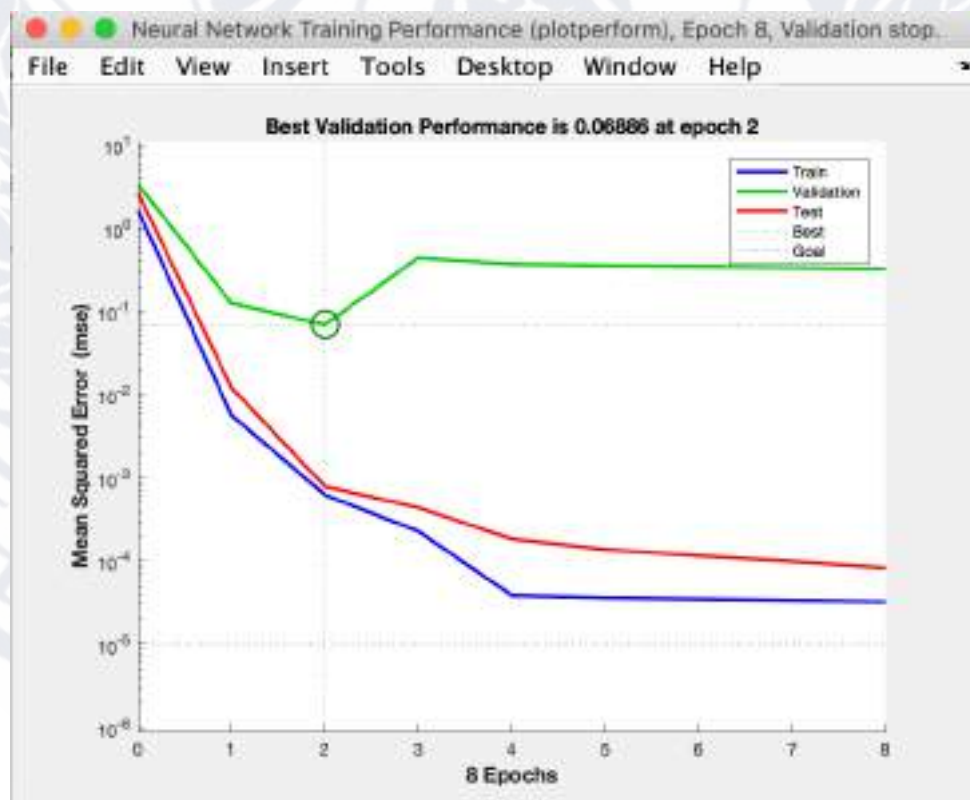


Рис 3.16 Графік ефективності навчання нейронної мережі

На рисунку 3.16 представлено зміну середньоквадратичної похибки (MSE) за 8 епох навчання простої рекурентної нейронної мережі для трьох наборів даних: навчання, валідація та тестування. Зміна середньоквадратичної похибки показує наскільки добре модель прогнозує вихідні дані. З цього графіку можна зробити висновки, що модель простої рекурентної нейронної мережі була успішно навчена у перші дві епохи. Щоправда, подальше навчання може призвести до перенавчання. Аби отримати оптимальну модель краще за все зупинитися на початку другої епохи.

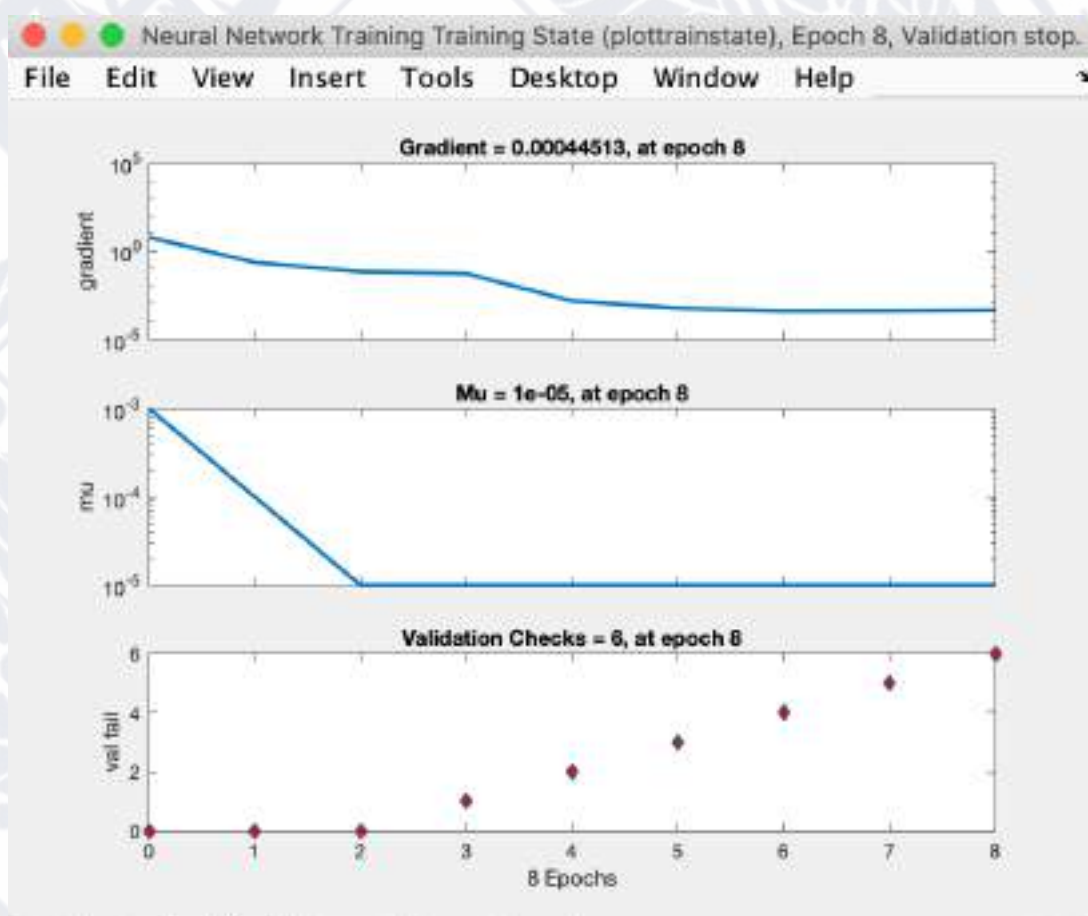


Рис 3.17 Графік навчання нейронної мережі

На рисунку 3.17 представлено як змінювались параметри простої рекурентної нейронної мережі під час навчання.

