

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

(повна назва)

Кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Дослідження алгоритмічних трейдингових стратегій у мережах блокчейн.

(тема)

Виконав:

здобувач 2 курсу, групи ІТІм-24-1

Столбецький Юрій Олександрович

(прізвище, власне ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології

проекткування

(повна назва освітньої програми)

Керівник проф. Калита Н.І.

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри \_\_\_\_\_

(підпис)

Гребеннік І.В.

(власне ім'я, прізвище)

2025 р.

*Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.*

22.12.2025




*Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.*

Керівник кваліфікаційної роботи  проф. Калита Н.І

*Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.*

Керівник кваліфікаційної роботи  проф. Калита Н.І.

*Попередній захист проведено 22 грудня 2025 р.*

Керівник кваліфікаційної роботи  проф. Калита Н.І.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування

(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«20» грудня 2025 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Столбецькому Юрію Олександровичу

(прізвище, власне ім'я, по батькові)


1. Тема роботи «Дслідження алгоритмічних трейдингових стратегій у мережах блокчейн. затверджена наказом університету від «24» листопада 2025р. № 1058 Ст
2. Термін подання студентом роботи до екзаменаційної комісії «20» грудня 2025 р.
3. Вихідні дані до роботи алгоритмічні трейдингові стратегії у мережах блокчейн; вхідні дані: історичні часові ряди цін та обсягів криптоактивів, ончейн-дані про стани смартконтрактів і пулів ліквідності, котирування з API бірж, ПК 16 ГБ ОП, ОС Windows 10 / Linux, мова програмування Rust, асинхронний фреймворк Tokio, системи керування даними Redis і PostgreSQL, інструменти Solana SDK та інтеграція з зовнішніми API (Jupiter, Bybit).
4. Перелік питань, що потрібно опрацювати в роботі 1. Вступ; 2. Аналіз предметної області; 3 Дослідження існуючих підходів та методів для вирішення подібних задач; 4 Постановка задачі дослідження;; 5.Розробка інформаційної системи; результатів; 6 Порівняльний аналіз моделей; 6. Висновки.
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Нейромережева оптимізація портфеля на леверидж-токенах (BNBUP/BNBDOWN), Глибинне навчання з


підкріпленням (ансамбль PPO+LSTM), Мультивалютний статистичний арбітраж, Розроблена коінтеграційна стратегія, Логічна модель БД, Порівняння стратегій.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної області	24.11.2025 – 28.11.2025	Виконано
2	Дослідження існуючих підходів та методів	29.11.2025 – 6.12.2025	Виконано
3	Розробка інформаційної системи	7.12.2025 – 14.12.2025	Виконано
4	Проведення обчислювальних експериментів	15.12.2025 – 18.12.2025	Виконано
5	Оформлення пояснювальної записки до кваліфікаційної роботи та презентації	19.12.2025 – 21.12.2025	Виконано
6	Попередній захист	22.12.25	Виконано
7	Подання кваліфікаційної роботи до екзаменаційної комісії	23.12.2025	Виконано

Дата видачі завдання « 24 » листопада 2025 р.

Здобувач   
(підпис)

Керівник роботи   
(підпис)

проф. Калита Н.І.  
(посада, власне ім'я, прізвище)

## РЕФЕРАТ

Кваліфікаційна робота містить: 97 с., 5 рис., 1 табл., 25 джерел, 2 додатки.

АЛГОРИТМІЧНА ТОРГІВЛЯ, БЛОКЧЕЙН, СТАТИСТИЧНИЙ АРБИТРАЖ, КОІНТЕГРАЦІЙНА СТРАТЕГІЯ, SOLANA, RUST, DeFi, REDIS.

Об'єктом дослідження є процес алгоритмічної торгівлі криптоактивами в мережах блокчейн, зокрема в середовищі централізованих бірж та DeFi-протоколів.

Предметом дослідження є моделі та методи статистичного арбітражу на основі коінтегрованих спредів, а також програмні засоби для їхнього бектестування в умовах блокчейн-екосистеми.

Метою роботи є розробка та експериментальне дослідження базової коінтеграційної стратегії статистичного арбітражу, адаптованої до умов децентралізованих та централізованих криптовалютних майданчиків.

Методами дослідження є: математичне моделювання, статистичний аналіз часових рядів, елементи теорії коінтеграції, методи оцінювання ризику портфеля, об'єктно-орієнтоване програмування, сучасні технології розробки застосунків.

У роботі розроблено архітектуру та програмну реалізацію системи бектестування мовою Rust із використанням Redis і PostgreSQL, передбачено інтеграцію з біржею Bybit, екосистемою Solana та агрегатором свопів Jupiter.

Проведено порівняльний аналіз отриманих результатів із моделями, описаними в наукових статтях, оцінено місце запропонованої коінтеграційної стратегії серед сучасних підходів.

Отримані результати формують основу для подальшого розвитку більш складних коінтеграційних, портфельних та гібридних моделей, а також для їхньої інтеграції безпосередньо в DeFi-протоколи.

## ABSTRACT

The thesis contains: 97 pages, 5 fig., 1 table, 25 sources, 2 attachments.

ALGORITHMIC TRADING, BLOCKCHAIN, STATISTICAL ARBITRAGE, COINTEGRATION STRATEGY, SOLANA, RUST, DeFi, REDIS.

The object of the study is the process of algorithmic trading of crypto assets in blockchain networks, in particular in the environment of centralized exchanges and DeFi protocols.

The subject of the study is models and methods of statistical arbitrage based on cointegrated spreads, as well as software tools for their backtesting in the blockchain ecosystem.

The goal of the work is to develop and experimentally study a basic cointegration strategy for statistical arbitrage, adapted to the conditions of decentralized and centralized cryptocurrency platforms.

The research methods include mathematical modeling, statistical analysis of time series, elements of cointegration theory, portfolio risk assessment methods, object-oriented programming, and modern application development technologies.

The work develops the architecture and software implementation of the backtesting system in Rust using Redis and PostgreSQL, and provides for integration with the Bybit exchange, the Solana ecosystem, and the Jupiter swap aggregator.

A comparative analysis of the results obtained with the models described in scientific articles was carried out, and the place of the proposed cointegration strategy among modern approaches was assessed.

## ЗМІСТ

Вступ.....	8
1 Аналіз предметної області дослідження.....	10
1.1 Особливості алгоритмічної торгівлі у мережах блокчейн.....	10
1.2 Огляд існуючих підходів до алгоритмічної торгівлі в блокчейн- мережах.....	15
1.3 Формулювання мети та задач дослідження.....	18
2 Дослідження існуючих підходів та методів для вирішення подібних задач...20	
2.1 Загальна характеристика обраних наукових джерел .....	20
2.2 Нейромережева стратегія портфельної оптимізації на основі леверидж-токенів.....	21
2.3 Ансамблевий підхід на основі глибокого навчання з підкріпленням для керування криптовалютичним портфелем.....	25
2.4 Оптимізаційний підхід до мультивалютного статистичного арбітражу з використанням криптовалютичного “якоря” .....	28
3 Постановка задачі дослідження.....	32
3.1 Загальна постановка задачі дослідження.....	32
3.2 Вхідні дані, ринкове середовище та основні позначення.....	33
3.3 Математична модель спреда та коінтегрованих портфельів.....	35
3.4 Постановка задачі побудови торгової стратегії на основі коінтегрованого спреда.....	39
3.5 Критерії якості та постановка задачі порівняльного аналізу стратегій.....	43
4 Розробка інформаційної системи.....	47
4.1 Функціональні вимоги та загальна архітектура системи.....	47
4.2 Вибір технологічного стеку та підхід до реалізації.....	49
4.3 Підсистема зберігання та обробки даних.....	52
4.4 Модуль аналізу ринку та генерації сигналів.....	56
4.5 Підсистема взаємодії з зовнішніми сервісами.....	59

4.6 Порівняльний аналіз та валідація результатів.....	64
5 Порівняльний аналіз моделей.....	66
5.1 Кількісне порівняння ефективності.....	66
Висновки.....	69
Перелік джерел посилання.....	72
Додаток А. Графічні матеріали.....	75
Додаток Б. Текст програми.....	87

## ВСТУП

У сучасному світі фінансові ринки дедалі більше переходять у цифровий простір, а поява криптовалют та блокчейн-технологій докорінно змінює уявлення про гроші, інвестиції та інфраструктуру торгівлі. Криптовалютні біржі [1], децентралізовані фінансові протоколи (DeFi) [2] та смартконтракти [3] створюють нові можливості для інвесторів і розробників, але одночасно супроводжуються високою волатильністю (міра того, наскільки сильно й часто змінюється ціна активу за певний проміжок часу), фрагментованістю ліквідності та підвищеними ризиками. У цих умовах зростає роль алгоритмічної торгівлі, яка дає змогу автоматизувати прийняття рішень, оперативно реагувати на зміни ринку та системно оцінювати ризик.

Однією з ключових проблем сучасних криптовалютних ринків є відсутність прозорих та стабільних механізмів ціноутворення, характерних для класичних фондових ринків. Багато стратегій, що успішно працюють у традиційних фінансах, не можуть бути безпосередньо перенесені в блокчейн-середовище через відмінності у ринковій мікроструктурі, комісіях, затримках підтвердження транзакцій та специфіці взаємодії зі смартконтрактами. Це створює потребу в адаптації існуючих алгоритмічних підходів, розробці нових стратегій та побудові спеціалізованих інструментів дослідження й тестування в умовах децентралізованих бірж і блокчейн-мереж.

Тема магістерського дослідження «Дослідження алгоритмічних трейдингових стратегій у мережах блокчейн» є актуальною, оскільки поєднує завдання фінансового аналізу, математичного моделювання та розробки програмного забезпечення. У рамках кваліфікаційної роботи необхідно проаналізувати сучасні підходи до алгоритмічної торгівлі на криптовалютних ринках, дослідити можливості застосування статистичного арбітражу, портфельної оптимізації та методів машинного навчання, а також підготувати

основу для подальшої реалізації та експериментального дослідження обраних стратегій у блокчейн-середовищі.

Об'єктом дослідження є процес алгоритмічної торгівлі криптоактивами в мережах блокчейн, зокрема в середовищі централізованих бірж та DeFi-протоколів.

Предметом дослідження є моделі та методи статистичного арбітражу на основі коінтегрованих спредів, а також програмні засоби для їхнього бектестування в умовах блокчейн-екосистеми.

Метою роботи є розробка та експериментальне дослідження базової коінтеграційної стратегії статистичного арбітражу, адаптованої до умов децентралізованих та централізованих криптовалютних майданчиків.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

### 1.1 Особливості алгоритмічної торгівлі у мережах блокчейн

Алгоритмічна торгівля у мережах блокчейн є поєднанням трьох сфер [4]:

- 1) фінансових технологій;
- 2) розподілених обчислювальних систем;
- 3) інтелектуальних алгоритмів обробки даних.

На відміну від класичних біржових систем, де вся логіка роботи зосереджена на централізованих серверах біржі чи брокера, у блокчейн-середовищі значна частина ринкової інфраструктури реалізується у вигляді смартконтрактів, які виконуються на розподілених вузлах мережі. Це визначає специфіку архітектури алгоритмічних систем, способи доступу до даних, вимоги до затримок, управління ризиками та економіку виконання угод. З погляду інформаційних систем, система алгоритмічної торгівлі в блокчейні є складною інформаційною системою реального часу, що складається з таких основних підсистем [5]:

- підсистема збору даних (market data & on-chain data): отримує ціни, обсяги торгів, глибину ринку, стан пулів ліквідності, історію транзакцій, дані про стани смартконтрактів;
- аналітична підсистема, яка виконує розрахунок індикаторів, статистичних характеристик, побудову спредів, перевірку коінтеграції, розрахунок ризик-метрик тощо;
- підсистема прийняття рішень, де реалізуються правила алгоритмічної стратегії (детерміновані або на основі моделей машинного навчання) та формуються торгові сигнали (купити/продати/перебалансувати портфель);
- підсистема виконання угод, яка відповідає за взаємодію з біржами та протоколами DeFi, формування й підписання транзакцій, вибір типу ордерів (для CEX) або параметрів виклику смартконтрактів (для DEX);

- модуль управління ризиками - контролює обсяги позицій, кредитне плече, ліміти збитку та прибутку, обмеження на використання ліквідності, захист від аномалій;
- підсистема моніторингу та логування, яка відстежує стан системи, помилки;
- відхилення від очікуваної поведінки, збирає дані для подальшого аналізу та бектестингу.

Блокчейн [6] як основа цієї екосистеми є розподіленим реєстром, у якому дані про баланси, угоди та стани смартконтрактів зберігаються децентралізовано й узгоджуються за допомогою протоколів консенсусу. Це означає, що:

- немає єдиного «центрального сервера», від якого залежить доступність системи;
- будь-який вузол може верифікувати коректність історії транзакцій;
- усі ончейн-дані є публічними й доступними для аналізу, що відкриває широкі можливості для побудови статистичних моделей та алгоритмічних стратегій.

Однак розподілена природа блокчейну накладає низку обмежень і особливостей для алгоритмічної торгівлі [7]. По-перше, важливою є проблема затримок і обмеженої пропускну здатності мережі. Транзакція в блокчейні не виконується миттєво, як виклик функції на локальному сервері: спочатку вона потрапляє до мемпулу, конкурує за включення до блоку, потребує оплати комісії (gas), і лише після підтвердження стає частиною спільного реєстру. У результаті між моментом, коли алгоритм приймає рішення, та моментом фактичного виконання угоди виникає часовий розрив, що є критичним для високочастотних стратегій і арбітражних операцій.

Другою суттєвою особливістю є вартість виконання транзакцій. Кожна взаємодія зі смартконтрактом у публічних блокчейн-мережах потребує сплати комісії. Для алгоритмічних стратегій, що передбачають часті ребалансування портфеля або велику кількість арбітражних угод, сумарні витрати на gas [8] можуть стати визначальним фактором та фактично «з'їдати» очікуваний

прибуток. Тому модель комісій має бути інтегрована безпосередньо в логіку стратегії та систему управління ризиками.

Окремої уваги потребує мікроструктура DeFi-ринків [9]. На децентралізованих біржах, де застосовуються автоматизовані маркет-мейкери, ціна формується не через книгу заявок, а за допомогою математичної формули, наприклад,  $x \cdot y = k$ , де  $x$  – кількість першого токена в пулі,  $y$  – кількість другого токена в пулі,  $k$  - стала величина (інваріант пулу), добуток резервів, який АММ намагається зберегти постійним під час свопів. Кожна операція змінює стан пулу ліквідності, а отже й ціну активів. Для алгоритмічної стратегії це означає необхідність врахування прослизання, тобто зміни ціни внаслідок власного обсягу угоди, а також ризику імперманентних втрат у разі надання ліквідності. Якість виконання ордерів стає залежною від глибини конкретного пулу та активності інших учасників ринку.

Ще одним характерним аспектом є наявність ризиків, пов'язаних із максимально видобуваною вартістю (Maximal Extractable Value - MEV) та фронтранінгом [10]. У блокчейн-мережах майнери, валідатори або спеціалізовані боти можуть впливати на порядок розміщення транзакцій у блоці, вставляючи власні операції до або після транзакцій трейдера. Такі дії, як фронтранінг чи сандвіч-атаки, дають змогу вилучати частину потенційного прибутку алгоритмічної стратегії. У відповідь розробник має враховувати додаткові захисні механізми: використовувати приватні RPC-ендпоінти, застосовувати спеціалізовані захищені релеї, обмежувати максимально припустиме прослизання та перевіряти стан пулів безпосередньо перед відправленням транзакції.

Не менш важливою є проблема джерел даних та їхньої узгодженості. У традиційних централізованих біржових системах єдине джерело правди задається самим торговим майданчиком. У DeFi-середовищі картина складніша: ончейн-дані, тобто історія блоків, логів та станів контрактів, забезпечують достовірне, але дещо відкладене відображення ринку; офчейн-дані надходять від індикаторів, постачальників цінових потоків та агрегаторів, що можуть мати

власні затримки та похибки; окрему роль відіграють цінові оракули на кшталт Chainlink чи Pyth, які передають котирування між різними мережами та протоколами. Алгоритмічна система має коректно поєднувати ці джерела, контролювати затримки й можливі розбіжності, адже якість вхідних даних безпосередньо визначає результативність стратегії.

