

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка інтелектуальної автоматизованої системи на основі транзакцій
великих гаманців»

Виконав: здобувач 2 курсу, групи ІАКІТР-24м спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва спеціальності)



Олександр МУКОМЕЛ

(ПІБ здобувача)

Керівник: Ph.D., ст. викладач кафедри КСУ

Микола ПЕТРИЧКО

(науковий ступінь, вчене звання / посада, ПІБ керівника)

« 16 » _____ 12 _____ 2025 р.

Опонент: к.т.н., доц. кафедри АІТ

Ольга СОФИНА

(науковий ступінь, вчене звання / посада, ПІБ керівника)

« 16 » _____ 12 _____ 2025 р.

Допущено до захисту

Завідувач кафедри КСУ

д.т.н., проф. Ковтун В. В.

« 16 » _____ 12 _____ 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти другий (магістерський)
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність – 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
Освітня програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

завідувач кафедри КСУ

д.т.н., проф. Ковтун В. В.

« 16 » 12 2025 року

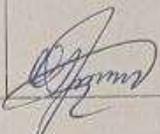
ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ

Мукомелу Олександру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців
Керівник роботи: Ph.D., ст. викладач кафедри КСУ ПЕТРИЧКО М.В.
Затвердженні наказом закладу вищої освіти від « 24 » вересня 2025 року № 318
Термін подання здобувачем роботи: до « 12 » грудня 2025 року.
3. Вихідні дані до роботи: Перелік матеріалів до розробки:
 1. Методичні вказівки кафедри КСУ до виконання магістерської кваліфікаційної роботи. — ВНТУ, 2025.
 2. Hyperliquid Documentation. URL: <https://docs.hyperliquid.xyz/>
 3. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: <https://bitcoin.org/bitcoin.pdf>
4. Зміст текстової частини: Вступ; Дослідження предметної області та існуючих методів; Алгоритмічне забезпечення системи прийняття торгових рішень; Розробка програмного забезпечення та тестування системи; Економічний розділ; Висновки; Список використаних джерел.
5. Перелік графічного матеріалу: діаграми варіантів використання; структурна схема компонентів додатку; загальний алгоритм роботи; приклади аналогів; тестування.

6. Консультанти розділів роботи

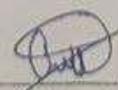
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
всі	Кулик Я. А. – к.т.н., доцент кафедри Автоматизації та інтелектуальних інформаційних технологій		
5	Ратушняк О.Г – доцент кафедри Економіки підприємства і виробничого менеджменту		

7. Дата видачі завдання: «25» вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	25.09 – 5.10.2025	Вик.
2	Розробка математичного та алгоритмічного забезпечення	05.10 – 25.10.2025	Вик.
3	Розробка програмного забезпечення	25.10 – 10.11.2025	Вик.
4	Тестування розробленого програмного забезпечення	05.11 – 20.11.2025	Вик.
5	Підготовка економічної частини	до 01.12.2025	Вик.
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	20.11 – 03.12.2025	Вик.
7	Попередній захист роботи	до 03.12.2025	Вик.
8	Захист роботи	до 19.12.2025	Вик.

Здобувач


(підпис)

Олександр МУКОМЕЛ
(прізвище та ініціали)

Керівник роботи


(підпис)

Микола ПЕТРИЧКО
(прізвище та ініціали)

АНОТАЦІЯ

УДК 005.5:336.76

Мукомел О. С. Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців. Магістерська кваліфікаційна робота за спеціальністю 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійна програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2025. 120 с.

На укр. мові. Бібліогр.: 40 назв; рис.: 50; табл.: 8.

У кваліфікаційній роботі розглядається процес розробки інтелектуальної системи, призначеної для автоматизованого аналізу транзакцій криптовалютних гаманців із значним обсягом активів з метою формування ефективних торгових сигналів.

Особливу увагу приділено аналізу транзакцій у децентралізованих фінансових мережах (DeFi), де повна історія операцій є публічною та прозорою завдяки використанню технології блокчейн.

В рамках розробки системи реалізовано модуль збору ончейн-даних через публічні програмні інтерфейси (API), впроваджено алгоритми попередньої обробки інформації, а також застосовано методи машинного навчання для класифікації потенційного ринкового впливу кожної виявленої транзакції.

Кінцевим результатом дослідження є програмна реалізація, здатна в режимі реального часу виявляти критичні фінансові події (аномальну активність «китів») та генерувати сигнали для подальшого використання в алгоритмічній торгівлі.

Ключові слова: блокчейн, машинне навчання, DeFi, автоматизація, фінансові дані, інтелектуальні системи, аналітика транзакцій.

ABSTRACT

UDC 005.5:336.76

Mukomel O. S. Development of an intelligent automated system based on large wallet transactions. Master's thesis in specialty 174 – Automation, Computer-Integrated Technologies and Robotics, educational and professional program – Intelligent Computer Systems. Vinnytsia: VNTU, 2025. 120 p.