1. Градієнт (gradient) це вектор, що вказує напрямок максимального зростання функції втрат. Його значення показує швидкість зміни ваг нейронної мережі за час навчання. На даному графіку видно, що швидкість зміни ваг спочатку була висока, а згодом зменшувалась.

Зменшення градієнта свідчить про те, що функція втрат даної моделі наближається до свого мінімального значення.

2. μ . Параметр алгоритму Левенберга-Марквардта, що впливає на швидкість навчання та стабільність алгоритму. На графіку видно, що даних параметр швидко зменшується і стабілізується на певному значення. Це свідчить про те, що значення параметру підібрано так, щоб забезпечити оптимальну швидкість навчання та стабільність алгоритму.
3. Кількість перевірок валідації (val fail). Даний графік показує, що валідація проводилась регулярно протягом усього процесу навчання, що у свою чергу дозволило контролювати якість моделі та вчасно зупинити навчання.

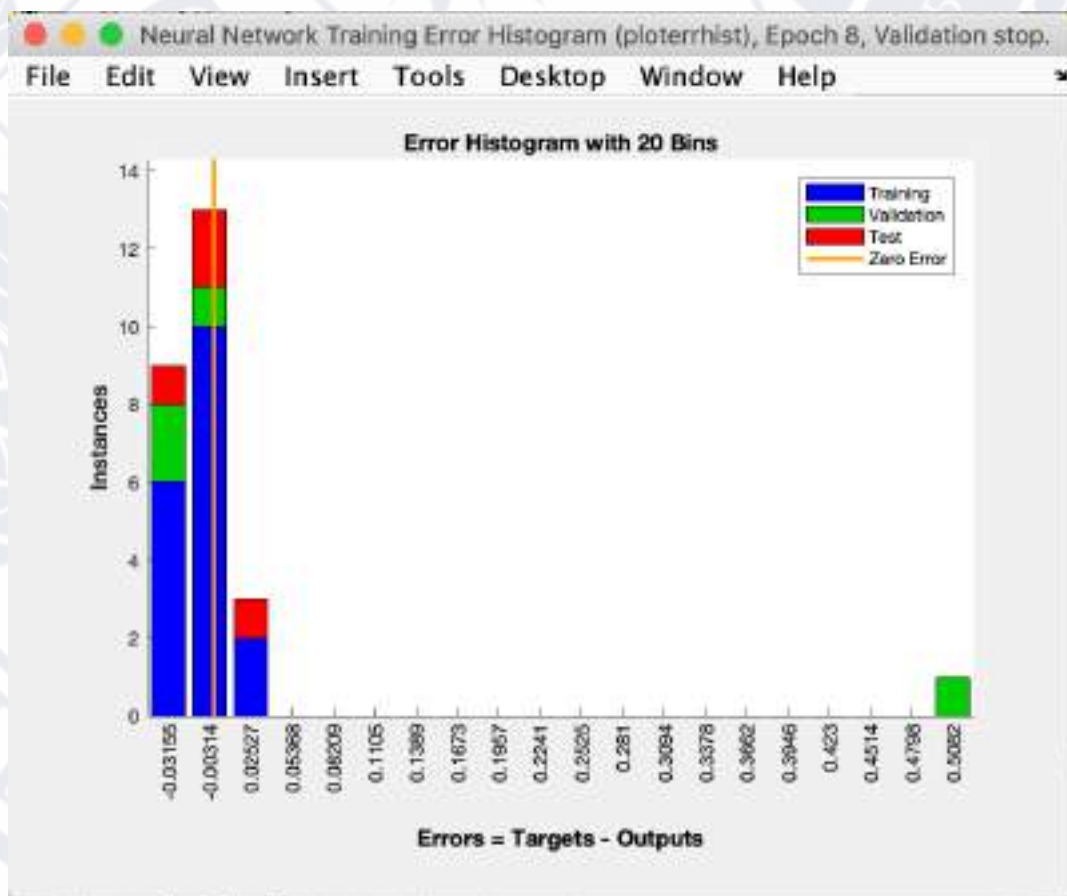


Рис 3.18 Гістограма помилок нейронної мережі

На рисунку 3.18 представлено гістограму помилок, яка дає візуальне уявлення про розподіл помилок, зроблених нейронною мережею під час

навчання та тестування. Із гістограми видно, що більшість помилок зосереджені біля нуля, що свідчить про те, що модель працює добре. Щоправда, розподіли помилок для навчального, валідаційного та тестового наборів даних відрізняються. Це означає те, що модель чудово підлаштовується під набори даних для тренування та гірше для наборів нових даних.