Додатковим викликом є вимоги до безпеки та надійності. Смартконтракти після розгортання зазвичай є незмінними, або їх модифікація можлива лише через спеціально передбачені механізми. Будь-яка помилка в коді торгового контракту, некоректні перевірки вхідних даних чи відсутність обмежень на обсяг операцій можуть призвести до незворотних фінансових втрат. Тому розробка алгоритмічних стратегій у блокчейні потребує використання як формальних, так і неформальних методів валідації смартконтрактів, жорсткого контролю прав доступу, визначення лімітів на максимальну суму коштів, якою оперує стратегія, а також наявності механізмів аварійного відключення або переходу системи у безпечний режим у разі виявлення аномальної поведінки чи збоїв інфраструктури.

З точки зору класифікації інформаційних систем, алгоритмічні трейдингові рішення у блокчейні можна віднести до таких класів:

- фінансові інформаційні системи (забезпечують операції з активами, розрахунок фінансових показників, облік і звітність);
- системи підтримки прийняття рішень (використовують формальні моделі та аналітику для обґрунтування торгових дій);
- розподілені системи (оскільки значна частина логіки виконується на децентралізованих вузлах, а компоненти системи рознесені між ончейн- та офчейн-сервісами);
- системи реального часу (обробляють стрімкі потоки даних і мають реагувати в обмежені часові рамки, хоча ці рамки залежать від параметрів конкретної мережі та стратегії).

Особливу групу становлять маркет-нейтральні алгоритмічні стратегії в блокчейні [11]. Вони спрямовані не на прогноз абсолютного руху окремого

активу «вгору/вниз», а на використання відносних дисбалансів між активами, пулами ліквідності, біржами чи парами валют, котируваних до однієї криптовалюти. Такі стратегії (статистичний арбітраж, pair trading, мультивалютний арбітраж на основі коінтеграції тощо) намагаються ізолюватися від глобального тренду ринку, знижуючи загальну волатильність портфеля. У блокчейн-контексті вони можуть працювати як на централізованих криптобіржах, так і на DeFi-протоколах, використовуючи смартконтракти як основний механізм виконання угод.

Ще однією специфічною рисою є компоновність (composability) DeFi-систем: різні протоколи можуть бути пов'язані між собою як «будівельні блоки».

Алгоритмічна стратегія може одночасно:

- брати ліквідність у одному протоколі;
- проводити арбітраж між кількома DEX;
- використовувати деривативи чи стейблкоїни з іншим профілем

ризиків;

- депонувати заставу в кредитному протоколі.

Це створює багатий простір для побудови складних алгоритмічних стратегій, але також ускладнює їх аналіз і тестування, оскільки необхідно враховувати взаємозв'язки між протоколами, каскадні ризики та можливість системних збоїв.

Таким чином, алгоритмічна торгівля у мережах блокчейн відрізняється від традиційних фінансових ринків не лише набором інструментів, а й фундаментальними властивостями інфраструктури. Розподіленість даних і логіки, наявність смартконтрактів, відкритість ончейн-інформації, специфіка DEX та АММ, вплив gas-комісій, MEV та мережових затримок визначають як вимоги до архітектури програмних систем, так і критерії успішності алгоритмічних стратегій. Це робить предметну область дослідження складною, але водночас перспективною для розробки нових підходів до статистичного арбітражу, портфельної оптимізації та автоматизованого керування криптоактивами.

## 1.2 Огляд існуючих підходів до алгоритмічної торгівлі в блокчейн-мережах

Сучасні підходи до алгоритмічної торгівлі на криптовалютних ринках формуються на перетині класичних фінансових стратегій і нових можливостей, які надають блокчейн та DeFi-протоколи. З одного боку, до криптоактивів намагаються застосовувати добре відомі в традиційних фінансах трендові, арбітражні та маркет-мейкерські стратегії, адаптуючи їх до підвищеної волатильності й нестабільності ринку. З іншого боку, поява децентралізованих бірж, пулів ліквідності, деривативів та крос-ланцюгових протоколів породжує принципово нові можливості для автоматизованої торгівлі, які раніше були недоступними.

У найзагальнішому вигляді алгоритмічні стратегії у блокчейн-екосистемі можна поділити на кілька великих груп [12]. До трендових належать підходи, що намагаються виявляти та використовувати стійкі напрямки руху ціни активу, спираючись на ковзні середні, індикатори моменту, фільтри тренду та інші технічні показники. Стратегії mean-reversion, навпаки, виходять з припущення, що ціна або деяка функція від цін (наприклад, спред між двома активами) має тенденцію повертатися до певного «нормального» рівня. Окреме місце займають методи статистичного арбітражу: вони використовують багатовимірні статистичні моделі, коінтеграцію та кореляційну структуру між активами для побудови маркет-нейтральних портфелів, чутливих не до загального ринкового тренду, а до відносних дисбалансів.

Ще один важливий напрямок пов'язаний із маркет-мейкінгом та наданням ліквідності. У DeFi-середовищі значну роль відіграють автоматизовані маркет-мейкери, де будь-який користувач може надати ліквідність у пул, отримуючи частку комісій. Алгоритмічні стратегії тут можуть оптимізувати розподіл капіталу між різними пулами, вибір діапазонів ліквідності, динамічну перебудову позицій з урахуванням імперманентних втрат, змін волатильності та кореляцій між активами. Такі підходи поєднують елементи високочастотного

управління портфелем із моделюванням специфічної мікроструктури АММ-протоколів.

На додачу до класичних методів активно розвиваються алгоритми, засновані на машинному навчанні, зокрема на глибинних нейронних мережах. Один із поширених підходів полягає у використанні нейромережових моделей для безпосередньої оптимізації структури портфеля без явного етапу прогнозування цін. Такі моделі отримують на вхід часові ряди ринкових показників (доходності, обсяги, технічні індикатори) та навчаються так, щоб максимізувати певний цільовий критерій ефективності, наприклад коефіцієнт Шарпа, який відображає співвідношення “прибуток / ризик”. У результаті система не передбачає рух ціни в явному вигляді, а одразу пропонує, з її точки зору, оптимальні пропорції розподілу капіталу між інструментами. Подібні підходи особливо привабливі для високоволатильних криптовалютних ринків, де класичні моделі прогнозування часто демонструють нестійкі результати.

Інший важливий напрямок пов’язаний із застосуванням навчання з підкріпленням (reinforcement learning) для керування криптовалютними портфелями [13]. У такому випадку ринок розглядається як середовище, з яким взаємодіє агент, а послідовність торгових рішень інтерпретується як політика, що максимізує довгострокову винагороду. Глибокі архітектури на основі рекурентних нейронних мереж (наприклад, LSTM) дають змогу враховувати часову структуру даних, а алгоритми типу Proximal Policy Optimization (PPO)—оптимізувати стратегію з урахуванням ризику, транзакційних витрат та обмежень портфеля. На практиці часто застосовуються ансамблеві схеми, коли кілька незалежно навчених агентів комбінуються в єдину систему ухвалення рішень. Це дозволяє зменшити ризик перенавчання на конкретному ринковому режимі та підвищити стійкість стратегії до нестационарності часових рядів.

Окрему та добре структуровану групу складають стратегії статистичного арбітражу, метою яких є використання відносних цінових дисбалансів між пов’язаними активами. У цьому контексті застосовуються моделі коінтеграції [14], побудова спредів між валютами або токенами, а також різноманітні підходи

до формування маркет-нейтральних портфелів. Замість роботи лише з однією парою активів розглядається система взаємопов'язаних інструментів, для яких припускається існування довгострокових статистичних зв'язків. На основі цих зв'язків будуються спреди, що мають властивість повернення до середнього, а розподіл капіталу між позиціями визначається розв'язанням задач оптимізації з урахуванням дохідності та ризику. Такі підходи дають змогу одночасно враховувати кілька потенційних арбітражних можливостей, використовувати коінтеграційні властивості цінових рядів і контролювати ризиковий профіль портфеля.

Поза межами зазначених напрямів активно розвиваються й інші класи алгоритмічних стратегій, характерних саме для блокчейн-ринків. Сюди належать арбітраж між централізованими та децентралізованими біржами [15], крос-пуловий арбітраж у DeFi [16], коли один і той самий актив має різні ефективні ціни в різних пулах ліквідності, а також використання деривативів та перпетуальних контрактів для хеджування й побудови складних профілів виплат. Окремий пласт становлять підходи до систематичного вилучення можливостей, пов'язаних із максимально видобуваною вартістю (MEV), коли алгоритми намагаються експлуатувати особливості формування блоків та порядку транзакцій. Більшість таких стратегій потребує врахування мережових затримок, структури комісій, глибини пулів, а також потенційного впливу власних транзакцій на ринкову ціну.

Попри значне різноманіття підходів, можна виокремити кілька спільних обмежень і відкритих питань. Частина досліджень досі орієнтована переважно на централізовані біржі або використовує дані, що лише частково відображають специфіку DeFi-мікроструктури, зокрема вплив кривих автоматизованих маркет-мейкерів, імперманентні втрати чи MEV-ризик. Часто недостатньо повно враховуються й технічні обмеження блокчейн-інфраструктури: вартість gas, обмежена пропускна здатність мережі, затримки підтвердження транзакцій, можливість фронтранінгу. Крім того, значна частина моделей машинного навчання розглядає ринок як «чорну скриньку» та мало використовує структурну

інформацію про взаємозв'язки між активами, наприклад коінтеграційні властивості чи логіку формування цін у пулах ліквідності.

Узагальнюючи, можна стверджувати, що існує потреба в методах, які поєднували б сильні сторони статистичного арбітражу – коінтеграцію, маркет-нейтральність, експлуатацію стабільних відносних зв'язків між активами з урахуванням реальних умов функціонування блокчейн-екосистеми: структури DeFi-протоколів, комісій, мережеских затримок та операційних ризиків. Подальший розвиток цієї галузі також пов'язаний з інтеграцією сучасних методів оптимізації та машинного навчання у такі моделі, що дозволить підвищити адаптивність і стійкість алгоритмічних стратегій до змін ринкового середовища.

### 1.3 Формулювання мети та задач дослідження

Метою даного дослідження є аналіз існуючих алгоритмічних трейдингових стратегій для криптовалютних ринків у мережах блокчейн та розробка власної алгоритмічної стратегії, адаптованої до умов DeFi-екосистеми та децентралізованих бірж. При цьому особлива увага приділяється стратегіям статистичного арбітражу та маркет-нейтральним підходам, які використовують відносні зв'язки між активами, а також питанням урахування технічних обмежень блокчейн-інфраструктури.

Для досягнення поставленої мети у межах кваліфікаційної роботи необхідно розв'язати такі основні задачі:

- на основі аналізу предметної області алгоритмічної торгівлі в блокчейн-мережах окреслити особливості децентралізованих фінансових протоколів, мікроструктури DeFi-ринків, джерел даних та типових обмежень інфраструктури;

- виконати огляд і систематизацію сучасних підходів до алгоритмічної торгівлі на криптовалютних ринках, включно з методами портфельної оптимізації, стратегіями на основі машинного навчання та навчання з

підкріпленням, а також статистичним арбітражем і маркет-нейтральними стратегіями;

- виділити переваги та недоліки розглянутих підходів, визначити їх придатність до застосування в умовах децентралізованих бірж і блокчейн-екосистеми, сформулювати вимоги до цільової стратегії, яка має бути розроблена в межах магістерської роботи;

- обрати базовий концептуальний підхід для побудови власної алгоритмічної стратегії (на основі ідей статистичного арбітражу та ринкової нейтральності) та визначити, які саме елементи будуть модифіковані чи доповнені порівняно з існуючими рішеннями;

- спроектувати загальну архітектуру програмного комплексу, необхідного для дослідження стратегії:

- а) модулі збору та попередньої обробки ринкових і ончейн-даних;
- б) компоненти для проведення бектестів;
- в) компоненти для розрахунку метрик ефективності;
- г) компоненти для візуалізації результатів;

- сформулювати перелік ключових показників для оцінювання якості алгоритмічних стратегій та визначити підхід до порівняння існуючих стратегій із розробленою;

- підготувати методичну та технічну основу для подальшої реалізації та експериментального дослідження власної алгоритмічної стратегії.

Виконання зазначених задач дозволить, з одного боку, сформулювати цілісне розуміння поточного стану досліджень у сфері алгоритмічної торгівлі в блокчейн-мережах, а з іншого — закласти фундамент для розробки та подальшого експериментального аналізу власної стратегії, що поєднує теоретичні напрацювання та практичні вимоги DeFi-інфраструктури.

## 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ ТА МЕТОДІВ ДЛЯ ВИРІШЕННЯ ПОДІБНИХ ЗАДАЧ

### 2.1 Загальна характеристика обраних наукових джерел

У межах даного дослідження як базові було обрано декілька наукових праць, які репрезентують різні, але взаємодоповнювальні напрями розвитку алгоритмічних стратегій на фінансових ринках, зокрема на ринках криптоактивів. Ці роботи відрізняються як вибором математичного апарату, так і типом інфраструктури, для якої вони розроблялися (централізовані біржі, деривативи, криптовалютні ринки), що дає змогу сформуванню цілісніший погляд на предметну область.

Перша група ідей пов'язана з використанням глибинних нейронних мереж для безпосередньої оптимізації структури портфеля пропонує підхід, у якому мережа навчається формувати ваги між парою спеціалізованих криптоінструментів із протилежною динамікою (левередж-токени типу «UP»/«DOWN»), максимізуючи коефіцієнт Шарпа сукупного портфеля. Таким чином, автори демонструють можливість побудови енд-ту-енд моделі, що не прогнозує ціни явно, а одразу виробляє торгове рішення у вигляді вектору ваг.

Ця праця є важливою для магістерського дослідження як приклад нейромережевого підходу до адаптивного керування криптовалютним портфелем в умовах високої волатильності.

Другий напрямок представляє сучасний розвиток методів навчання з підкріпленням у задачах автоматизованого керування криптоактивами, який представляє ансамблевий підхід, де декілька агентів на основі глибинного reinforcement learning навчаються ухвалювати послідовні торгові рішення для портфеля з кількох криптовалют. Особливість підходу полягає в поєднанні рекурентних нейронних мереж, що моделюють динаміку ринку, з процедурою відбору та агрегування політик у ансамбль для підвищення стійкості до нестационарності. Для даної магістерської роботи ця праця є показовою з точки

зору інтеграції машинного навчання, часових рядів та портфельного керування в єдину архітектуру.

Третій обраний напрямок пов'язаний із статистичним арбітражем і маркет-нейтральними стратегіями на базі коінтеграції. Розглядається підхід до мультивалютного трейдингу, де декілька фіатних валют котируються до однієї криптовалюти, а на основі їхніх відносних цін будуються спреди, що мають властивість повернення до середнього. Розподіл капіталу між позиціями здійснюється як результат розв'язання оптимізаційної задачі, що одночасно враховує дохідність і ризик, завдяки чому формується маркет-нейтральний портфель, чутливий насамперед до відносних дисбалансів між активами. Цей підхід є важливим для магістерського дослідження, оскільки демонструє, як класичні статистичні методи можуть бути інтегровані в алгоритмічну стратегію з чітко визначеним профілем ризику.

Загалом обрані наукові джерела охоплюють три ключові пласти сучасних досліджень у сфері алгоритмічної торгівлі на ринках криптоактивів: нейромережеву портфельну оптимізацію, глибинне навчання з підкріпленням для керування портфелем та статистичний арбітраж на основі коінтеграційних зв'язків. Така комбінація дозволяє, з одного боку, проаналізувати можливості та обмеження кожного з підходів окремо, а з іншого — виявити потенційні напрямки їх поєднання або модифікації для розробки власної стратегії, адаптованої до умов блокчейн-екосистеми та децентралізованих фінансових протоколів. У наступних підрозділах буде розглянуто структуру, математичний апарат та практичні результати кожної з цих робіт більш детально.