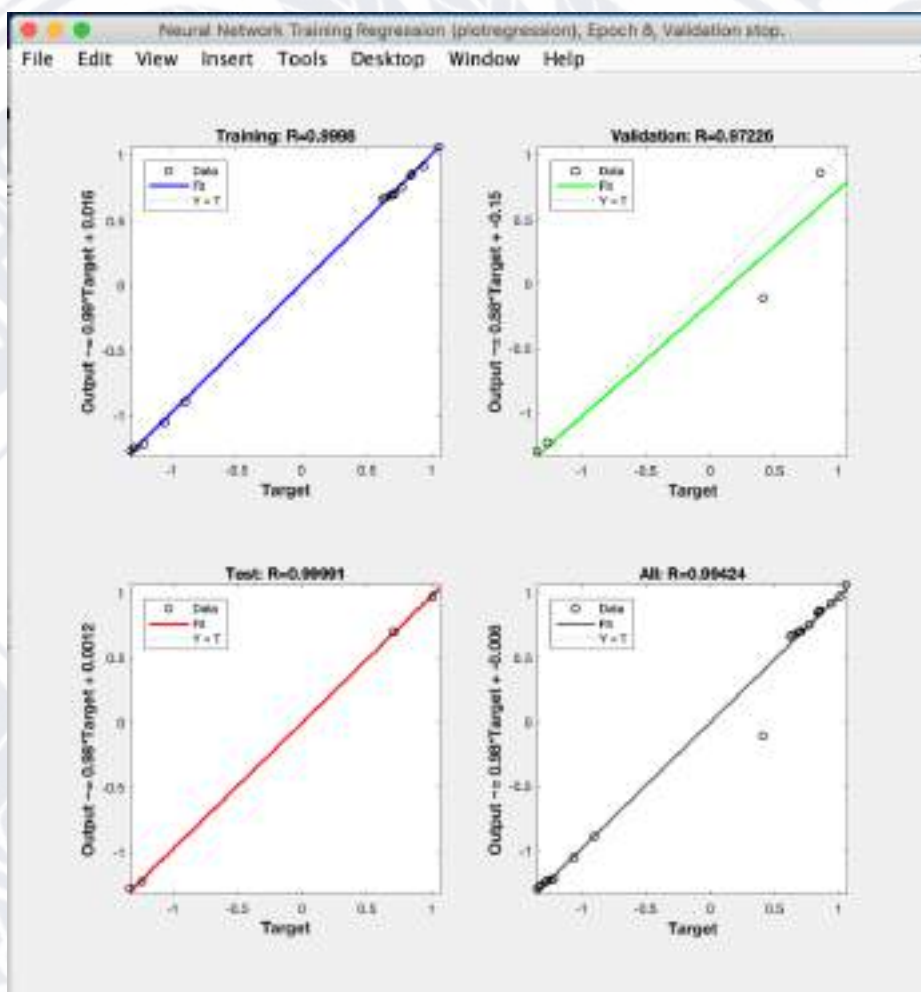


Рис 3.19 Графік регресії нейронної мережі

На рисунку 3.19 представлено графіки регресії, що показуються як добре проста рекурентна нейронна мережа навчилась прогнозувати цільові значення (target) на основі вхідних даних. На кожному графіку представлені відповідні значення коефіцієнту регресії, який є близьким до 1, що у свою чергу демонструє високу якість виявлення залежності моделі між вхідними даними та цільовою змінною.

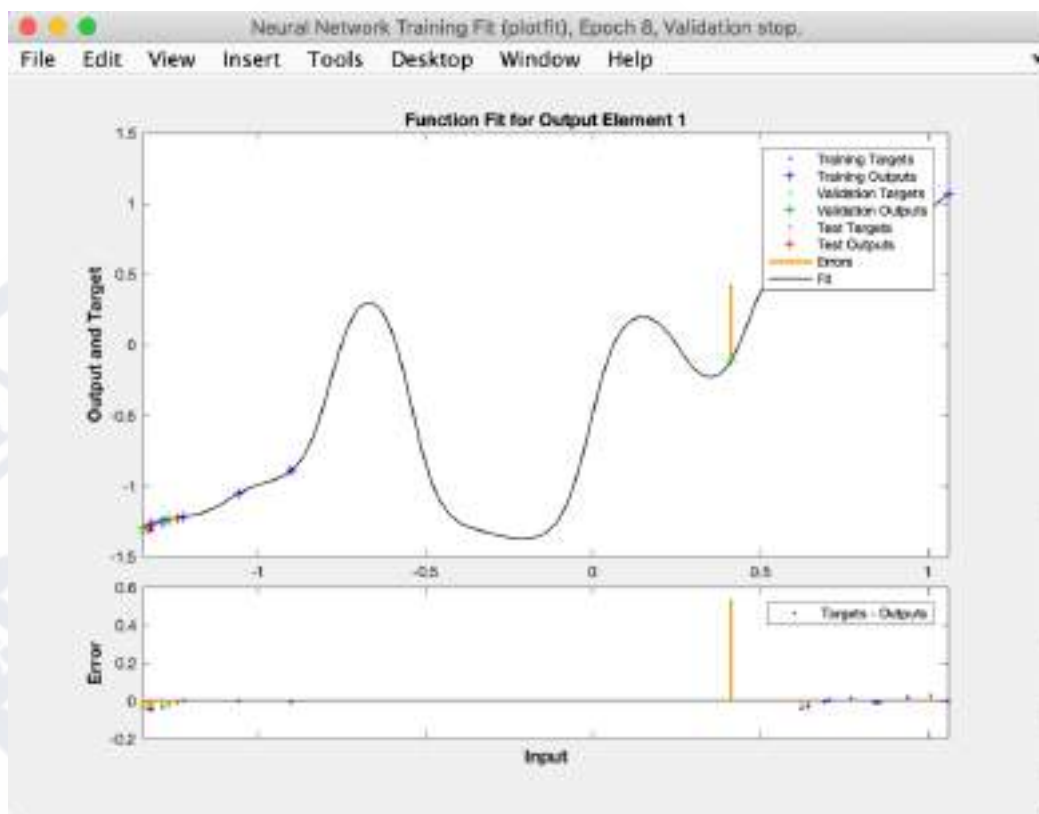


Рис 3.20 Графік відповідності функції нейронної мережі

На рисунку 3.20 представлено графік відповідності функцій нейронної мережі з якого видно, що проста рекурентна нейронна мережа навчилась добре апроксимувати цільову функцію. Щоправда із графіку видно, що кількість помилок у тестовому наборі трохи більші ніж у навчальному та валідаційному наборах, що свідчить про перенавчання мережі.

Четвертим етапом є моделювання мережі, тобто отримання графіків та прогнозування створеної моделі.

```

Command Window
best_tper: 7.8720e-04

>> % Спрогнозуємо результати моделі
>> forecast = net(test_data_norm')

forecast =

    -0.8809    -1.0204     1.2754     0.9701     0.2648    -0.7841
  
```

Рис 3.21 Прогнозування нормалізованих результатів моделі

```

>> %Денормалізуємо результати прогнозування
>> forecast_norm = forecast+(mat(train_date)-min(train_date))*min(train_date)
Undefined function or variable 'mat'.

Did you mean:
>> forecast_norm = forecast+(max(train_date)-min(train_date))*min(train_date)

forecast_norm =

1.0e+04 *

    2.2699    2.2175    3.8800    2.9653    2.7883    2.3862

```

Рис 3.22 Прогнозування де-нормалізованих результатів моделі

```

>> % Візуалізуємо отримані результати
>> figure
>> plot(test_date, 'b', 'LineWidth', 1.5)
>> hold on
>> plot(forecast_norm, 'r', 'LineWidth', 1.5)
>> legend('Тестові дані', 'Прогноз')
>> title('Прогнозування цін на криптовалюту')
xlabel('Номер прикладу')
ylabel('Ціна (USD)')
>> grid on

```

Рис 3.23 Вивід отриманих даних у графік

У даному випадку нам прийшлося виконати де-нормалізацію отриманих результатів, так як на другому етапі для продуктивнішого навчання ми виконали нормалізацію даних. До прикладу, якщо би ми не виконали де-нормалізацію, то прогноз у нас був або нульовим, або із від'ємним знаком попереду.

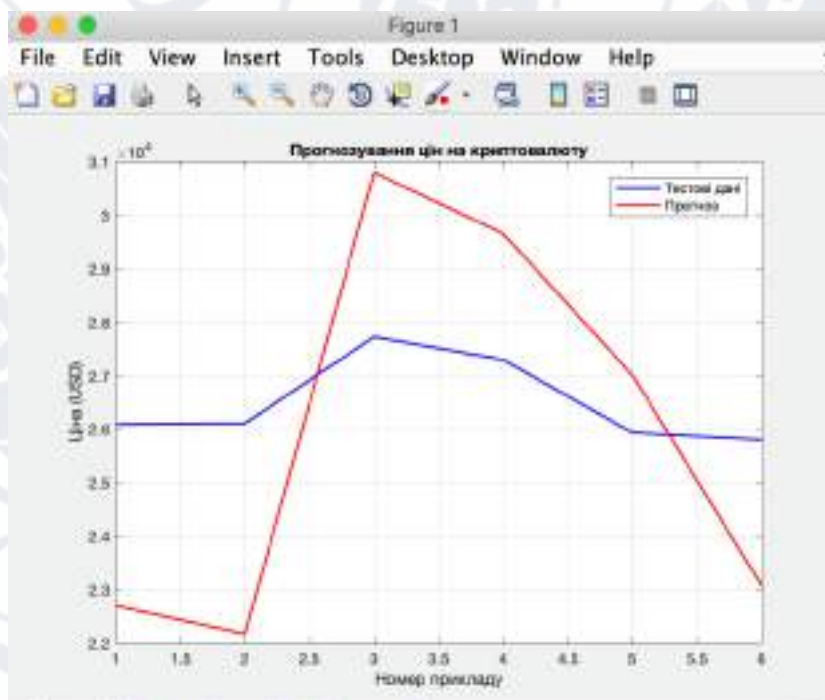


Рис 3.24 Графік прогнозування цін на криптовалюту за допомогою простих рекурентних нейронних мереж

Із рисунку 3.24 видно, видно що представлена модель відображає загальну тенденцію зростання цін на криптовалюту в цілому. Що правда із графіку видно, що прогнози мають значні відхилення від реальних значень. Прогноз вийшов досить нестабільним, відбувалися різкі стрибки, що може свідчити про низьку якість моделі або про те, що дані можуть бути надмірно складними.

3.3 Висновок до розділу 3

У даному розділі ми працювали із простими рекурентними нейронними мережами (SRN). Спробували змодельовати мережу, що допоможе отримувати дані про ціни на цифрові активи на основі історичних даних отриманих із відкритих джерел. Загалом використали чотири етапи у роботі:

1. Завантажили та відфільтрували потрібні нам дані у програмному середовищі matlab.
2. Розбили набір даних на навчальний та тестовий набори.
3. Створили просту рекурентну нейронну мережу на основі усіх заданих раніше налаштувань та параметрів.
4. Прогнозували майбутні дані на основі тестового набору та створили графік із прогнозуванням цін на криптовалюту.

ВИСНОВОК

У даній роботі проведено аналіз відомих моделей машинного навчання для контролю за обігом цифрових фінансових активів та передбачення цін на криптовалюту.

Для вибору оптимальної моделі нейронної мережі для розробки методики контролю було проаналізовано нейронні мережі та створено порівняльні таблиці для вибору найоптимальнішої нейронної мережі.