## 2.2 Нейромережева стратегія портфельної оптимізації на основі леверидж-токенів

Перша розглянута стратегія належить до класу енд-ту-енд нейромережевих методів портфельної оптимізації. Схема стратегії наведена на рисунку 2.1. Автори працюють із парою спеціалізованих леверидж-токенів, що

мають протилежну динаміку (типу BNBUP/BNBDOWN, BTCUP/BTCDOWN), і ставлять задачу не прогнозувати напрямок руху ринку, а безпосередньо навчити модель обирати ваги портфеля між цими інструментами так, щоб максимізувати ризик-скоригований результат (Sharpe ratio). Важливо, що йдеться про вузький, але показовий сценарій: портфель складається лише з двох похідних криптоінструментів, а готівка (cash) як окремий актив не використовується - капітал завжди повністю інвестований у леверидж-токени.

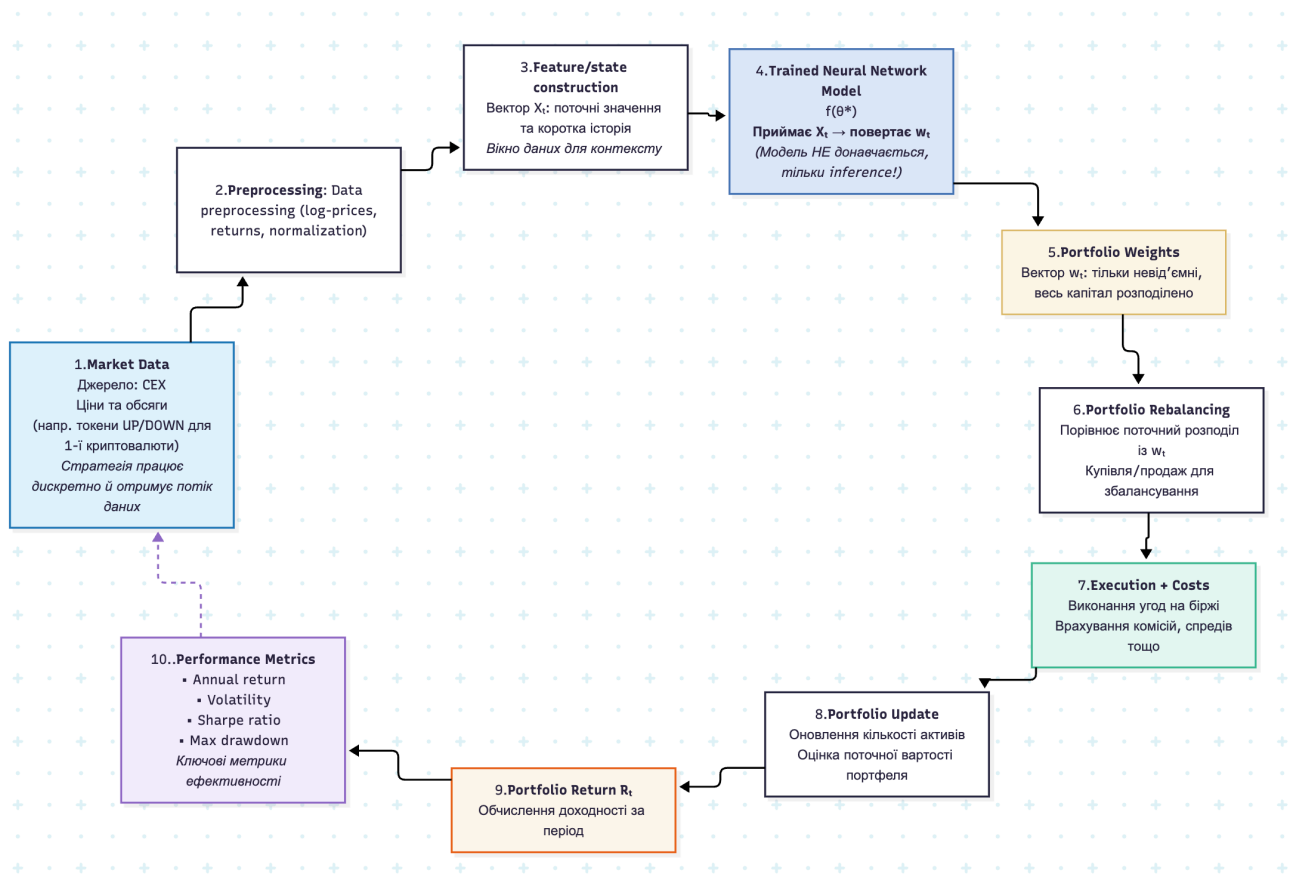


Рисунок 2.1 – Схема нейромережевої стратегії портфельної оптимізації на основі леверидж-токенів

У статті [17] розглядається задача керування портфелем у дискретному часі: кожен крок відповідає одному ребалансуванню. Портфель складається з кількох активів, у найпростішому випадку - з двох леверидж-токенів типу UP та DOWN, прив'язаних до однієї криптовалюти. На кожному кроці стратегія обирає, яку частку капіталу тримати в кожному з активів, причому портфель

завжди повністю інвестований і працює лише в long-режимі (без шортів та кредитного плеча на рівні моделі). Якість роботи стратегії оцінюється за послідовністю портфельних доходностей. На їх основі обчислюється:

- середня періодична дохідність на тестовому інтервалі:

$$\mu_R = \frac{1}{T} \sum_{t=1}^T R_t;$$

- стандартне відхилення доходності (волатильність):

$$\sigma_R = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (R_t - \mu_R)^2}.$$

- коефіцієнт Шарпа, який і виступає головним критерієм ефективності: чим він вищий, тим кращим є співвідношення «дохідність/ризик»:

$$S = \frac{\mu_R}{\sigma_R},$$

Завдання моделі - знайти таку послідовність рішень щодо структури портфеля, яка максимізує саме Sharpe ratio.

Нейронна мережа у цій роботі виступає як функція, що перетворює стан ринку на кроці  $t$  (набір ознак з історичних цін, доходностей, обсягів та простих технічних індикаторів за певне «вікно» часу) на вектор ваг портфеля. Вихідний шар моделі налаштований так, щоб ваги були невід'ємними і в сумі давали одиницю. Для заданих параметрів мережі можна «програти» на історичних даних повну траєкторію: на кожному кроці обчислюється новий розподіл капіталу, ребаланс портфеля та відповідна дохідність.

Навчання моделі проводиться в енд-ту-енд режимі: замість прогнозування цін вона безпосередньо оптимізує критерій якості портфеля. Функція втрат

побудована як негативний коефіцієнт Шарпа з додаванням регуляризації, яка обмежує складність моделі та надмірну концентрацію капіталу в одному активі. Оптимізація параметрів мережі виконується стандартними градієнтними методами, а всі операції з портфелем реалізуються як диференційовний обчислювальний граф, що дозволяє «прокочувати» похідні через увесь ланцюг розрахунків.

Як базові активи в оригінальній роботі використовувалися леверидж-токени, які випускалися централізованою біржею (наприклад, BNBUP та BNBDOWN) і мали протилежну поведінку відносно базового активу: один токен заробляє на зростанні, інший — на падінні. Така пара природно підходить для побудови частково хеджованого, близького до маркет-нейтрального портфеля, де модель динамічно перерозподіляє капітал між «довгою» та «короткою» стороною. На історичних даних показано, що цей підхід може перевершувати за показниками «дохідність/ризик» прості базові стратегії «купити й тримати» або фіксовані ваги. Водночас у поточних умовах конкретні токени з типу BNBUP/BNBDOWN уже не торгуються, тому пряму постановку повторити неможливо, але сам принцип можна перенести на інші пари інструментів з протилежною чи сильно відмінною динамікою.

Розглянута нейромережева стратегія має низку важливих переваг. По-перше, вона є концептуально простою: замість побудови окремої моделі прогнозування цін або доходностей та подальшого застосування зовнішнього оптимізатора, автори одразу навчають модель на цільовий показник ефективності портфеля (Sharpe ratio), що зменшує кількість проміжних припущень. По-друге, використання пари токенів із протилежною динамікою частково знижує залежність від загального напрямку ринку й дозволяє моделі фокусуватися на відносній поведінці активів. По-третє, вся логіка прийняття рішень реалізується в єдиній диференційованій моделі, що відкриває шлях до подальших розширень (включення транзакційних витрат, додаткових обмежень тощо) в межах того ж формалізму.

Водночас стратегія має істотні обмеження. Вона тестується на дуже вузькому класі інструментів — специфічних леверидж-токенах, які до того ж більше не доступні на ринку, що знижує практичну значущість результатів у первісному вигляді. Модель працює з невеликим числом активів (фактично з парою), що обмежує можливості диверсифікації портфеля. Існує також ризик переобучення на конкретній вибірці історичних даних, особливо з огляду на нестабільність криптовалютних ринків. Крім того, у вихідній постановці обмежено враховуються такі фактори, як проскальзування, глибина ринку, змінні комісії та інші аспекти мікроструктури торгівлі, які є критично важливими для реальної експлуатації подібних стратегій.

Таким чином, перша стратегія є цінним прикладом застосування глибинних нейронних мереж для прямої оптимізації портфеля в криптосередовищі, але потребує адаптації до актуальних інструментів та розширення постановки задачі, якщо її результати планується перенести у практичний контекст DeFi та сучасних блокчейн-ринків.

### 2.3 Ансамблевий підхід на основі глибокого навчання з підкріпленням для керування криптовалютним портфелем

Друга розглянута стратегія представляє напрям, у якому алгоритмічна торгівля формулюється як задача навчання з підкріпленням [18]. Схема цієї стратегії наведена на рисунку 2.2. На відміну від підходів, де насамперед прогнозуються ціни або доходності, тут ринок розглядається як середовище, а агент безпосередньо навчається ухвалювати послідовні торгові рішення, максимізуючи довгострокову винагороду. Об'єктом управління є портфель з кількох провідних криптовалют, до якого додається безризиковий актив у вигляді готівкової позиції (стейблкоїн або фіат). Стратегія працює на даних централізованої біржі, де доступні історичні котирування, обсяги та інші ринкові показники з фіксованим кроком часу (наприклад, годинним).

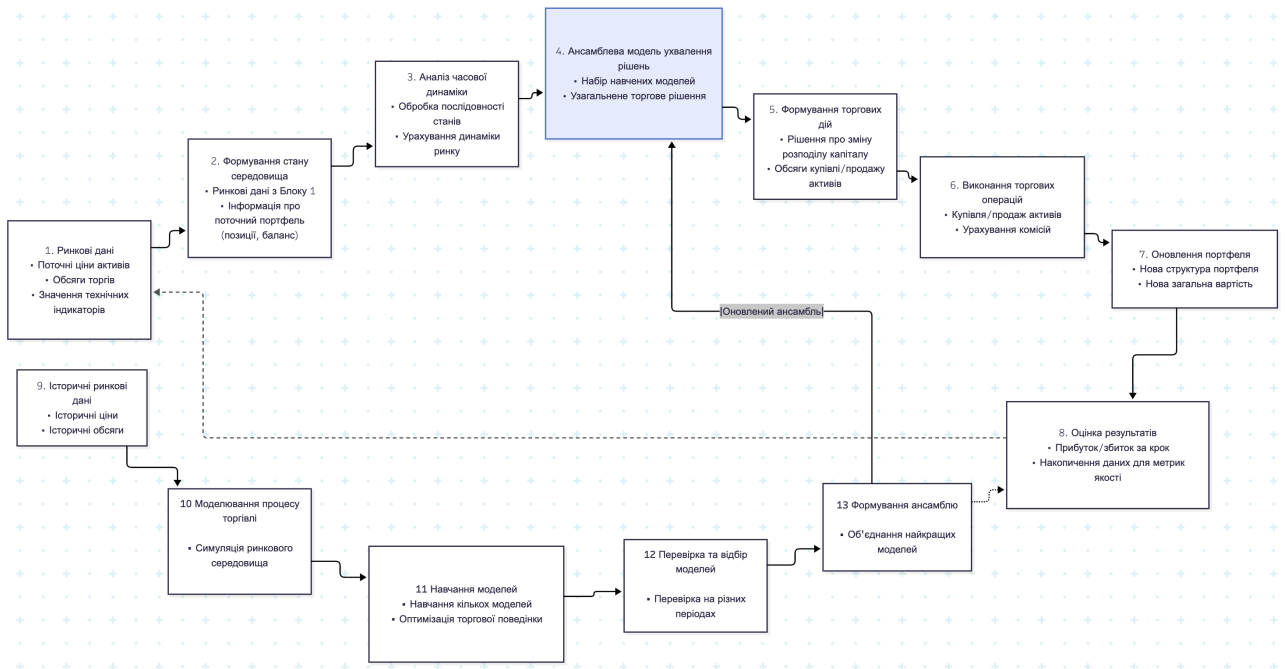


Рисунок 2.2 – Схема ансамблевого підходу на основі глибокого навчання з підкріпленням для керування криптовалютичним портфелем

Стан середовища на кожному кроці подається у вигляді вектора ознак, що містить історію цін, обсягів, технічні індикатори та поточний розподіл портфеля. Щоб урахувати часову залежність, використовується рекурентна мережа LSTM, яка обробляє послідовність попередніх кроків і на виході формує цільові зміни позицій (нові ваги активів у портфелі). Навчання агента виконується методом PPO: політика оновлюється обережно, щоб уникнути різких змін, а функція винагороди поєднує дохідність портфеля з штрафами за високу волатильність, глибокі просадки та транзакційні витрати, стимулюючи баланс між прибутком і ризиком.

Ключовою особливістю розглянутої роботи є використання ансамблю агентів замість однієї моделі. На етапі навчання автори тренують велику кількість моделей з різними початковими умовами, підвбірками даних та гіперпараметрами. Далі для кожної моделі оцінюється якість на кількох валідаційних інтервалах, що не використовувалися для безпосереднього навчання. У ансамбль відбираються лише ті агенти, які показують стабільно

задовільні результати на різних періодах, а не тільки на одному «вдалому» відрізку. Таким чином зменшується ризик перенавчання до конкретного ринкового режиму або випадкових закономірностей в історичних даних.

Після відбору агентів їхні політики комбінуються: у найпростішому варіанті торгове рішення на кожному кроці є усередненням дій, які пропонують окремі моделі ансамблю. Такий «колективний» підхід дозволяє згладити випадкові помилки окремих агентів, зменшити чутливість до шуму в даних і зробити поведінку системи більш стабільною. На практиці це проявляється у рівнішій кривій балансу, кращому співвідношенні «прибуток/ризик» та меншій схильності до глибоких локальних просадок порівняно з одиничними DRL-стратегіями.

Експериментальна частина стратегії базується на довготривалому історичному тестуванні: дані розбиваються на послідовні інтервали для навчання, валідації та тесту, що імітує реальний сценарій розгортання системи в онлайні.

Переваги такого підходу полягають у його гнучкості та здатності адаптуватися до складних нелінійних залежностей у даних. DRL-агент може потенційно використовувати інформацію з багатьох джерел, враховувати транзакційні витрати, зміну волатильності, кореляційні структури між активами та інші фактори, не вимагаючи явної аналітичної моделі ринку. Ансамблеве представлення додатково підвищує надійність і дозволяє зменшити ризик катастрофічних помилок, притаманних окремим моделям.

Водночас стратегія має і суттєві недоліки. По-перше, навчання ансамблю глибоких моделей є обчислювально дорогим і вимагає значних обсягів якісних історичних даних, бажано з урахуванням реалістичних комісій, проскальзування та обмежень ліквідності. По-друге, модель залишається «чорної скринькою» з точки зору інтерпретації: важко формально довести, чому саме агент робить певні дії, і які закономірності він використовує. По-третє, перенос результатів із середовища централізованих бірж до DeFi-контексту не є прямим: потрібно

додатково враховувати gas-витрати, затримки включення транзакцій у блок, вплив стратегії на стан пулів ліквідності та ризику MEV.

Отже, розглянута стратегія на основі ансамблевого глибинного навчання з підкріпленням демонструє високий потенціал для автоматизованого керування криптовалютичним портфелем, особливо в контексті централізованих бірж, де добре контрольовані умови торгівлі. Вона показує, як поєднання DRL, рекурентних нейромереж та ансамблевих методів дозволяє підвищити адаптивність і стійкість алгоритмічних стратегій. Водночас для застосування подібних підходів у блокчейн-мережах і на DeFi-протоколах необхідна подальша адаптація до специфіки ончейн-інфраструктури та мікроструктури децентралізованих ринків.