Розроблено методику контролю обігу цифрових фінансових активів на основі криптовалютних транзакцій за допомогою простих рекурентних нейронних мереж. Дана методика у подальшому може використовуватись як навчальна для студентів вищих навчальних закладів для наочного прикладу роботи рекурентних нейронних мереж щодо контролю за обігом цифрових фінансових активів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. P.1-9
2. V. Buterin. Ethereum: Platform Review Opportunities and Challenges for Private and Consortium Blockchains. 2016.
3. Qichen Huang. Ethereum: Introduction, Expectation, and Implementation. Highlights in Science, Engineering and Technology. 2023. No. 41. P. 175–182.
4. DR Gavin WOOD. Ethereum: a secure decentralised generalised transaction ledger EIP-150 revision. P. 1–32.
5. Michael Devetsikiotis K. C. Blockchain and Smart Contracts for the Internet of Things. Special section on the plethora of research in internet of things(IoT)). 2016. No. 4. P. 2292–2303.
6. Ahmed M. Khedr, Ifra Arif, Pravija Raj P.V., Magdi El-Bannany, Saadat M/Alhashmi, Mennu Sreedharan. Cryptocurrency price prtdiction using traditional statistical amd machine-learning techniques: : A survey. Intelligent Systems in Accounting, Finance and Management, Volume 28, Issue 1. March 2021. P.3-34
7. Sai Saketh Y H. K. R., Venkata Tej Vaibhav. Cryptocurrency Mining - Transition to Cloud. International Journal of Advanced Computer Science and Applications. 2015. Vol. 6, no. 9. 115-124.
8. Дмитренко Т.Л., Любіч О.О., Пархоменко Ю. Регулювання ринку віртуальних активів: глобальний та національний рівні імплементації міжнародних стандартів ПВК/ФТ. Економічна безпека: теорія, практика, управління. Україна, 2021. 31-52 с.
9. Рекомендації FATF. Міжнародні стандарти бороть з відмиванням коштів, фінансуванням тероризму і розповсюдженням зброї масового знищення. Страсбург, 2018. 266 с.
10. Про віртуальні активи : Закон України від 17.02.2022 р. № 2074-IX. URL: <https://zakon.rada.gov.ua/laws/show/2074-20#Text> (дата звернення: 10.12.2023).

11. Мазур Ю.О., Загоруйко Л.В. Огляд сучасних моделей машинного навчання для аналізу криптовалютних транзакцій. II Міжнародна науково-практична конференція «Прикладні аспекти сучасних міждисциплінарних досліджень», ДонНУ імені Василя Стуса, м.Вінниця. 223-226 с.
12. Kotu, V.; Deshpande, B. Chapter 2: Data Mining Process. In Predictive Analytics and Data Mining; Elsevier: Amsterdam, The Netherlands, 2015; P. 26.
13. Jang, H.; Lee, J. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access 2017. P. 5427–5437.
14. Mittal, R.; Arora, S.; Bhatia, M. Automated cryptocurrencies prices prediction using machine learning. Div. Comput. Eng. Netaji Subhas Inst. Technol. India 2018. P. 2229–6956.
15. Poongodi, M.; Sharma, A.; Vijayakumar, V.; Bhardwaj, V.; Sharma, A.P.; Iqbal, R.; Kumar, R. Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. Comput. Electr. Eng. 2020
16. Lamon, C.; Nielsen, E.; Redondo, E. Cryptocurrency price prediction using news and social media sentiment. SMU Data Sci. Rev. 2017. P. 1–22.
17. Kim, Y.B.; Kim, J.G.; Kim, W.; Im, J.H.; Kim, T.H.; Kang, S.J.; Kim, C.H. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. PLoS ONE 2016.
18. Pang, X.; Zhou, Y.; Wang, P.; Lin, W.; Chang, V. An innovative neural network approach for stock market prediction. J. Supercomput. 2020. P. 2098–2118.
19. Bitcoin Price Forecasting: An Integrated Approach Using Hybrid LSTM-ELM Models / c. Luo et al. Mathematical Problems in Engineering. 2022. P. 17.
20. M. R. Islam, M. Bashed-Al-Mahfuz, S. Ahmad, and M. K. I. Molla, “Multiband prediction model for financial time series with multivariate empirical mode decomposition,” Discrete Dynamics in Nature and Society. 2012, pages, 2012.
21. Y. M. Huang, X. Y. Dai, Q. W. Wang, and D. Q. Zhou, “A hybrid model for carbon price forecasting using GARCH and long short-term memory network,” Applied Energy, vol. 285, № 5. 2021.

22. K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.
23. M. Minsky and S. Papert, "Perceptrons." 1969.
24. Poongodi M., Vijayakumar V., Naveen Chalamkurti. Bitcoin Price Prediction using ARIMA Model. International Journal of Web Based Communities. 201X. P. 1–10.
25. Alex Greaves. Benjamin au using the bitcoin transaction graph to predict the price of bitcoin.
26. Evita Stenqvist, Jacob Lnn Predicting Bitcoin price fluctuation with Twitter sentiment analysis
27. SpaCy Industrial-strength Natural Language Processing in Python. SpaCy Industrial-strength Natural Language Processing in Python. URL: <https://spacy.io/> (дата звернення: 16.12.2023).
28. A study on Cryptocurrency Log-Return Price Prediction Using Multivariate Time-Series Model / Sang-Ha Sung et al. axioms. 2022. P. 1–17.
29. Hsu MW, Lessmann S, Sung MC. Bridging the divide in financial market forecasting: machine learners vs. financial economists. Expert Syst Appl 2016. P. 215–234
30. Chen E-H, QiuS-Y, XuCetal. Word embedding —continuous space representation of natural language. 2014. P. 19–29
31. Jaiswal S. Multilayer Perceptrons in Machine Learning: A Comprehensive Guide. Learn Data Science and AI Online. DataCamp. URL <http://surl.li/sypga>. (дата звернення :22.04.2024.)
32. M. Popescu, V. Balas, L. Perescu-Popescu, N. Mastorakis. Multilayer Perceptron and Neural Networks. WSEAS TRANSACTIONS on CIRCUITS and SYSTEM. 2009. No. 8. P. 579–588.
33. Abdel-aziem A. H. A., Soliman T. H. M. A Multi-Layer Perceptron (MLP) Neural Networks for Stellar Classification: A Review of Methods and Results. International Journal of Advances in Applied Computational Intelligence. 2023. Vol. 3, no. 2. P. 29–37.

34. What are Convolutional Neural Networks? IBM. IBM in Deutschland, Österreich und der Schweiz. URL: <http://surl.li/syqgqx>. (дата звернення: 23.04.2024.)
35. A Comprehensive Study on Skin Cancer Detection using Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). A. Shah et al. Clinical eHealth. 2023.
36. A robust deformed convolutional neural network (CNN) for image denoising .Q. Zhang et al. CAAI Transactions on Intelligence Technology. 2022.
37. Das, S., Tariq, A., Santos, T., Kantareddy, S.S., Banerjee, I. Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research. In: Colliot, O. (eds) Machine Learning for Brain Disorders. Neuromethods, vol 197. Humana, New York, NY. 2023
38. Medsker, L., & Jain, L.C. (Eds.. Recurrent Neural Networks: Design and Applications (1st ed.). CRC Press. 1999
39. A Brief Overview of Recurrent Neural Networks (RNN). Analytics Vidhya. URL: <http://surl.li/syqqc>. (дата звернення: 23.04.2024.)
40. A Complete Guide to Recurrent Neural Networks (RNNs). Built In. URL: <http://surl.li/syqru>. (дата звернення: 24.04.2024.)
41. Seabe P. L., Moutsinga C. R. B., Pindza E. Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. Fractal and Fractional. 2023. Vol. 7, no. 2. P. 203.
42. Liang X. Convolutional and recurrent neural networks. Probabilistic Graphical Models. 2017. P. 1–7.
43. Gers F. Long Short-Term Memory in Recurrent Neural Networks. 2001. P. 103.
44. Determination of Deep Learning Model and Optimum Length of Training Data in the River with Large Fluctuations in Flow Rates. K. Park та ін. *Water*. 2020. Т. 12, № 12. С. 3537. (дата звернення: 07.10.2024)
45. Shewalkar A., Nyavanandi D., Ludwig S. A. Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU. Journal of Artificial Intelligence and Soft Computing Research. 2019. Vol. 9, no. 4. P. 235–245.

46. Das, S., Tariq, A., Santos, T., Kantareddy, S.S., Banerjee, I. Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research. In: Colliot, O. (eds) Machine Learning for Brain Disorders. Neuromethods, vol 197. Humana, New York, NY.
47. Мазур Ю.О., Загоруйко Л.В. Розробка методики контролю за обігом цифрових фінансових активів на основі криптовалютних транзакцій. Вісник студентського наукового товариства Донецького національного університету імені Василя Стуса. Випуск 15, том 2. Вінниця 2023. 214-217 с.
48. MATLAB. URL: <https://www.mathworks.com/products/matlab.html> (дата звернення 03.09.2024)

ДОДАТКИ

ДОДАТОК А Розробка схеми архітектури нейронної мережі прямого поширення у matlab

ДОДАТОК Б Розробка схеми архітектури згорткової нейронної мережі у matlab

ДОДАТОК В Розробка схеми архітектури рекурентної нейронної мережі у python

ДОДАТОК Г Таблиця 2.1 Переваги та недоліки нейронних мереж

ДОДАТОК Ґ Розробка схеми архітектури типової моделі LSTM у python

ДОДАТОК Д Розробка схеми архітектури схеми GRU у python

ДОДАТОК Е Розробка схеми архітектури простої рекурентної нейронної мережі у matlab

ДОДАТОК Є Код представлення методології за контролем обігу цифрових активів

ДОДАТОК А

Розробка схеми архітектури нейронної мережі прямого поширення у **matlab**

% Створення нейронної мережі прямого поширення

```
>> net = feedforwardnet(1)
```

% Вивід структурної схеми мережі

```
>> view(net);
```

ДОДАТОК Б

Розробка схеми архітектури згорткової нейронної мережі у matlab

% Створення згорткової нейронної мережі

```
>> layers = [  
    imageInputLayer([28 28 1], 'Name', 'input')  
    convolution2dLayer(5, 20, 'Name', 'conv')  
    reluLayer('Name', 'relu')  
    maxPooling2dLayer(2, 'Stride', 2, 'Name', 'maxpool')  
    fullyConnectedLayer(10, 'Name', 'fc')  
    softmaxLayer('Name', 'softmax')  
    classificationLayer('Name', 'output')]
```

% Візуалізація структури мережі

```
>> plot(layerGraph(layers))
```

ДОДАТОК В

Розробка схеми архітектури рекурентної нейронної мережі у python

```

import networkx as nx
import matplotlib.pyplot as plt
# Створення графу
G = nx.DiGraph()
# Додавання вершин
layers = ['Input', 'Hidden', 'Output']
num_nodes = [2, 3, 1] # Кількість вузлів у кожному прошаркові
for i, layer_name in enumerate(layers):
    for j in range(num_nodes[i]):
        node_name = f'{layer_name}({j})'
        G.add_node(node_name, pos=(i, j))
# Додавання зв'язків
for i in range(len(layers) - 1):
    current_layer_nodes = [f'{layers[i]}({j})' for j in range(num_nodes[i])]
    next_layer_nodes = [f'{layers[i+1]}({j})' for j in range(num_nodes[i+1])]
    for node in current_layer_nodes:
        for next_node in next_layer_nodes:
            G.add_edge(node, next_node)
# Отримання позицій вершин
pos = nx.get_node_attributes(G, 'pos')
# Відображення графу
nx.draw(G, pos, with_labels=True, node_size=2000, node_color='skyblue',
        font_size=10, font_weight='bold')
plt.show()