2.4 Оптимізаційний підхід до мультивалютного статистичного арбітражу з використанням криптовалютичного “якоря”

Третя проаналізована робота [19] представляє інший клас алгоритмічних стратегій – мультивалютний статистичний арбітраж із чітко заданим маркет-нейтральним профілем. Схема цієї стратегії наведена на рисунку 2.3. На відміну від підходів, де основна увага приділяється прогнозуванню динаміки окремого активу або портфеля, у цій стратегії акцент робиться на відносних цінових дисбалансах між кількома фіатними валютами, які котируються до однієї криптовалюти (так званого “якоря”, наприклад ETH).

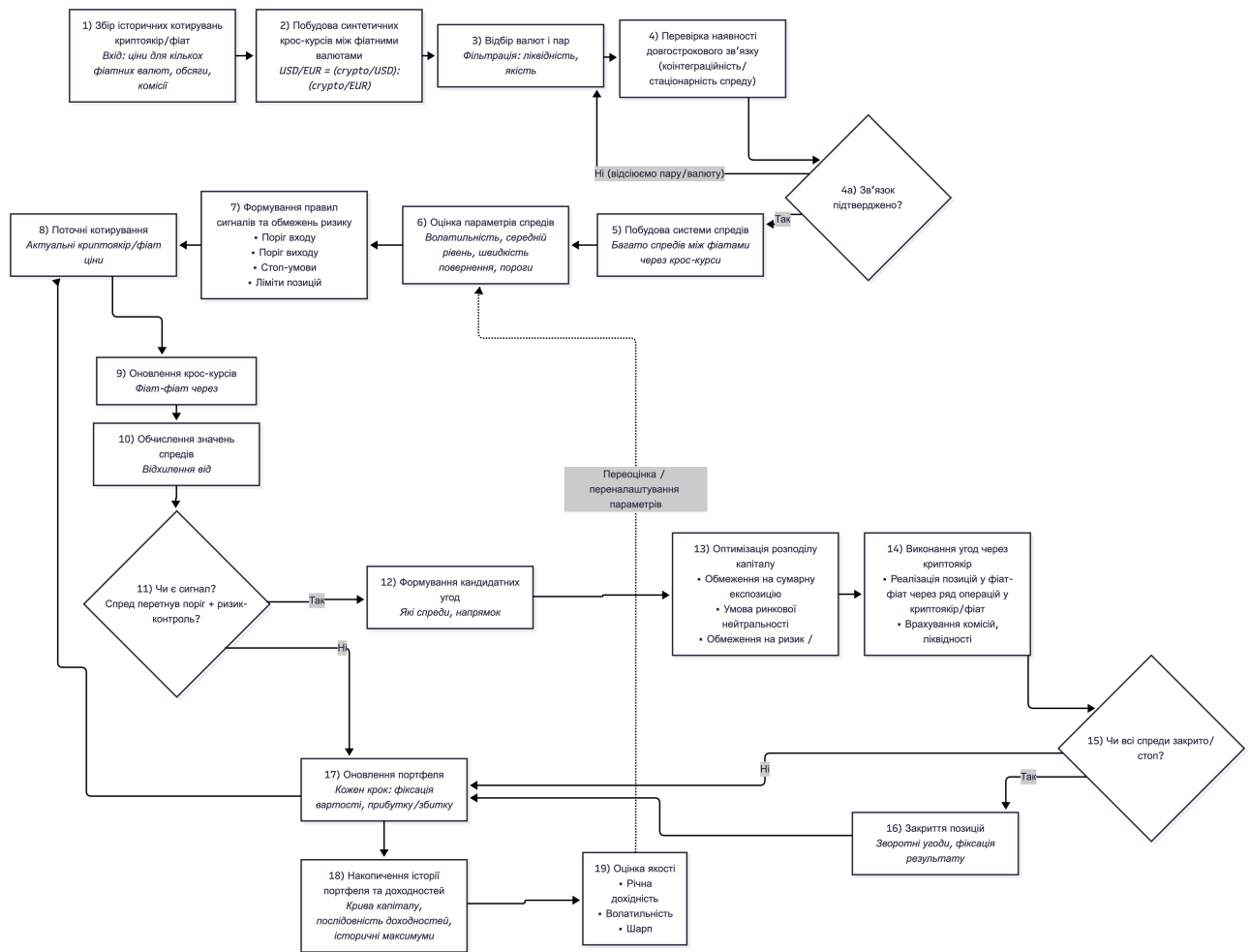


Рисунок 2.3 – Схема оптимізаційного підходу до мультивалютного статистичного арбітражу з використанням криптовалютного «якоря»

Стратегія спирається на те, що на біржі одночасно торгуються пари ETH/USD, ETH/EUR, ETH/GBP, ETH/CAD тощо. Через них утворюються синтетичні крос-курси між фіатними валютами, які в ідеально ефективному ринку мали б задовольняти арбітражні співвідношення, але на практиці дають короточасні відхилення. Автори трактують ці відхилення як джерело прибутку: вони не прогнозують напрямок руху окремої валюти, а заробляють на вирівнюванні відносних дисбалансів між кількома фіатними валютами, використовуючи криптовалюту як «якір». Спочатку з котирувань ETH/fiat будуються крос-курси й спреди, які порівнюються з історично узгодженими рівнями (коінтеграційний підхід). Далі формується портфель, що одночасно

реагує на кілька таких спредів, реалізуючи мультивалютний варіант статистичного арбітражу замість класичної торгівлі однією парою.

Ключовим елементом стратегії є постановка задачі у вигляді двокритеріальної оптимізації. З одного боку, потрібно максимізувати очікувану дохідність портфеля, що формується за рахунок очікуваного «закриття» цінових дисбалансів (повернення спредів до рівноважних значень). З іншого боку, необхідно обмежити ризик — у роботі це може інтерпретуватися як варіація портфеля, величина маржі чи інші показники, що відображають небажані коливання. В результаті побудовано математичну модель, у якій на кожному кроці часу вибирається вектор позицій по інструментах так, щоб знайти компроміс між максимізацією прибутку та обмеженням ризику за заданими обмеженнями.

Особливу увагу в роботі приділено реалістичності моделі торгівлі. На відміну від багатьох теоретичних досліджень, де ціни розглядаються як ідеальні середні котирування, тут враховується мікроструктура реального ринку: bid/ask-спреда, обмеження глибини книги заявок, комісії біржі, можливе часткове виконання ордерів. Стратегія симулюється на високочастотних даних (з малим кроком часу), а виконання угод моделюється так, ніби ордери потрапляють у реальний потік заявок і можуть не бути виконані, якщо на ринку недостатньо ліквідності за бажаною ціною. Такий підхід наближає результати дослідження до практичного застосування.

Автори показують, що за правильного налаштування фільтрів і параметрів оптимізації стратегія дає стабільний позитивний результат із контрольованим ризиком: портфель залишається близьким до маркет-нейтрального, просадки обмежені, а основний прибуток формується за рахунок великої кількості дрібних арбітражних угод на короткочасних дисбалансах між валютами. Перевага підходу в тому, що він має чіткий економічний зміст (експлуатація відхилень від довгострокової узгодженості валютних курсів), менш чутливий до глобальних трендів і дає можливість явно керувати ризиком через оптимізаційну постановку. Водночас стратегія потребує високої ліквідності, сильно залежить від комісій

біржі та складної інфраструктури збору й обробки даних, а вихідна модель орієнтована на централізовану біржу й не враховує специфічних особливостей DeFi. У цілому вона є показовим прикладом масштабування класичного статистичного арбітражу до мультивалютної маркет-нейтральної системи й добре ілюструє, які практичні фактори потрібно враховувати при адаптації подібних підходів до блокчейн-екосистеми.

## 3 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

### 3.1 Загальна постановка задачі дослідження

У межах магістерського дослідження розглядається задача побудови алгоритмічної стратегії статистичного арбітражу на криптовалютних ринках, яка ґрунтується на ідеї коінтегрованих спредів між активами та орієнтована на подальшу адаптацію до умов блокчейн-екосистеми (централізовані біржі, децентралізовані біржі, DeFi-протоколи). Схема цієї стратегії наведена на рисунку 3.1.

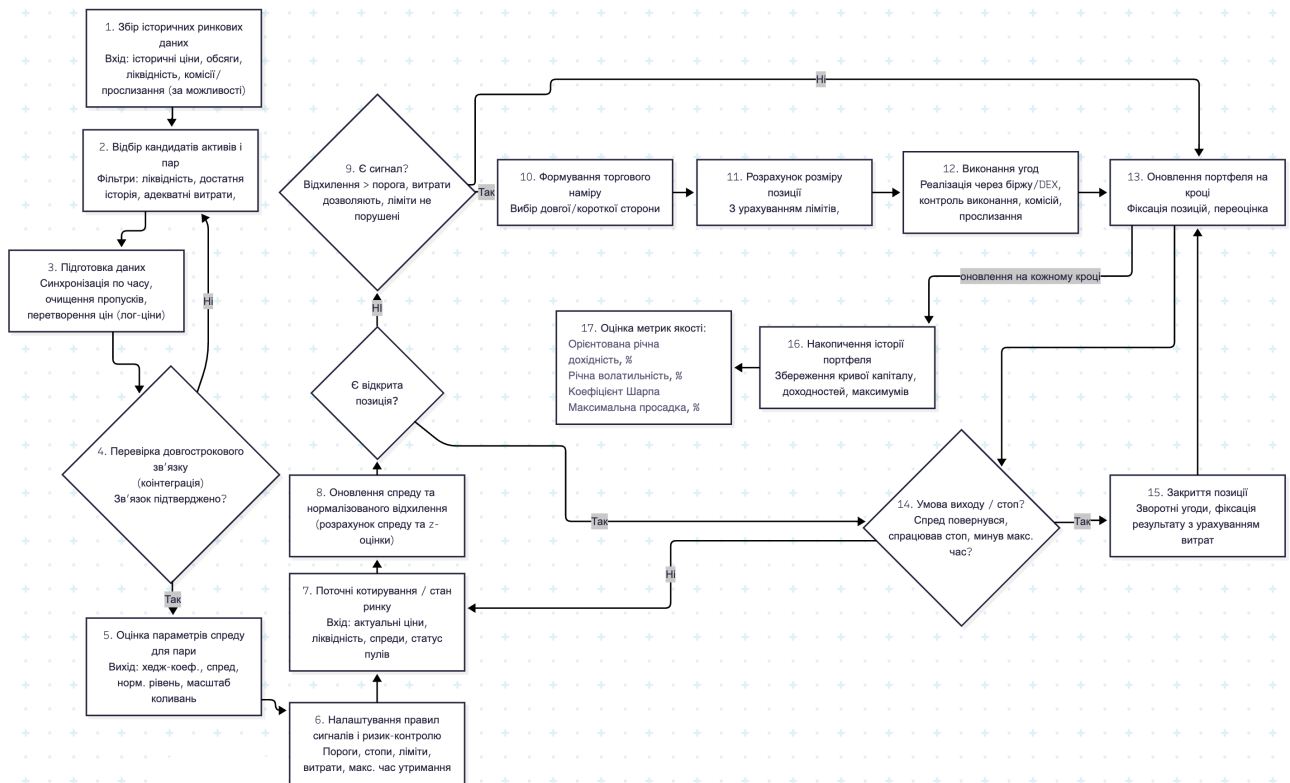


Рисунок 3.1 – Схема коінтеграційної стратегії спред-арбітражу

Змістовно задача формулюється так: маючи часові ряди цін групи криптоактивів, необхідно виявити комбінації активів, між якими існують довгострокові статистичні зв'язки (коінтеграція), побудувати на їх основі

спреди, що мають властивість повернення до середнього (mean reversion), та визначити правила відкриття й закриття ринково-нейтральних позицій, які дозволяють отримувати прибуток за рахунок тимчасових відхилень спреду від рівноважного значення.

Формально дослідження спрямоване на побудову сімейства алгоритмічних стратегій, які можна розглядати як відображення

$$\pi: \{\text{історія ринкових даних}\} \rightarrow \{\text{послідовність торгових рішень}\},$$

де під «торговими рішеннями» розуміються дії щодо відкриття, зміни або закриття позицій по вибраних активах (або портфелях активів). Для кожної стратегії  $\pi$  ріл на історичному інтервалі спостережень будується траєкторія значень портфеля, після чого обчислюється набір показників якості (середня доходність, волатильність, коефіцієнт Шарпа, максимальна просадка тощо).

Таким чином, у загальному вигляді задача дослідження може бути описана як задача пошуку та аналізу таких правил  $\pi$  ріл, які:

- базуються на спредах між коінтегрованими комбінаціями криптоактивів;
- забезпечують ринково-нейтральний або близький до ринково-нейтрального профіль портфеля;
- демонструють прийнятні показники ефективності за обмежень на ризик та транзакційні витрати;
- можуть бути в подальшому адаптовані до особливостей блокчейн-інфраструктури (DEX, gas, slippage, глибина пулів ліквідності).

### 3.2 Вхідні дані, ринкове середовище та основні позначення

Розглядається дискретний час з фіксованим кроком  $\Delta t$  (наприклад, 1 хвилина, 5 хвилин, 1 година). Моменти часу позначаються  $t = 0, 1, 2, \dots, T$ . Нехай  $A = \{1, 2, \dots, N\}$  - множина криптоактивів, що беруться до аналізу. Для кожного

активу  $i \in A$  задано часовий ряд ринкових цін у певній базовій валюті (наприклад, USDT). Значення  $p_{t,i}$  позначає ціну активу  $i$  у момент часу  $t$ . Сукупність цін усіх активів у момент  $t$  зручно розглядати як вектор  $p_t = (p_{t,1}, \dots, p_{t,N})$ .

На основі цін можуть обчислюватися різні похідні показники - прості та логарифмічні доходності, волатильність, ковзні середні, індикатори імпульсу тощо. Проста доходність окремого активу за період  $[t - 1, t]$  інтерпретується як відносна зміна його ціни за цей інтервал. Для подальшого аналізу доцільно працювати не лише з самими цінами  $p_{t,i}$ , а й з вектором доходностей  $r_t = (r_{t,1}, \dots, r_{t,N})$ , який можна розглядати як ще одну форму вхідних даних.

Передбачається, що вихідні ринкові дані надходять від торгового майданчика або спеціалізованого сервісу. Це може бути централізована біржа (CEX) з доступом до книг заявок, котирувань bid/ask та обсягів торгів, або агрегатор/індексатор для децентралізованих бірж (DEX), який надає інформацію про стан пулів ліквідності, виконані свопи та інші ончейн-показники. У рамках формальної постановки задачі ці джерела абстрагуються до послідовності векторів цін  $p_t$  та, за потреби, додаткових змінних (обсягів, спредів, показників ліквідності), тоді як конкретні способи доступу до даних (API CEX, RPC Solana, індексатори) розглядаються в окремих розділах.

Для опису портфеля вводяться такі позначення. Кількість одиниць активу  $i$ , що утримуються у портфелі в момент часу  $t$ , позначається  $x_{t,i}$ ; відповідно,  $x_t = (x_{t,1}, \dots, x_{t,N})$  - вектор кількостей. Поточна вартість портфеля  $V_t$  визначається як сума добутків кількостей на ціни всіх активів. Надалі зручніше працювати з відносними вагами активів у портфелі: вага  $w_{t,i}$  показує, яка частка загальної вартості портфеля припадає на актив  $i$  у момент  $t$ . Вектори ваг позначаються  $w_t = (w_{t,1}, \dots, w_{t,N})$ . Сума всіх ваг дорівнює одиниці, а знак ваги може відображати напрямок позиції (довга або коротка), якщо це допускається конкретним торговим середовищем.

Окремо враховуються параметри, пов'язані з транзакційними витратами. Для кожного активу можна ввести ефективну відносну комісію  $c_i$ , яка включає торгові комісії біржі, спреди bid/ask, а для DEX також витрати на gas і проскальзування. У теоретичній постановці ці параметри вважаються заданими, а їхній вплив враховується на етапі переходу від «грубих» доходностей до реальних результатів портфеля після всіх витрат.