```

ДОДАТОК Г

Таблиця 2.1 Переваги та недоліки нейронних мереж

Нейронні мережі прямого поширення	
Переваги	Недоліки
○ MLP легше зрозуміти, ніж більш складні моделі глибокого навчання, завдяки їхній простій архітектурі та принципам роботи ○ Кількість прошарків і нейронів, функції активації та алгоритми навчання можна налаштовувати для досягнення оптимальних результатів під конкретну задачу. ○ Розуміння того, як працює MLP є ключом для розуміння та розробки більш складних моделей.	○ Алгоритми навчання, що використовуються в MLP можуть не справлятися із обробкою великих обсягів даних, що у свою чергу призводить до повільного навчання та низької точності ○ Навчання MLP займає багато часу і вимагає значних обчислювальних ресурсів, особливо при роботі з великими масивами даних ○ Відсутність чіткого уявлення про внутрішні процеси MLP може ускладнити аналіз їхньої діяльності та виявлення можливих помилок і упереджень.
Згорткові нейронні мережі	
Переваги	Недоліки
○ CNN можуть автоматично вивчати відповідні функції з необроблених вхідних даних, усуваючи необхідність ручної розробки функцій.	○ CNN потребують великих обчислювальних ресурсів, включаючи високопродуктивні графічні процесори для навчання та розгортання.

Продовження таблиці 1.1

- о CNN можуть досягати найсучаснішої продуктивності в різноманітних завданнях розпізнавання зображення та відео. - о CNN мають високу стійкість до шуму і спотворень вхідних даних, що робить їх високоефективними в реальних умовах	- о Для досягнення високої точності CNN потребують великих обсягів маркованих навчальних даних, які може бути важко отримати в певних областях науки. - о CNN схильні до перенавчання, коли навчальні дані обмежені або зашумлені.
Рекурентні нейронні мережі	
Переваги	Недоліки
- о RNN може запам'ятовувати попередні входи і виходи. Підвищення запам'ятовування відбувається за рахунок мережі LSTM - о RNN може вчитися на минулому досвіді, на основі цього вона може робити точні прогнози - о RNN ідеально підходить для обробки послідовних даних, оскільки вони розуміють часові аспекти даних. - о RNN мають широкий спектр застосувань, оскільки вони можуть обробляти послідовні дані, такі як часові ряди, аудіо та мовлення.	- о Алгоритм присвоює вагам необґрунтовано високі значення без жодної конкретної причини - о Коли значення градієнта занадто мале, модель перестає навчатися або навчається занадто довго - о RNN обробляє дані послідовно, що може зробити процес навчання нудним - о Чим довша послідовність даних, тим складніше для RNN запам'ятовувати попередню інформацію - о RNN обробляє дані послідовно, що може бути трудомістким і неефективним методом

ДОДАТОК Г

Розробка схеми архітектури типової моделі LSTM у python

```

import matplotlib.pyplot as plt
import networkx as nx
import numpy as np

# Оголошення графа LSTM-комірки
G = nx.DiGraph()

# Додавання вузлів
G.add_node('x', pos=(0, 0))
G.add_node('f', pos=(1, 0))
G.add_node('i', pos=(2, 0))
G.add_node('c', pos=(3, 0))
G.add_node('o', pos=(4, 0))
G.add_node('h', pos=(5, 0))

# Додавання ребер
G.add_edge('x', 'i', color='gray', arrowstyle='<-')
G.add_edge('x', 'f', color='gray', arrowstyle='<-')
G.add_edge('h', 'f', color='gray', arrowstyle='<-')
G.add_edge('x', 'c', color='gray', arrowstyle='<-')
G.add_edge('h', 'c', color='gray', arrowstyle='<-')
G.add_edge('f', 'c', color='gray', arrowstyle='->')
G.add_edge('i', 'c', color='gray', arrowstyle='->')
G.add_edge('c', 'o', color='gray', arrowstyle='->')
G.add_edge('o', 'h', color='gray', arrowstyle='->')

# Функція LSTM-комірки
def lstm_cell(x, h_prev, c_prev, w_i, w_f, w_c, w_o, b_i, b_f, b_c, b_o):
    i = sigmoid(np.matmul(x, w_i) + np.matmul(h_prev, w_f) + b_i)
    f = sigmoid(np.matmul(x, w_f) + np.matmul(h_prev, w_f) + b_f)
    c_new = f * c_prev + i * tanh(np.matmul(x, w_c) + b_c)
    o = sigmoid(np.matmul(x, w_o) + np.matmul(h_prev, w_o) + b_o)

```

```

h_new = o * tanh(c_new)

return h_new, c_new

# Активаційна функція сигмоїда
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Активаційна функція tangens хиперbolicus
def tanh(x):
    return (np.exp(x) - np.exp(-x)) / (np.exp(x) + np.exp(-x))

# Оформлення LSTM-комірки
fig, ax = plt.subplots()
pos = nx.get_node_attributes(G, 'pos')
nx.draw_networkx_nodes(G, pos, ax=ax)
nx.draw_networkx_edges(G, pos, ax=ax)
nx.draw_networkx_labels(G, pos, ax=ax)

# Додавання міток для вхідного та вихідного вузлів
x_label = ax.text(pos['x'][0], pos['x'][1], 'x', ha='center', va='center')
h_label = ax.text(pos['h'][0], pos['h'][1], 'h', ha='center', va='center')

# Додавання міток для ребер
for edge in G.edges():
    nx.draw_networkx_edge_labels(G, pos, edge_labels={edge: ""}, ax=ax)

# Встановлення меж осі
ax.set_xlim([-0.5, 5.5])
ax.set_ylim([-0.5, 1.5])
plt.show()