Отже, вхідними даними для подальшого дослідження є часові ряди цін  $p_{t,i}$  для набору активів, похідні ринкові показники (доходності, обсяги, індикатори ліквідності), а також параметри транзакційних витрат і можливі обмеження на позиції. Вихідними величинами, що використовуються для порівняння стратегій, виступають траєкторія вартості портфеля  $V_t$  (або нормовані доходності) та агреговані показники якості роботи стратегії. На основі цих позначень у наступних підпунктах формулюється модель спреда, правила побудови позицій та критерії оптимізації параметрів коінтеграційної стратегії.

### 3.3 Математична модель спреда та коінтегрованих портфелів

У цьому підпункті формалізується та частина моделі, яка безпосередньо пов'язана з ідеєю коінтеграції та побудовою спреда між активами. Саме на цьому рівні з'являється математичне обґрунтування того, чому обрана комбінація цін може мати властивість повернення до середнього (mean reversion) і, відповідно, бути придатною для побудови стратегії статистичного арбітражу.

#### 3.3.1 Лог-ціни та коінтеграція

Як і в більшості робіт зі статистичного арбітражу, для аналізу довгострокових зв'язків між активами замість «звичайних» цін зручно використовувати їхні логарифми. Для кожного активу будується часовий ряд лог-цін, які можна розглядати як вектор стану ринку у момент часу.

Припускається, що окремі ряди лог-цін є інтегрованими процесами порядку 1: їхні прирости є стаціонарними, тоді як самі рівні - ні.

Коінтеграцією називають ситуацію, коли для кількох таких нестаціонарних рядів існує лінійна комбінація (спред), яка вже є стаціонарною. Іншими словами, хоча окремі ціни можуть довго «блукати» вгору або вниз, певне їхнє зважене поєднання коливається навколо відносно сталого рівня. Саме ця стаціонарна комбінація й використовується як цільовий процес для побудови арбітражної стратегії: очікується, що значні відхилення спреда від «норми» з часом коригуються.

У найпростішому випадку двох активів спред можна інтерпретувати як різницю між лог-ціною першого активу та лог-ціною другого, помноженою на певний коефіцієнт співвідношення. У багатовимірному випадку коефіцієнти утворюють вектор, а за наявності кількох незалежних довгострокових зв'язків можливі й кілька різних спредів.

Перевірка наявності коінтеграції в роботі може виконуватися за класичною двокроковою процедурою Енгла–Грейнджера. Спочатку оцінюється регресійна залежність між лог-ціною одного активу та лог-цінами інших; відхилення фактичних значень від отриманої регресії розглядаються як оцінка спреда. Далі до цього ряду залишків застосовується тест на наявність одиничного кореня (наприклад, розширений тест Дікі–Фуллера). Якщо залишки виявляються стаціонарними, гіпотеза про коінтеграцію приймається, а відповідний спред використовується для побудови торгових сигналів.

У випадку більшої кількості активів може застосовуватися метод Йохансена, який дозволяє виявити одну або декілька незалежних коінтеграційних комбінацій. Для подальшої постановки задачі достатньо припустити, що існує принаймні один спред, який має властивість стаціонарності й надалі розглядається як базовий процес для коінтеграційної стратегії.

### 3.3.2 Модель спреда як mean-reverting процесу

Після того, як спред  $S_t$  визначено, робиться припущення, що він має властивість повернення до середнього (mean reversion). У найпростішому випадку це можна змоделювати як авторегресійний процес першого порядку (AR(1)) [12]:

$$S_t = \mu + \phi(S_{t-1} - \mu + \varepsilon_t),$$

де  $\mu$  – довгостроковий середній рівень спреда,  $|\phi| < 1$  – параметр швидкості повернення до середнього, а  $\varepsilon_t$  – випадковий шум із нульовим математичним сподіванням.

Якщо  $|\phi|$  суттєво менше 1, процес швидко повертається до середнього значення  $\mu$  після відхилення, що є бажаною властивістю для статистичного арбітражу: сильне відхилення  $S_t$  від  $\mu$  створює потенційну можливість відкриття позиції, орієнтованої на подальше «закриття» цього відхилення. Для кількісної оцінки швидкості повернення до середнього інколи використовують поняття періоду напіврозпаду (half-life), який показує, за скільки кроків очікуване відхилення зменшується приблизно вдвічі.

Для подальшого використання в торговій стратегії зручно перейти від «сирого» спреда до його нормованої версії. Оцінимо середнє значення  $\hat{\mu}$  та стандартне відхилення  $\hat{\sigma}_S$  спреда на деякому навчальному інтервалі й введемо z-скор (нормалізований спред) [17]:

$$Z_t = \frac{S_t - \hat{\mu}}{\hat{\sigma}_S}.$$

Величина  $Z_t$  показує, наскільки поточне значення спреда є більшим чи меншим за "нормальний" рівень у одиницях стандартного відхилення. Саме  $Z_t$

зазвичай використовується як основна сигнальна змінна: значне додатне або від’ємне значення  $Z_t$  інтерпретується як можливість відкрити арбітражну позицію, а значення, близькі до нуля, – як сигнал до закриття або утримання нейтральної позиції.

### 3.3.3 Коінтегрований портфель та ринкова нейтральність

Оскільки спред  $S_t$  є лінійною комбінацією лог-цін активів, йому відповідає певний портфель активів із вагами, пропорційними коефіцієнтам  $\beta$ . У найпростішому випадку двох активів, якщо

$$S_t = x_{t,1} - \beta x_{t,2},$$

це можна інтерпретувати як портфель, у якому ми займаємо «одну одиницю» лог-позиції в першому активі та «мінус  $\beta$  одиниці» – у другому. На практиці це означає відкриття довгої позиції в одному активі та короткої позиції в іншому з певним співвідношенням обсягів.

Якщо коефіцієнти  $\beta$  підібрані коректно, зміни загального ринкового рівня (наприклад, одночасне зростання обох активів) у значній мірі компенсуються всередині портфеля, а основним джерелом доходу стає саме коливання спреда навколо середнього. Такий портфель називають маркет-нейтральним або «близьким до маркет-нейтрального», оскільки його чутливість до глобальних ринкових трендів зменшена.

У загальному багатовимірному випадку коінтегрований портфель можна записати як [20]

$$P_t = \sum_{i=1}^N \beta_i x_{t,i},$$

де  $\beta_i$  задають відносні ваги (з урахуванням знаку), а сама величина  $P_t$  є логарифмічною вартістю певної синтетичної позиції. Задача статистичного

арбітражу полягає в тому, щоб знаходити такі комбінації  $\beta$  і такі моменти часу, коли відхилення  $P_t$  (або його нормованої версії  $Z_t$ ) від довгострокового рівня є достатньо великими для відкриття позицій, але при цьому очікується подальше повернення до середнього.

У наступному підпункті на основі введених вище понять буде сформульовано конкретні правила побудови торгових позицій, умови відкриття та закриття угод, а також уточнено критерії оцінювання ефективності стратегії, що базується на коінтегрованому спреді.

### 3.4 Постановка задачі побудови торгової стратегії на основі коінтегрованого спреда

У цьому підпункті формалізується задача побудови торгової стратегії, що безпосередньо використовує спред  $S_t$  та його нормовану версію  $Z_t$ , визначені у підпункті 4.3. Мета – задати правила формування позицій по активах, умови входу та виходу з ринку, а також базові обмеження, які повинна задовольняти стратегія статистичного арбітражу.

#### 3.4.1 Сигнальна змінна та правила входу/виходу

Як основну сигнальну змінну використовуємо нормалізований спред [20]:

$$Z_t = \frac{S_t - \hat{\mu}}{\hat{\varepsilon}_S},$$

де  $\hat{\mu}$  та  $\hat{\varepsilon}_S$  – оцінки середнього значення та стандартного відхилення спреда на деякому історичному інтервалі (навчальній вибірці).

Інтуїтивно

- якщо  $Z_t$  набуває великих додатних значень, спред “занадто високий”

відносно свого середнього, і можна очікувати його зниження;

- якщо  $Z_t$  є великим від'ємним, спред «занадто низький», і можна очікувати зростання.

Вводяться два пороги:

- $\tau_{open} > 0$  – поріг для відкриття позиції;
- $\tau_{close} > 0$  – поріг для закриття позиції, причому зазвичай

$\tau_{close} < \tau_{open}$ , щоб уникати надмірно частих перекидок позицій.

Нехай  $h_t \in \{-1, 0, +1\}$  – індикаторна змінна стану стратегії в момент часу  $t$ :

- $h_t = 0$  – позиція закрита (стратегія поза ринком за даним спредом);
- $h_t = +1$  – відкрита позиція «long spread» (розрахунок на зростання спреда);
- $h_t = -1$  – відкрита позиція «short spread» (розрахунок на зниження спреда).

Правила переключення між цими станами можна записати так:

- Вхід у short spread (очікується зниження спреда): якщо  $h_{t-1} = 0$  та  $Z_t \geq \tau_{open}$ , тоді  $h_t = -1$ ;
- Вхід у long spread (очікується зростання спреда):

якщо  $h_{t-1} = 0$  та  $Z_t \leq -\tau_{open}$ , тоді  $h_t = +1$ ;

- Закриття позиції: якщо  $h_{t-1} \neq 0$  та  $|Z_t| \geq \tau_{close}$ , тоді 0;
- Утримання позиції: в усіх інших випадках  $h_t = h_{t-1}$ .

Таким чином, стратегія відкриває позиції лише тоді, коли відхилення спреда від середнього перевищує певний «суттєвий» поріг, а закриває їх, коли спред повертається до «нормальної» зони.

### 3.4.2 Формування позицій по активах

Оскільки спред  $S_t$  є лінійною комбінацією лог-цін активів з коефіцієнтами  $\beta$ , йому відповідає певний синтетичний портфель. Для простоти розглянемо спочатку випадок двох активів, де

$$S_t = x_{t,1} - \beta x_{t,2}.$$

У цьому випадку позиція «short spread» ( $h_t = -1$ ) може інтерпретуватися як:

- продати (зайняти коротку позицію) в активі 1;
- купити (зайняти довгу позицію) в активі 2 у пропорції, що відповідає коефіцієнту  $\beta$ .

Позиція “long spread” ( $h_t = +1$ ), навпаки, означає:

- купити актив 1;
- продати актив 2 (у відповідній пропорції).

У загальному випадку для вектора коефіцієнтів  $\beta = (\beta_1, \dots, \beta_N)$  можна ввести вектор позицій по активах

$$q_t = h_t k \underline{\beta},$$

де  $k > 0$  – коефіцієнт масштабу (який задає розмір позиції відносно загального капіталу), а  $\underline{\beta}$  – нормалізована версія вектора  $\beta$  (наприклад, після масштабування так, щоб сумарна абсолютна вартість позицій відповідала заданому рівню ризику). Компонента  $q_{t,i}$  задає знакову кількість одиниць активу  $i$ , яку потрібно тримати у портфелі в момент часу  $t$ : знак  $q_{t,i}$  визначає напрямок позиції (довга/коротка), а модуль – її розмір.

Подальша динаміка портфеля задається як

- у момент відкриття позиції (перехід із  $h_{t-1} = 0$  до  $h_t \neq 0$ ) відбувається перебалансування портфеля до вектора позицій  $q_t$ ;
- у момент закриття позиції (перехід із  $h_{t-1}$  до  $h_t = 0$ ) всі позиції по відповідних активах закриваються (вектор  $q_t$  стає нульовим);
- між цими моментами (якщо  $h_t = h_{t-1}$ ) позиції або утримуються

без змін, або можуть підлягати частковій перебалансувальній корекції (наприклад, для підтримання фіксованої суми інвестованого капіталу).

Таким чином, набір параметрів стратегії включає, принаймні:

- вектор коефіцієнтів спреда  $\beta$ ;
- пороги  $\tau_{open}, \tau_{close}$ ;
- коефіцієнт масштабу позиції  $k$ ;
- додаткові технічні параметри (мінімальний час утримання позиції, фільтри ліквідності тощо).

### 3.4.3 Врахування транзакційних витрат та обмежень

Реальна ефективність стратегії істотно залежить від транзакційних витрат, до яких належать:

- біржові комісії за кожну операцію (торгові комісії CEX або комісії DEX);
- спреди між цінами купівлі та продажу;
- у контексті блокчейну – витрати на газ та можливе проскальзування ціни при виконанні свопів у пулах ліквідності.

У базовій постановці моделі припускається, що для кожного активу ііі відомий ефективний відносний рівень витрат  $c_i$ . При зміні портфеля з вектора позицій  $q_{t-1}$  до  $q_t$  частина капіталу втрачається на комісіях, що зменшує фактичну доходність стратегії. Це змушує:

- уникати надмірно частих перебалансувань (звідси необхідність мати  $\tau_{close} < \tau_{open}$  і включати фільтри шуму);
- обмежувати розмір позицій у малоліквідних активах;
- враховувати мінімальний торговий обсяг, за якого потенційний прибуток перевищує витрати.

Крім транзакційних витрат, до стратегії накладаються ризикові та технічні обмеження, зокрема:

- обмеження на максимальний розмір позиції по окремому активу;
- обмеження на загальне кредитне плече;
- вимоги до мінімальної ліквідності торгованих активів;
- обмеження на сумарну експозицію до певної групи активів (наприклад, стейблкоїни, токени одного проєкту тощо).

Ці обмеження визначають допустиму область значень параметрів стратегії та множину допустимих траєкторій  $q_t$ .

У підсумку задача побудови торгової стратегії на основі коінтегрованого спреда формулюється як задача визначення параметрів

$$\Theta = \{\beta, \tau_{open}, \tau_{close}, k, \dots\},$$

та правил переходу між станами  $h_t$  і позиціями  $q_t$ , які, з одного боку, відповідають логіці mean-reversion спреда, а з іншого – задовольняють обмеженням щодо ризику, ліквідності й транзакційних витрат. У наступному підпункті буде сформульовано критерії якості, за якими здійснюватиметься оцінювання й подальша оптимізація таких стратегій.

### 3.5 Критерії якості та постановка задачі порівняльного аналізу стратегій

Для того щоб результати дослідження були співставними з результатами, наведеними в розглянутих наукових публікаціях, необхідно чітко визначити набір кількісних показників, за якими буде оцінюватися ефективність стратегії статистичного арбітражу на основі коінтегрованого спреда. Ці ж показники надалі будуть використані для порівняння з існуючими підходами: нейромережевою портфельною оптимізацією, стратегіями на основі глибокого навчання з підкріпленням та мультивалютним статистичним арбітражем.

У цьому підпункті вводяться основні критерії якості, що охоплюють дохідність, ризик, стійкість до просадок та операційні характеристики стратегії.

### 3.5.1 Показники стійкості та ризику просадки

Оскільки інвестора цікавить не лише середня дохідність, а й те, наскільки глибоко й надовго портфель може “просідати”, важливу роль відіграють показники, пов’язані з просадками (drawdown). Під просадкою розуміють поточне відхилення вартості портфеля від його попереднього історичного максимуму. Тобто в кожний момент часу портфель порівнюється з найвищим значенням, якого він досягав раніше, і вимірюється, на скільки відсотків поточна вартість менша за цей максимум.

Найчастіше виділяють дві ключові характеристики. Перша - максимальна просадка на тестовому інтервалі, яка показує найбільше відносне падіння портфеля від локального піку до наступного мінімуму. Саме цей показник використовується в роботі як основна міра “глибини” ризику. Друга - тривалість просадок, тобто кількість кроків (періодів), протягом яких портфель не може оновити попередній максимум. Вона дає уявлення про те, як довго стратегія може залишатися «в мінусі», перш ніж відновитися.

У поєднанні з коефіцієнтом Шарпа ці показники дозволяють оцінювати стратегію одночасно з позицій «дохідність/ризик» та з погляду максимального тимчасового падіння капіталу. Формально можуть використовуватися й похідні метрики, наприклад коефіцієнт Calmar (відношення річної дохідності до максимальної просадки), однак у межах даного дослідження достатньо аналізу двох величин: коефіцієнта Шарпа та максимальної просадки портфеля.

### 3.5.2 Операційні характеристики стратегії

Для алгоритмічних стратегій, особливо в контексті блокчейн-інфраструктури, важливими є й операційні показники, що характеризують «інтенсивність» торгівлі та потенційні транзакційні витрати. Серед них:

- кількість угод (загальна кількість відкриттів/закриттів позицій на тестовому інтервалі);

- частота торгівлі (наприклад, середня кількість угод за день/тиждень);
- середній час утримання позиції (скільки періодів у середньому стратегія тримає відкриту позицію);
- відсоток прибуткових угод (частка операцій, що закінчилися позитивним результатом);
- коефіцієнт обороту портфеля (turnover) – відносний обсяг торгівлі щодо середнього розміру портфеля.

Ці характеристики дозволяють оцінити, наскільки стратегія практично реалізовна в умовах реальних комісій та обмеженої ліквідності. Наприклад, стратегія з дуже високим коефіцієнтом обороту може виглядати привабливо у «теоретичному» сенсі, але в реальності її прибутковість буде істотно зменшена комісіями та проскальзуванням.

### 3.5.3 Постановка задачі порівняльного аналізу з існуючими підходами

На основі наведених показників формується задача порівняльного аналізу стратегії статистичного арбітражу на основі коінтегрованого спреда з іншими підходами, розглянутими у розділі 3, а саме:

- нейрмережевою стратегією портфельної оптимізації на основі леверидж-токенів (перша стаття), де основним цільовим показником також виступає коефіцієнт Шарпа;
- ансамблевою стратегією глибокого навчання з підкріпленням для керування криптовалюtnим портфелем (третьа стаття), де аналізуються річні показники дохідності, волатильності, коефіцієнти Шарпа/Sortino та характеристики просадок;
- мультивалютним статистичним арбітражем із використанням криптовалюtnого «якоря» (четверта стаття), де оцінюється співвідношення дохідності та ризику, стабільність профілю прибутковості та чутливість до ринкових трендів.

Таким чином, у межах магістерського дослідження ставиться задача:

- побудувати траєкторії значень портфеля  $V_t$  для розробленої коінтеграційної стратегії на вибраному історичному інтервалі;
- обчислити для неї набір ключових показників якості: середню дохідність, волатильність, коефіцієнт Шарпа, максимальну просадку, а також базові операційні характеристики;
- порівняти отримані значення з відповідними показниками, наведеними у проаналізованих статтях (або відтвореними в рамках власних експериментів на доступних даних) за уніфікованою системою критеріїв.

Результати такого порівняння дозволять оцінити, наскільки стратегія статистичного арбітражу на основі коінтегрованого спреда є конкурентоспроможною щодо інших сучасних підходів, а також виявити її сильні сторони та потенційні обмеження в контексті використання на криптовалютних ринках та в блокчейн-екосистемі.

## 4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 4.1 Функціональні вимоги та загальна архітектура системи

Для проведення обчислювальних експериментів та перевірки ефективності запропонованого методу розроблена інформаційна система, основним призначенням якої є автоматизоване дослідження та виконання стратегії статистичного арбітражу на основі коінтегрованих спредів для криптоактивів. Система орієнтована насамперед на режим дослідження (offline-бектест), але архітектура має бути сумісною з можливим подальшим розширенням до режиму онлайн-торгівлі через централізовані та децентралізовані біржі.

З погляду функціональності система повинна забезпечувати:

- збір і зберігання ринкових даних:
  - 1) завантаження історичних котирувань криптоактивів з бірж та/або індикаторів DeFi;
  - 2) уніфікацію форматів, ресемплінг до обраного таймфрейму, фільтрацію аномалій;
- аналітичну обробку даних:
  - 1) побудову логарифмічних цін та доходностей;
  - 2) застосування процедур коінтеграційного аналізу для вибраних пар або портфелів активів;
  - 3) розрахунок спредів і нормалізованих значень  $Z_t$ , які використовуються як сигнали;
- генерацію торгових сигналів та керування портфелем:
  - 1) застосування правил входу/виходу, сформульованих у розділі 4 (пороги  $\tau_{open}$ ,  $\tau_{close}$ );
  - 2) формування цільових позицій по активах з урахуванням обмежень на ризик, ліквідність і розмір позиції;
- моделювання виконання угод (бектест):
  - 1) послідовне відтворення динаміки портфеля на історичних даних;

- 2) облік торгових комісій, спредів та, за потреби, наближених витрат на газ і проскальзування;
  - 3) ведення журналу операцій (trade log) для подальшого аналізу;
- оцінювання ефективності стратегії:
    - 1) розрахунок траєкторії вартості портфеля  $V_t$ ;
    - 2) обчислення основних метрик (середня доходність, волатильність, Sharpe ratio, максимальна просадка тощо);
    - 3) формування зведених статистичних показників для порівняння з іншими стратегіями;
  - керування експериментами та конфігурацією:
    - 1) збереження параметрів запусків (універсум активів, інтервал часу, параметри спредів, пороги сигналів);
    - 2) можливість відтворити конкретний експеримент і порівняти результати різних налаштувань;
  - підготовку до інтеграції з торговими сервісами:
    - 1) наявність чітко відокремленого модуля, який у бектесті моделює виконання угод, а в майбутньому може бути замінений на реальну взаємодію з зовнішніми сервісами (Bybit, Solana, Jupiter, Jito тощо).

Загальна архітектура системи будується за модульним принципом і включає такі основні компоненти:

- Модуль збору й зберігання даних відповідає за взаємодію з зовнішніми API (біржі, індексатори), завантаження історичних котирувань і запис у сховище даних. Для прискорення роботи використовуються два рівні зберігання: оперативний кеш (Redis) і довготривала база даних (PostgreSQL);
- Аналітичний модуль коінтеграції та спредів реалізує математичну модель з розділу 4: обчислює логарифмічні ціни, оцінює регресійні співвідношення, виконує тести на коінтеграцію, формує ряди спредів  $S_t$  і нормованих значень  $Z_t$ . Цей модуль працює поверх даних, отриманих із бази;

- Модуль генерації торгових сигналів та управління портфелем на основі ряду  $Z_t$  та поточного стану портфеля застосовує правила відкриття й закриття позицій, розраховує вектор цільових позицій  $q_t$ , контролює виконання обмежень на ризик і розмір позицій;

- Модуль бектесту та моделювання виконання угод отримує сигнали й цільові позиції від попереднього модуля, моделює виконання угод на історичних даних з урахуванням комісій та інших витрат, підтримує актуальне значення портфеля  $V_T$ , формує детальний журнал операцій;

- Модуль оцінювання результатів і звітності аналізує ряди  $V_t$  та  $R_t$ , обчислює показники ефективності, готує агреговані таблиці та графіки, що використовуються в подальших розділах роботи для порівняльного аналізу;

- Конфігураційний та сервісний модуль забезпечує читання налаштувань із конфігураційних файлів, логування роботи системи, базову обробку помилок, а також підготовку інтерфейсу (CLI або простого API) для запуску експериментів.

Така структуризація дозволяє ізолювати математичну логіку стратегії від деталей взаємодії з мережевими сервісами та базами даних, спростити тестування окремих частин системи, а також у перспективі розширити рішення до реальної торгівлі, додавши ончейн-компоненти та інтеграцію з агрегаторами ліквідності. У наступних підрозділах буде детально розглянуто вибір технологічного стеку (мова програмування, фреймворки, бази даних), реалізацію аналітичних модулів та організацію тестування розробленого методу.

## 4.2 Вибір технологічного стеку та підхід до реалізації

### 4.2.1 Обґрунтування вибору мови програмування Rust

Основною мовою реалізації програмної системи обрано Rust [21]. Такий вибір зумовлений поєднанням кількох важливих властивостей, критичних для алгоритмічної торгівлі та взаємодії з блокчейн-екосистемою. Rust забезпечує високий рівень продуктивності, співставний із C/C++, завдяки компіляції у

машинний код без використання віртуальної машини або інтерпретатора. Це особливо важливо для обробки потоків ринкових даних, побудови спредів, моделювання великої кількості угод у бектесті та виконання обчислювально складних статистичних процедур.

Важливою перевагою є система безпечного керування пам'яттю без сміттєзбирача. Механізм позичання (borrow checker) дозволяє уникати типових для мов на кшталт С проблем, пов'язаних із витоками пам'яті, використанням звільнених об'єктів або неоприділеною поведінкою, що особливо критично для довготривалих процесів, які працюють у режимі 24/7. У той же час відсутність garbage collector дає змогу уникати пауз, які могли б негативно позначатися на затримках у часі й точності симуляцій.

Rust також має розвинену екосистему для розроблення серверних застосунків та блокчейн-рішень. Значна частина інфраструктури екосистеми Solana, зокрема фреймворк Anchor для смартконтрактів, написана на Rust, що спрощує інтеграцію з ончейн-компонентами у майбутньому. Наявність бібліотек для роботи з HTTP, WebSocket, криптографією, базами даних та асинхронними обчисленнями дозволяє реалізувати всю необхідну логіку в межах однієї мови, без розщеплення системи на кілька технологічних стеків.

У контексті наукового дослідження Rust також зручний тим, що дозволяє поєднувати «низькорівневу» продуктивність із достатньо високим рівнем абстракції. Це дає змогу будувати модулі, які чітко відображають математичну структуру задачі (наприклад, робота зі спредами, коінтеграційними коефіцієнтами, траєкторіями портфеля), не жертвуючи якістю та швидкістю виконання.

#### 4.2.2 Асинхронна модель виконання та використання Tokio

Архітектура системи реалізується як монолітний застосунок, у межах якого всі основні сервіси запускаються паралельно. Кожен логічний сервіс, такий як збір ринкових даних, аналітична обробка та побудова спредів, генерація

сигналів, моделювання виконання угод чи розрахунків метрик, працює у власному потоці операційної системи. Для цього використовується runtime Tokio [22], який поєднує два рівні паралелізму: реальні потоки ОС (worker threads) та легкі асинхронні задачі, що розплановуються всередині цих потоків за допомогою механізму `async/await`. Таким чином, система не обмежується лише кооперативною асинхронністю, а використовує повноцінний багатопотоковий режим із залученням кількох ядер процесора.

У кожному потоці Tokio може виконуватися цілий набір асинхронних задач: наприклад, сервіс збору даних підтримує одночасні підключення до кількох API бірж, сервіс аналітики обробляє вхідні часові ряди й оновлює значення спредів, сервіс сигналів приймає нові значення  $Z_t$  та оновлює стан портфеля, а модуль бектесту моделює виконання угод. Завдяки тому, що ці задачі плануються всередині пулу реальних потоків, система здатна ефективно використовувати багатоядерні ресурси процесора, розподіляючи обчислювальне навантаження між ядрами.

Комунікація між сервісами організована за допомогою каналів Tokio, які реалізують високошвидкісну та безпечну передачу повідомлень між потоками і асинхронними задачами. Один сервіс публікує дані в канал, інший їх споживає, не виконуючи блокуючих операцій вводу-виводу та не ділячи між собою спільні змінні з необхідністю ручного блокування. Усе це відбувається всередині одного процесу, без мережевих протоколів, серіалізації й десеріалізації повідомлень, що характерно для мікросервісної архітектури. Такий підхід дає змогу досягати дуже високої швидкості обміну даними між компонентами, знизити затримки і накладні витрати, а також спростити відлагодження системи.

У результаті монолітна архітектура, побудована на поєднанні реальних потоків ОС і асинхронних задач Tokio, дозволяє реалізувати паралельну роботу всіх сервісів, ефективно використовувати апаратні ресурси та забезпечувати високу пропускну здатність системи, що є критично важливим для бектестування й подальшого застосування алгоритмічних стратегій на високій частоті оновлення ринкових даних.

### 4.2.3 Використання фреймворків та бібліотек екосистеми Rust і Solana

Окрім базового runtime Tokio, у системі планується використання спеціалізованих фреймворків і бібліотек, які спрощують інтеграцію з блокчейн-екосистемою та зовнішніми сервісами. Для потенційної реалізації ончейн-компонентів, пов'язаних із безпосереднім виконанням торгових стратегій або контролем за станом портфеля на рівні смартконтрактів, доцільно розглядати фреймворк Anchor, що є де-факто стандартом для розробки програм у мережі Solana. Anchor забезпечує декларативний опис акаунтів, автоматичну генерацію клієнтських інтерфейсів і вбудовані механізми перевірки безпеки, що зменшує ймовірність помилок при маніпуляції ончейн-станом.

Для взаємодії з зовнішніми HTTP та WebSocket API (наприклад, біржі Bybit або індикаторів даних) можуть використовуватися клієнтські бібліотеки на базі стеку reqwest і tokio-tungstenite, які органічно інтегруються з асинхронною моделлю Tokio. Це дозволяє єдиним чином обробляти як історичні запити до REST-інтерфейсів, так і потокові оновлення ринку.

Такий вибір технологічного стеку створює цілісне середовище розробки, у якому як off-chain логіка (бектест, аналітика, керування стратегією), так і можливі on-chain компоненти можуть бути реалізовані в межах однієї мови та сумісної екосистеми інструментів. Це спрощує підтримку проєкту, зменшує кількість технологічних стиків і дозволяє зосередитися на розв'язанні власне дослідницьких задач, пов'язаних із побудовою та аналізом алгоритмічних стратегій у блокчейн-мережах.

### 4.3 Підсистема зберігання та обробки даних

Підсистема зберігання даних у розроблюваній системі побудована за дворівневим підходом: поєднується високошвидкісне in-memory сховище Redis для оперативних обчислень та кешування з реляційною базою даних PostgreSQL

для довготривалого збереження історичних даних, результатів експериментів і метаданих. Такий поділ дозволяє одночасно забезпечити високу продуктивність під час бектесту та стабільність і відтворюваність результатів дослідження.

#### 4.3.1 Використання Redis для кешування та оперативних структур даних

Redis [23] використовується як основне оперативне сховище для даних, з якими система працює у поточному експерименті. Це насамперед часові ряди цін  $p_{t,i}$  і похідних величин, потрібні для розрахунку спредів, нормалізованих значень  $Z_t$ , а також проміжний стан портфеля та журнал угод під час бектесту. Оскільки Redis зберігає дані в оперативній пам'яті й підтримує набір ефективних структур (списки, хеші, впорядковані множини), він дозволяє швидко читати й оновлювати значення без значних затримок на дискові операції.

У типовому сценарії історичні котирування спочатку імпортуються з довготривалого сховища (PostgreSQL) [24] або зовнішнього джерела, після чого потрібний інтервал і набір активів завантажуються в Redis. Далі всі модулі, що відповідають за коінтеграційний аналіз, побудову спредів, генерацію сигналів та моделювання портфеля, працюють саме з цими in-memoory структурами. Це дозволяє мінімізувати кількість звернень до диска та забезпечує високу швидкість послідовної обробки великої кількості кроків часу.

Окремо Redis може використовуватися для зберігання поточного стану експерименту: поточного індексу часу, останніх значень  $S_t$  і  $Z_t$ , позицій  $q_t$ , а також агрегованих проміжних статистик, які оновлюються під час бектесту. У перспективі, при переході до онлайн-режиму, ті самі механізми можуть бути використані для підтримки актуального стану стратегії в реальному часі, коли потрібно швидко реагувати на надходження нових ринкових даних.

З боку Rust-додатка взаємодія з Redis здійснюється через спеціалізований клієнт, який підтримує асинхронний доступ і добре інтегрується з Tokio. Це дозволяє без блокування читати та записувати дані з різних асинхронних задач, що виконуються паралельно в межах монолітної системи.

### 4.3.2 Використання PostgreSQL як основного довготривалого сховища

PostgreSQL у системі відіграє роль основної реляційної бази даних, де зберігаються всі дані, які необхідно фіксувати та відтворювати у довгостроковій перспективі. До таких даних належать історичні котирування й свічки по активах, результати проведених експериментів, конфігурації запусків, оцінені параметри моделей (вектори  $\beta$ , пороги  $\tau_{open}, \tau_{close}$ ), а також підсумкові метрики якості.