```

ДОДАТОК Д

Розробка схеми архітектури схеми GRU у python

```

import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
# Створення графа для візуалізації GRU
G = nx.DiGraph()
# Додавання вузлів
G.add_node('x', pos=(0, 0)) # 'x' - вхідний вектор
G.add_node('z', pos=(1, 0)) # 'z' - перезаписувальна брама (update gate)
G.add_node('r', pos=(2, 0)) # 'r' - ресетувальна брама (reset gate)
G.add_node('h', pos=(3, 0)) # 'h' - вихідний вектор (hidden state)
# Додавання ребер, що з'єднують вузли
G.add_edge('x', 'z', color='gray', arrowstyle='<-') # Зв'язок від входу до
перезаписувальної брами
G.add_edge('x', 'r', color='gray', arrowstyle='<-') # Зв'язок від входу до
ресетувальної брами
G.add_edge('h', 'z', color='gray', arrowstyle='<-') # Зв'язок від попереднього виходу
до перезаписувальної брами
G.add_edge('h', 'r', color='gray', arrowstyle='<-') # Зв'язок від попереднього виходу
до ресетувальної брами
G.add_edge('z', 'h', color='gray', arrowstyle='->') # Зв'язок від перезаписувальної
брами до виходу
G.add_edge('r', 'h', color='gray', arrowstyle='->') # Зв'язок від ресетувальної брами
до виходу
# Функція GRU клітини (логіка не реалізована, тільки сигнатура)
def gru_cell(x, h_prev, w_z, w_r, w_h, b_z, b_r, b_h):
    pass # Тула для функції GRU клітини
# Функція активації sigmoid
def sigmoid(x):

```

```
return 1 / (1 + np.exp(-x))

# Візуалізація графа GRU клітини
fig, ax = plt.subplots()
pos = nx.get_node_attributes(G, 'pos')
nx.draw_networkx_nodes(G, pos, ax=ax)
nx.draw_networkx_edges(G, pos, ax=ax)
nx.draw_networkx_labels(G, pos, ax=ax)
# Додавання міток для вузлів входу та виходу
x_label = ax.text(pos['x'][0], pos['x'][1], 'x', ha='center', va='center')
h_label = ax.text(pos['h'][0], pos['h'][1], 'h', ha='center', va='center')
# Налаштування відображення
ax.set_xlim([-0.5, 3.5])
ax.set_ylim([-0.5, 1.5])
ax.set_title('GRU клітина')
plt.show()
```

ДОДАТОК Е

Розробка схеми архітектури простої рекурентної нейронної мережі у **matlab**

% Створення простої рекурентної нейронної мережі (SRNN)

>> input = 1:2 %створюємо затримку на вході

>> hidden = 10 %вказуємо кількість нейронів у прихованому прошарку

>> net = layrecnet(input, hidden) %створення простої рекурентної нейронної мережі

>> view(net) % виведення схеми SRN

ДОДАТОК Є

Код представлення методології за контролем обігу цифрових активів

```
% Завантаження даних
>> df = readtable('/Users/mac/Desktop/bitcoin_prices(USD).csv');
%Вкажемо назви стовпців з якими будемо працювати (у нашому випадку це
стовпець data та closing prises)
>> data = df.Date;
>> price = df.ClosingPrice;
% Конвертуємо дані для належної роботи у подальшому
>> data = datetime(data, 'InputFormat', 'dd.MM.yyyy');
% Так як ми плануємо працювати з даними тільки за один місяць, то вкажемо
відрізок дат
>> start_date = datetime('01.08.2023','InputFormat','dd.MM.yyyy');
>> end_date = datetime('01.09.2023','InputFormat','dd.MM.yyyy');
% Відфільтруємо наші дати, аби уникнути помилок із пропущеними значеннями,
тощо
>> filter_date = df(data >= start_date & data <= end_date,:);
>> filter_price = filter_date.ClosingPrice;
>> %Обчислимо загальну кількість рядків у filter_price
>> point_date = size(filter_price, 1)
%Визначимо розмір навчального набору, так як у нас це 80%, вказуємо 0.8
>> train_size = round(point_date*0.8)
>> %Вкажемо що отримані 26 рядків будуть у навчальному наборі даних
>> train_date = filter_price(1:train_size)
>> %Вкажемо, що інші 6 будуть використовуватись для тестування
>> test_date = filter_price(train_size+1:end)
%Нормалізуємо наші дані поєднаним методом логарифмічної трансформації та
стандартизації, що може забезпечити більш стабільну нормалізацію у випадку із
коливаннями цін
```

```

>> train_date_log = log(train_date)
>> test_date_log = log(test_date)
>> train_data_norm = (train_date_log - mean(train_date_log))/std(train_date_log)
>> test_data_norm = (test_date_log - mean(test_date_log))/std(test_date_log)
%Створюємо просту рекурентну нейронну мережу (SRN)
% Задамо кількість нейронів
>> hidden_layer = 10
>> net = fitnet(hidden_layer)
% Вказуємо параметри мережі
>> net.trainParam.epochs = 50
>> net.trainParam.goal = 1e-5
% Вважаючи на задані параметри та дані навчаємо просту рекурентну нейронну
мережу
>> [net, train] = train(net, train_data_norm',train_data_norm')
% Спрогнозуємо результати моделі
>> forecast = net(test_data_norm')
%Денормалізуємо результати прогнозування
>> forecast_norm = forecast*(max(train_date)-min(train_date))+min(train_date)
% Візуалізуємо отримані результати
>> figure
>> plot(test_date, 'b', 'LineWidth', 1.5)
>> hold on
>> plot(forecast_norm,'r', 'LineWidth', 1.5)
>> legend('Тестові дані', 'Прогноз')
>> title('Прогнозування цін на криптовалюту')
>> xlabel('Номер прикладу')
>> ylabel('Ціна (USD)')
>> grid on

```