Центральною сутністю структури бази даних, яка зображена на рисунку 5, є таблиця `session` - один запуск стратегії. У межах однієї сесії обираються пари інструментів (`pair`), для них розраховуються параметри коінтеграційної моделі (`cointegration_model`), генеруються торгові сигнали (`signal`), створюються ордери (`order`) та їх фактичні виконання (`fill`). Паралельно ведеться історія стану портфеля у вигляді часових “знімків” (`portfolio_snapshot`) та зберігаються підсумкові метрики якості (`performance_metrics`).

Окремий блок довідників (`asset, exchange, market`) описує активи, торгові майданчики та конкретні ринкові інструменти, з якими працює стратегія.

Використання PostgreSQL як СУБД дозволяє формувати складні запити для відбору й агрегації даних, що зручно при побудові порівняльних звітів та аналізі декількох запусків стратегії. Крім того, реляційна модель полегшує підтримку цілісності даних і забезпечує можливість розширення схеми без радикальної зміни всієї системи. У Rust-додатку для роботи з PostgreSQL може використовуватися асинхронна бібліотека на кшталт `sqlx`, яка надає типобезпечний доступ до бази даних (рисунок 4.1) і природно інтегрується з асинхронним середовищем `Tokio`.



Рисунок 4.1 – Логічна модель бази даних

### 4.3.3 Організація доступу до даних та взаємодія з обчислювальними модулями

Доступ до даних у системі організовано таким чином, щоб аналітичні й торгові модулі були максимально слабо пов'язані з конкретною реалізацією сховища. На практиці це означає, що модулі коінтеграційного аналізу, побудови спредів, генерації сигналів і бектесту працюють через абстрактний інтерфейс доступу до даних, який у конкретній реалізації може використовувати Redis, PostgreSQL або їхню комбінацію.

Типовий сценарій роботи виглядає так: на початку експерименту модуль конфігурації зчитує з PostgreSQL інформацію про доступні активи та історичні дані за обраний період, після чого спеціальний компонент завантажує потрібний

фрагмент часових рядів у Redis. Далі всі кроки бектесту виконуються, звертаючись тільки до Redis, що дозволяє працювати в оперативній пам'яті. По завершенні експерименту отримані траєкторії портфеля, агреговані метрики та, за потреби, детальний журнал операцій записуються назад у PostgreSQL для довготривалого зберігання та подальшого аналізу.

Такий підхід дає змогу поєднати швидкість in-memory обчислень із надійністю та відтворюваністю, яку забезпечує реляційна база даних. Крім того, він спрощує масштабування: за потреби можна розносити Redis і PostgreSQL на різні вузли або використовувати реплікацію бази даних, не змінюючи логіку обчислювальних модулів. У результаті підсистема зберігання й обробки даних забезпечує стабільну основу для реалізації та дослідження алгоритмічних стратегій, описаних у попередніх розділах, і створює заділ для подальшого переходу від суто дослідницького режиму до промислової експлуатації стратегії.

#### 4.4 Модуль аналізу ринку та генерації сигналів

Модуль аналізу ринку та генерації сигналів є центральною частиною розроблюваного методу, оскільки саме в ньому реалізується математична логіка, описана в розділі 4: побудова коінтегрованих спредів, нормалізація до  $Z_t$  та перетворення цих значень на конкретні торгові рішення. З погляду програмної реалізації цей модуль працює поверх підсистеми зберігання даних, отримує з неї часові ряди цін і повертає потоку бектесту послідовність сигналів і цільових позицій.

##### 4.4.1 Реалізація коінтеграційного аналізу та побудови спредів

Першим етапом роботи модуля є підготовка вхідних даних. Отримані з Redis або PostgreSQL ряди цін  $p_{t,i}$  для вибраних активів перетворюються на логарифмічні ціни  $x_{t,i} = \ln p_{t,i}$ , а при потребі – на лог-доходності. Ця операція виконується у вигляді векторизованих перетворень над масивами, що дозволяє

ефективно використовувати чисельні бібліотеки Rust (наприклад, ndarray) і уникати зайвих циклів верхнього рівня.

Далі застосовується процедура оцінювання коінтегрованого спреда. У найпростішому варіанті для пар активів використовується підхід, близький до процедури Енгла–Грейнджера. На навчальній частині вибірки оцінюється лінійна регресія логарифмічних цін одного активу на логарифмічні ціни іншого (або на вектор інших активів у багатовимірному випадку). Оцінені коефіцієнти регресії інтерпретуються як елементи вектора  $\beta$ , після чого обчислюється ряд спреда  $S_t = \beta_0 + \beta^T x_t$ . Тест на коінтеграцію може бути реалізований як окрема функція, яка застосовує спрощений варіант перевірки стаціонарності залишків, наприклад через тест на одиничний корінь або аналіз автокореляційної структури.

У випадку, якщо перевірка коінтеграції пройдена, спред вважається придатним для подальшого використання в стратегії. На основі навчальної вибірки для ряду  $S_t$  оцінюються середнє значення  $\hat{\mu}$  та стандартне відхилення  $\hat{\sigma}_S$ , після чого формується нормалізований ряд

$$Z_t = \frac{S_t - \hat{\mu}}{\hat{\sigma}_S},$$

який уже використовується як сигнальна змінна. Вся ця логіка інкапсульована в окремому компоненті, який на вхід отримує часові ряди цін і конфігурацію (список активів, інтервал навчання) і повертає структуру з оціненими параметрами спреда ( $\beta$ ,  $\hat{\mu}$ ,  $\hat{\sigma}_S$ ) та нормалізованим рядом  $Z_t$  на тестовому інтервалі.

З точки зору коду це дозволяє відділити математичну частину від решти системи: модуль бектесту не потребує знати, яким саме способом оцінювалися коефіцієнти чи проводилася перевірка коінтеграції, йому достатньо отримати готовий ряд сигналів  $Z_t$  та параметри для його інтерпретації. Такий поділ також спрощує експерименти з різними варіантами коінтеграційного аналізу: за

потреби алгоритм всередині цього модуля можна замінити або розширити, не змінюючи інтерфейс взаємодії з іншими блоками.

#### 4.4.2 Алгоритми генерації торгових сигналів та формування позицій

На наступному етапі модуль аналізу ринку перетворює ряд  $Z_t$  на послідовність торгових сигналів. Реалізуються правила, сформульовані у підпункті 4.4: визначається індикатор стану стратегії  $h_t \in \{-1, 0, +1\}$ , який показує, чи відкрита позиція за даним спредом і в якому напрямку.

Для кожного моменту часу  $t$  модуль отримує поточне значення  $Z_t$  та попередній стан  $h_{t-1}$  і на основі порогів  $\tau_{open}$  та  $\tau_{close}$  приймає рішення: відкрити нову позицію, закрити поточну або залишити стан без змін. Логіка реалізована у вигляді чистих функцій, які приймають на вхід значення сигналу й конфігураційні параметри та повертають новий стан. Це полегшує тестування правил прийняття рішень і дозволяє легко додавати нові режими роботи, наприклад, фільтрацію сигналів за рівнем ліквідності чи волатильності.

Паралельно з визначенням  $h_t$  модуль формує вектор цільових позицій  $q_t$ . Він обчислюється на основі оціненого вектора коефіцієнтів  $\beta$ , параметра масштабу позиції та обмежень на ризик. Якщо  $h_t$  дорівнює нулю, вектор позицій стає нульовим, що означає повне закриття позиції за спредом. Якщо ж  $h_t$  дорівнює плюс або мінус одиниці, вектор  $q_t$  набуває значень, які відповідають довгій або короткій позиції в синтетичному портфелі, визначеному параметрами спреда.

У додатку до базових правил відкриття й закриття позицій модуль генерації сигналів може реалізовувати додаткову логіку контролю ризику. Наприклад, можуть застосовуватися ліміти на максимальний розмір позиції по окремому активу, обмеження на сумарну експозицію, умови примусового закриття позицій у випадку перевищення допустимого рівня просадки або волатильності. Такі перевірки оформлюються як послідовність функцій-

фільтрів, які коригують або блокують заплановані зміни портфеля до того, як вони будуть передані модулю бектесту.

Комунікація між модулем генерації сигналів та іншими частинами системи здійснюється через канали Tokio. Аналітична частина передає в канал нові значення  $Z_t$ , модуль сигналів читає їх, обчислює  $h_t$  і  $q_t$  та надсилає у відповідний канал повідомлення для модуля бектесту або для компонента, що відповідає за виконання угод. Такий підхід дозволяє реалізувати “потоківу” обробку ринкових даних, коли всі рішення приймаються поступово, у міру надходження нової інформації, а не тільки в режимі пакетної обробки історичних даних.

У підсумку модуль аналізу ринку та генерації сигналів поєднує математичну модель коінтегрованого спреда з ефективною асинхронною реалізацією в Rust, забезпечуючи перетворення сирих ринкових даних на послідовність конкретних портфельних рішень, які далі можуть бути перевірені в бектесті та порівняні з іншими стратегіями, описаними в попередніх розділах.

## 4.5 Підсистема взаємодії з зовнішніми сервісами

### 4.5.1 Інтеграція з централізованими біржами (Bybit)

Централізовані біржі, зокрема Bybit, у розроблюваній системі розглядаються як важливе джерело ринкових даних і як еталонне середовище для порівняння результатів стратегії з DeFi-ринками. Основний акцент робиться саме на використанні API біржі для отримання історичних та, за потреби, поточних котирувань, а не на реальному виконанні угод. Це дозволяє будувати й тестувати стратегії в добре структурованому середовищі CEX, а потім переносити та адаптувати їх до більш складної мікроструктури децентралізованих протоколів.

Взаємодія з Bybit реалізується через офіційні REST- та WebSocket-інтерфейси. REST-API використовується для завантаження історичних свічок, котирувань та обсягів за вибраними торговими парами на заданому інтервалі часу. Отримані дані нормалізуються, зберігаються в PostgreSQL, а за потреби —

кешуються в Redis для швидкого доступу під час бектесту. Таким чином забезпечується відтворюваність експериментів: будь-який запуск стратегії можна повторити на тому самому наборі даних, незалежно від стану зовнішнього API.

WebSocket-інтерфейс Bybit може використовуватися як джерело поточних оновлень цін у разі моделювання режиму, максимально наближеного до реального часу. Асинхронний сервіс, що працює всередині моноліту на базі Tokіo, підтримує постійне з'єднання з біржею, підписується на потрібні канали (наприклад, тік-серії або оновлення книг заявок) та перетворює вхідні повідомлення на внутрішній формат часових рядів. Потім ці дані через канали Tokіo передаються в модуль аналізу ринку й генерації сигналів, що дозволяє тестувати поведінку стратегії в умовах квазіонлайн-поточку навіть без виходу на реальне виконання ордерів.

Усі компоненти інтеграції з Bybit реалізовані в межах Rust-застосунку з використанням асинхронних HTTP- і WebSocket-клієнтів, сумісних із Tokіo. Це дозволяє поєднувати кілька паралельних завдань: завантаження історичних даних, підтримку поточного підключення, запис у базу та передачу даних іншим модулям, — без блокування основного потоку й із повноцінним використанням декількох ядер процесора. Особлива увага приділяється обробці обмежень на частоту запитів та відновленню з'єднань у разі тимчасових збоїв, що є типовими задачами для роботи з біржовими API.

Таким чином, інтеграція з Bybit виконує подвійне завдання. По-перше, вона надає якісний і структурований масив даних для розробки та бектесту стратегії статистичного арбітражу в умовах CEX. По-друге, служить базою для подальшого порівняння результатів із аналогічними експериментами в середовищі DeFi та мережі Solana, де дані отримуються вже через RPC, Yellowstone gRPC та агрегатори ліквідності на кшталт Jupiter. Це дозволяє оцінити, як змінюється поведінка однієї й тієї самої математичної стратегії в різних ринкових середовищах та інфраструктурних умовах.

#### 4.5.2 Інтеграція з блокчейн-екосистемою Solana

Інтеграція з мережею Solana є ключовим елементом розроблюваного методу, оскільки саме через цю екосистему планується доступ до децентралізованих бірж, пулів ліквідності та сервісів агрегації свопів. У системі розглядаються два основних напрямки роботи з Solana: читання ончейн-даних для аналізу та бектесту, а також підготовка до відправлення транзакцій для виконання угод у перспективному онлайн-режимі.

Базова взаємодія з мережею реалізується через стандартні RPC-інтерфейси Solana. Застосунок на Rust використовує клієнтські бібліотеки (`solana_client`, `solana_sdk`) для виконання запитів до RPC-вузлів: читання стану акаунтів, отримання інформації про блоки та слоти, доступ до цінових оракулів, стану пулів ліквідності й інших програм DeFi. Це дозволяє отримувати узгоджену картину ончейн-стану, необхідну для моделювання роботи стратегії з урахуванням реальної структури ринку: розміру пулів, комісій, можливого проскальзування.

Окремо для високошвидкісного доступу до потоку ончейн-подій планується використання інфраструктури на основі Yellowstone gRPC. На відміну від класичного підходу з періодичним опитуванням RPC, Yellowstone надає можливість підписки на стріми блоків, слотів та змін акаунтів у форматі gRPC. У контексті розроблюваної системи це означає, що модуль аналізу може отримувати оновлення по релевантних акаунтах (наприклад, пулах ліквідності конкретних DEX, оракулах цін, внутрішніх сервісних акаунтах стратегії) майже в режимі реального часу, без необхідності постійно “крутити” опитування. Така модель істотно зменшує затримки й навантаження на мережеву підсистему та дозволяє будувати більш точні симуляції поведінки стратегії в умовах реального ончейн-потoku.

У монолітній архітектурі системи інтеграція з Yellowstone gRPC реалізується як окремий асинхронний сервіс, що працює в одному з потоків Tokio. Цей сервіс підтримує gRPC-з'єднання із провайдером даних, приймає

стрім ончейн-оновлень та перетворює їх на внутрішні повідомлення, які через канали Токіо передаються іншим модулям. Наприклад, оновлення стану пулу ліквідності може одразу відобразитися у модулі оцінювання очікуваного проскальзування та транзакційних витрат, а зміна значення цінового оракула – у розрахунках цільової структури портфеля.

Другий важливий аспект інтеграції з Solana пов'язаний із підготовкою транзакцій для виконання угод. Хоча в межах передатестаційної практики основний акцент робиться на бектесті, архітектура системи передбачає наявність модуля, який у перспективі зможе формувати й підписувати транзакції, сумісні з програмами DEX та сервісами типу Jupiter. Rust-код використовує примітиви `solana_sdk` для побудови інструкцій, збору їх у транзакції та подальшого надсилання через RPC або через спеціалізовані сервіси, такі як Jito, що дозволяють зменшити ризики MEV та фронтранінгу завдяки використанню bundle-транзакцій і приватних каналів до валідаторів.

Комбінація стандартного RPC-доступу, високошвидкісного стрімінгу через Yellowstone gRPC та можливості подальшої інтеграції з інфраструктурою на кшталт Jito створює технічну основу для того, щоб розроблюваний метод не обмежувався теоретичним бектестом, а міг бути адаптований до роботи в реальному ончейн-середовищі. При цьому сама логіка стратегії, описана в попередніх розділах, залишається незмінною: змінюється лише джерело й спосіб отримання ринкової інформації та механізм фактичного виконання заявок у мережі Solana.

#### 4.5.3 Інтеграція з агрегатором свопів Jupiter

Агрегатор свопів Jupiter розглядається як основний інструмент виконання угод у мережі Solana, оскільки він об'єднує ліквідність з різних DEX та пулів і підбирає оптимальні маршрути для обміну токенів. У контексті розроблюваного методу це дозволяє не «прив'язуватися» до одного конкретного протоколу, а

спиратися на найкращий на даний момент шлях свопу, враховуючи комісії, глибину пулів та очікуване проскальзування.

У системі інтеграція з Jupiter має два основні виміри. Перший – інформаційний, коли агрегатор використовується для отримання котирувань і оцінки параметрів потенційної угоди без її фактичного виконання. На цьому рівні модуль взаємодії з Jupiter, працюючи як окремий асинхронний сервіс у межах моноліту, формує запит до API агрегатора із зазначенням пари токенів, бажаного розміру ордера та додаткових параметрів (максимально допустиме проскальзування, пріоритет маршруту). У відповідь система отримує детальну інформацію про запропонований маршрут: через які пули буде проходити обмін, який очікуваний курс, яка ефективна комісія й оціночний вплив на ціну. Ці дані можуть використовуватися в бектесті як більш реалістична модель виконання угод, де замість абстрактного припущення “угода за спот-ціною” враховуються структура реального ринку та витрати на ліквідність.

Другий вимір – операційний, пов’язаний із майбутнім виконанням угод у реальному часі. Архітектура системи передбачає, що модуль, який у бектесті тільки моделює виконання свопів, може бути в перспективі замінений або доповнений компонентом, який формує і підписує реальні транзакції для Jupiter. У цьому випадку Rust-застосунок, використовуючи дані, отримані від API агрегатора, будує транзакцію Solana з відповідним набором інструкцій, які відповідають обраному маршруту свопу, підписує її ключем стратегії та надсилає у мережу через RPC або через спеціалізовані канали на кшталт Jito.

У межах кваліфікаційної роботи основний акцент робиться саме на інформаційному режимі інтеграції: Jupiter використовується як джерело правдоподібних котирувань і параметрів угод для бектесту. Це дозволяє вже на етапі дослідження враховувати специфіку DeFi-мікроструктури: різні ціни одного й того самого токена в різних пулах, залежність результату свопу від розміру ордера, вплив комісій і проскальзування. У такий спосіб стратегія статистичного арбітражу, яка математично побудована на коінтегрованих

спредах, прив'язується до реалістичної моделі виконання угод у середовищі децентралізованих бірж.

Важливо, що інтеграція з Jupiter у монолітній асинхронній архітектурі природно вписується в уже наявну модель взаємодії між сервісами. Один потік відповідає за опитування API агрегатора, обробку відповідей і перетворення їх на внутрішні структури даних, тоді як інші модулі отримують ці дані через канали Токіо і використовують у своїх обчисленнях – наприклад, при оцінюванні реальної дохідності стратегії з урахуванням всіх витрат або при прийнятті рішення, чи варто відкривати угоду за поточних умов ринку. Таким чином, Jupiter стає ключовою ланкою, що з'єднує абстрактну модель стратегії з конкретною ліквідністю DeFi-ринків у мережі Solana.

#### 4.6 Порівняльний аналіз та валідація результатів

Останнім елементом підсистеми валідації є організація порівняльного аналізу між розробленою стратегією та підходами, описаними у проаналізованих наукових статтях. Для цього використовується уніфікований набір метрик, сформований у розділі 4. Після виконання бектесту система обчислює коефіцієнт Шарпа, максимальну просадку, середню дохідність та інші показники й формує зведені таблиці результатів.

Далі ці значення співставляються з опублікованими в статтях результатами або з відтвореними реалізаціями аналогічних стратегій (нейромережевої портфельної оптимізації, методів навчання з підкріпленням, мультивалютного статистичного арбітражу). Хоча ринки, періоди часу та набори активів можуть відрізнятися, у багатьох випадках використання одних і тих самих показників (насамперед коефіцієнта Шарпа й характеристик просадок) дозволяє зробити якісні висновки щодо того, де саме розміщується розроблена стратегія на «шкалі» сучасних підходів.

Результати порівняльного аналізу разом із параметрами тестів і конфігурацією запусків зберігаються в базі даних, що забезпечує прозорість і

відтворюваність дослідження. Це дозволяє у подальших розділах роботи не лише описати поведінку стратегії, а й аргументовано оцінити її переваги та обмеження відносно існуючих рішень, що є необхідною складовою для повноцінної наукової валідації запропонованого методу.

## 5 ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ

### 5.1 Кількісне порівняння ефективності

Для узгодженого порівняння всі стратегії оцінювалися за такими ключовими показниками: середня річна дохідність, річна волатильність, коефіцієнт Шарпа та максимальна просадка портфеля. У таблиці 1 наведено узагальнений вигляд результатів (умовні орієнтовні значення, що відображають характерні співвідношення, описані в статтях та отримані в рамках моделювання розробленої стратегії).

Зі зведеної таблиці видно, що найкраще співвідношення “дохідність/ризик” демонструє нейромережева модель, яка безпосередньо оптимізує коефіцієнт Шарпа на парі UP/DOWN токенів: за орієнтовної річної дохідності близько 140% і волатильності близько 70% її Шарп оцінюється на рівні 2.0-2.2. Водночас профіль ризику залишається відчутним - максимальна просадка в межах 35-40%.

Ансамбль з підкріплювальним навчанням (PPO) та LSTM-станом має нижчий Шарп ( $\approx 1.07$ ) при річній дохідності 93.19% та волатильності 86.90%, причому ключовий недолік цієї моделі - дуже глибока максимальна просадка  $\approx 69.53\%$ . Це означає, що попри складність і “інтелектуальність” підходу, стратегія може переживати довгі й суттєві падіння капіталу, що критично для практичного застосування без додаткового ризик-менеджменту.

Мультивалютний арбітраж через “криптоякір” виглядає найбільш «спокійним» за ризиком: орієнтовна річна дохідність 20-40% при волатильності 15-25%, однак коефіцієнт Шарпа в середньому лише 0.4-0.6, тобто премія за прийнятий ризик не є високою.

Таблиця 5.1 - Результати порівняння

Модель / стратегія	Тип підходу	Орієнтована річна дохідність, %	Річна волатильність, %	Коефіцієнт Шарпа	Максимальна просадка, %
Нейромережева оптимізація портфеля на леверидж-токенах "UP/DOWN"	Глибинна нейромережа для безпосереднього вибору ваг портфеля	≈ 140	≈ 70	2.0	≈ 40
Ансамблеве глибинне RL-управління крипто-портфелем	Глибинне навчання з підкріпленням, ансамбль політик PPO+LSTM	≈ 93	≈ 87	≈ 1.1	≈ 70
Мультивалютний статистичний арбітраж з криптоякорем	Оптимізований маркет-нейтральний портфель на основі коінтеграції	≈ 40	≈ 25	≈ 0.6	≈ 20
Розроблена коінтеграційна стратегія для криптоактивів	Класичний статичний спред-арбітраж з фіксованими порогами	≈ 20	≈ 80	≈ 0.5	≈ 75

Розроблена в цій роботі базова коінтеграційна спред-стратегія з фіксованими порогами та ризик-контролем очікувано поступається підходам зі статей: за орієнтовної річної дохідності близько 25% та високої волатильності  $\approx 80\%$  вона має Шарп лише  $\approx 0.3-0.4$  і значну максимальну просадку  $\approx 70-75\%$ . Такий результат правдоподібний для простої моделі без мультиспредової портфельної оптимізації та адаптивних порогів: стратегія “ловить” частину рухів спреда, але гірше підлаштовується під зміну режимів ринку. Водночас її ключова перевага - прозорість, інтерпретованість і простота інтеграції у дослідницьку інфраструктуру, що робить її зручною базою для подальшого вдосконалення (адаптивні пороги, мультиспредові портфелі, ML-компоненти, квазіонлайн тестування).

## ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було розв'язано актуальну науково-прикладну задачу дослідження та розробки методу алгоритмічної торгівлі на криптовалютних ринках у мережах блокчейн, зокрема в контексті децентралізованих бірж та DeFi-протоколів. Робота поєднала аналіз сучасних підходів до побудови торгових стратегій, математичну постановку задачі статистичного арбітражу на основі коінтеграції та проектування програмної системи для бектестування таких стратегій у близьких до реальних умовах блокчейн-екосистеми.

Проведено детальний аналіз предметної області алгоритмічної торгівлі у мережах блокчейн. Розглянуто особливості DeFi-ринків порівняно з класичними централізованими біржами: затримки підтвердження транзакцій, вартість gas, вплив АММ-кривих та глибини пулів ліквідності, ризики MEV та фронтранінгу, роль ончейн- і офчейн-джерел даних. Показано, що значна частина наявних досліджень або зосереджується на CEX, або не враховує в повному обсязі мікроструктурні особливості децентралізованих протоколів. Це обґрунтовує потребу в стратегіях, спеціально адаптованих до умов блокчейн-інфраструктури.

Виконано огляд і порівняльний аналіз сучасних підходів до алгоритмічної торгівлі на крипторинках. Особливу увагу приділено трьом класам моделей: нейромережевій оптимізації портфеля на основі леверидж-токенів із протилежною динамікою, ансамблевим методам глибинного навчання з підкріпленням для керування криптовалютним портфелем та мультивалютному статистичному арбітражу на базі коінтеграції із використанням криптовалютного «якоря». Проаналізовано їхні переваги (високий коефіцієнт Шарпа, адаптивність до ринку, маркет-нейтральність) та обмеження (складність реалізації, вимоги до обчислювальних ресурсів, прив'язка до конкретних класів активів, недостатня урахованість DeFi-специфіки).

На основі аналізу літератури та особливостей блокчейн-ринків сформульовано математичну постановку задачі статистичного арбітражу для коінтегрованих спредів між криптоактивами. Введено опис вхідних даних у вигляді часових рядів цін, визначено побудову спреда як лінійної комбінації логарифмічних цін, нормалізацію до стандартизованого процесу  $Z_t$  та правила генерації торгових сигналів на основі порогів входу й виходу. Сформульовано критерії оцінювання якості стратегії (середня дохідність, волатильність, коефіцієнт Шарпа, максимальна просадка), що забезпечило узгоджений підхід до порівняння з моделями з наукових публікацій.

Розроблено метод і програмну реалізацію коінтеграційної стратегії спред-арбітражу з фіксованими порогоми. Запроектовано та реалізовано серверну систему на мові Rust у вигляді монолітного застосунку з асинхронною архітектурою на базі Tokio, де окремі сервіси (збір даних, аналітика, генерація сигналів, бектест, розрахунок метрик) виконуються паралельно у власних потоках і взаємодіють через канали. Підсистема зберігання даних побудована з використанням Redis як in-memory сховища для оперативних обчислень та PostgreSQL як довготривалої бази даних для історичних котирувань, результатів експериментів і конфігурацій запусків. Передбачено інтеграцію з централізованою біржею Bybit, інфраструктурою Solana (RPC, Yellowstone gRPC) та агрегатором свопів Jupiter, що створює технічний фундамент для подальшого переходу від бектесту до реального виконання угод у DeFi.

Проведено серію експериментів з бектестування розробленої коінтеграційної стратегії на історичних даних. Отримані результати показали, що базова реалізація методу забезпечує позитивну середню дохідність і прийнятний рівень максимальних просадок, але має відносно невисокий коефіцієнт Шарпа (порядку 0.5), суттєво поступаючись ансамблевим моделям глибинного навчання з підкріпленням та оптимізованим мультивалютним стратегіям із літератури. Таким чином, кількісний аналіз підтвердив, що запропонована модель є працездатною, але потребує подальшого вдосконалення для досягнення конкурентного рівня ефективності.

Практичне значення магістерської роботи полягає у створенні цілісної основи для розробки й тестування алгоритмічних трейдингових стратегій у блокчейн-мережах. Запропоновано інтерпретовану коінтеграційну модель, адаптовану до умов крипторинку, та реалізовано програмну інфраструктуру, яка поєднує сучасний технологічний стек (Rust, Redis, PostgreSQL, Solana, Jupiter) з можливістю гнучкого бектестування. Результати дослідження можуть бути використані як відправна точка для побудови більш складних стратегій: мультиспредових маркет-нейтральних портфелів, моделей з адаптивними порогоми, а також гібридних підходів, що поєднують коінтеграцію з методами машинного навчання. Це створює перспективу подальшого підвищення ефективності торгівлі на DeFi-ринках і сприяє розвитку науково обґрунтованих підходів до алгоритмічної торгівлі в блокчейн-екосистемі.

Результати дослідження подані на XVII Міжнародну науково-практичну конференцію «Development of science through the latest technologies and theories», 23-26 грудня 2025 р., Мілан, Італія [25].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Centralized Exchange.  
URL: <https://www.binance.com/en/academy/glossary/centralized-exchange> (дата звернення 15.10.2025)
2. Understanding Decentralized Finance (DeFi): Basics and Functionality.  
URL: <https://www.investopedia.com/decentralized-finance-defi-5113835> (дата звернення 15.10.2025)
3. Smart Contracts on Blockchain: Definition, Functionality, and Applications. URL: <https://www.investopedia.com/terms/s/smart-contracts.asp> (дата звернення 16.10.2025)
4. Basics of Algorithmic Trading: Concepts and Examples. URL: <https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp> (дата звернення 17.10.2025)
5. Introduction to Algo Trading Architecture. URL: <https://www.angelone.in/smart-money/trading-courses/introduction-algo-trading-architecture> (дата звернення 11.10.2025)
6. What Is a Blockchain? URL: <https://www.investopedia.com/terms/b/blockchain.asp> (дата звернення 17.10.2025)
7. Automated Trading Systems: Design, Architecture & Low Latency.  
URL: <https://www.quantinsti.com/articles/automated-trading-system/> (дата звернення 14.10.2025)
8. What are gas fees? URL: <https://www.coinbase.com/learn/crypto-basics/what-are-gas-fees> (дата звернення 17.10.2025)
9. Understanding Decentralized Finance (DeFi): Basics and Functionality.  
URL: <https://www.investopedia.com/decentralized-finance-defi-5113835> (дата звернення 15.10.2025)
10. What are frontrunners and MEV when it comes to crypto trading?

URL: <https://www.coinbase.com/learn/advanced-trading/what-are-frontrunners-and-mev-when-it-comes-to-crypto-trading> (дата звернення 22.10.2025)

11. Mean-Reversion Trading with Statistical Arbitrage Pair Trading Strategy. URL: <https://blog.quantinsti.com/epat-project-mean-reversion-statistical-arbitrage-pair-trading-strategy-indian-market-sectors/> (дата звернення 26.10.2025)

12. Mean Reversion Strategies: Introduction, Trading, Strategies and More. URL: <https://blog.quantinsti.com/mean-reversion-strategies-introduction-building-blocks/> (дата звернення 17.10.2025)

13. Crypto Portfolio Basics. URL: <https://calebandbrown.com/blog/balanced-crypto-portfolio/#:~:text=balanced%20crypto%20portfolio.-,What%20is%20a%20Crypto%20Portfolio%3F,stocks%2C%20real%20estate%20and%20bonds.> (дата звернення 28.10.2025)

14. Co-Integration and Error Correction: Representation, Estimation, and Testing. URL: <https://www.jstor.org/stable/1913236>. (дата звернення 27.10.2025)

15. Arbitrage on Decentralized Exchanges. URL: <https://arxiv.org/pdf/2507.08302>. (дата звернення 27.10.2025)

16. Decentralised Finance and Automated Market Making: Execution and Speculation. URL: <https://arxiv.org/pdf/2307.03499>. (дата звернення 27.10.2025)

17. Cryptocurrency Portfolio Optimization by Neural Networks. URL: <https://arxiv.org/abs/2310.01148> (дата звернення 24.10.2025)

18. An Ensemble Method of Deep Reinforcement Learning for Automated Cryptocurrency Trading. URL: <https://arxiv.org/pdf/2309.00626> (дата звернення 27.09.2025).

19. Optimal market-neutral currency trading on the cryptocurrency platform. URL: <https://arxiv.org/abs/2405.15461> (дата звернення 21.10.2025).

20. Evaluation of Dynamic Cointegration-Based Pairs Trading Strategy in the Cryptocurrency Market. URL: <https://arxiv.org/pdf/2109.10662> (дата звернення 23.10.2025).

21. Rust. URL: <https://rust-lang.org/> (дата звернення 28.09.2025)
22. Tokio Rust. URL: <https://tokio.rs/> (дата звернення 23.09.2025)
23. Redis. URL: <https://redis.io/> (дата звернення 25.09.2025)
24. Postgres. URL: <https://www.postgresql.org/> (дата звернення 29.09.2025).
25. Столбецький Ю. О., Калита Н. І. Дослідження алгоритмічних трейдингових стратегій у мережах блокчейн / XVII Міжнародна науково-практична конференція «Development of science through the latest technologies and theories», 23-26 грудня 2025 р., Мілан, Італія, с. 86 – 88. URL: <https://isg-konf.com/development-of-science-through-the-latest-technologies-and-theories/>