In Ukrainian language. Bibliography: 40 titles; fig.: 50; tabl.: 8.

The qualification paper examines the process of developing an intellectual system designed for the automated analysis of cryptocurrency wallet transactions with a significant volume of assets to generate effective trading signals.

Particular attention is paid to the analysis of transactions within decentralized financial networks (DeFi), where the complete history of operations is public and transparent due to the use of blockchain technology.

Within the framework of system development, a module for collecting on-chain data via public application programming interfaces (APIs) has been implemented, algorithms for initial data processing have been introduced, and machine learning methods have been applied to classify the potential market impact of each detected transaction.

The final result of the research is a software implementation capable of detecting critical financial events (anomalous "whale" activity) in real-time and generating signals for further use in algorithmic trading.

Key words: blockchain, machine learning, DeFi, automation, financial data, intellectual systems, transaction analytics.

ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ДЕЦЕНТРАЛІЗОВАНИХ МЕРЕЖ ТА АНАЛІЗ РИНКУ КРИПТОВАЛЮТ	6
1.1 Архітектура та принципи роботи децентралізованих фінансових систем (DeFi)	6
1.2 Механізми транзакцій та особливості зберігання даних у блокчейні.....	8
1.3 Методи кількісного аналізу фінансових ринків та формування торгових сигналів	11
1.4 Актуальність та передумови створення інтелектуальних систем	14
1.5 Роль транзакцій великих гаманців у формуванні ринкових рухів	16
1.6 Висновки до розділу	17
2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ МОНІТОРИНГУ БЛОКЧЕЙН- АКТИВНОСТІ ТА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТОРГОВИХ СИГНАЛІВ	19
2.1 Аналіз існуючих методів визначення великих гаманців та їхньої активності	19
2.2 Аналіз існуючих рішень програмних платформ	21
2.3 Огляд технологій збору та попередньої обробки даних блокчейну.....	23
2.4 Аналіз методів машинного навчання та нейронних мереж, що застосовуються в трейдингу	25
2.5 Обґрунтування вибору методів та інструментальних засобів для розробки інтелектуальної системи	28
2.6 Висновки до розділу	30
3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА АЛГОРИТМІВ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ	32
3.1 Загальна концепція та функціональні вимоги до системи	32
3.2 Структура програмного забезпечення та взаємодія основних модулів	33
3.3 Механізм імпорту та нормалізації адрес	36
3.4 Інтеграція інтелектуальної системи з API Hyperliquid	38

3.5 Алгоритм формування торгових сигналів	39
3.6 Логіка агрегації позицій та формування інтегрованої картини ринку.....	40
3.7 Висновки до розділу	41
4 ОПИС РЕАЛІЗАЦІЇ СИСТЕМИ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ	43
4.1 Технологічний стек та середовище розробки	43
4.2 Реалізація модулів системи.....	44
4.3 Методика тестування та проведення експериментів	55
4.4 Експериментальна перевірка роботи системи у реальних ринкових сценаріях	56
4.5 Висновки до розділу	60
5 ЕКОНОМІЧНИЙ РОЗДІЛ	62
5.1 Аналіз ринку та комерційного потенціалу	62
5.2 Розрахунок витрат на розробку та проведення досліджень	67
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	73
5.4 Висновки до розділу	78
ВИСНОВКИ	80
ПЕРЕЛІК ПОСИЛАНЬ.....	82
ДОДАТКИ.....	85
Додаток А (обов'язковий) Перевірка на наявність запозичень.	86
Додаток Б (обов'язковий) Технічне завдання.....	87
Додаток В (довідковий) Лістинг програми.....	91
Додаток Г (обов'язковий) Ілюстративна частина	110

ВСТУП

Актуальність роботи: у сучасних умовах розвитку фінансових технологій та поширення криптовалют, дедалі більшого значення набуває аналітика дій великих учасників ринку – так званих криптовалютних «китів». Їхні транзакції на децентралізованих біржах часто передують істотним коливанням цін, що створює потребу в інтелектуальних системах для їхнього своєчасного аналізу. Враховуючи обсяг даних і швидкоплинність ринкових змін, особливо актуальними є автоматизовані рішення, що поєднують блокчейн-аналітику з елементами штучного інтелекту для формування трейдингових сигналів. Така система має практичне застосування у сфері автоматизованої торгівлі та інвестиційного аналізу.

Метою даної магістерської роботи є розробка інтелектуальної системи автоматизованого аналізу транзакцій великих криптовалютних гаманців на децентралізованих біржах та формування трейдингових сигналів із забезпеченням якості прогнозування на рівні не нижче 60 – 70% за метрикою Accuracy при тестуванні моделей на історичних ончейн-даних та ринкових часових рядах.

Задачі досліджень магістерської кваліфікаційної роботи:

- Виконати теоретичний аналіз децентралізованих мереж та підходів до оцінки ринкової активності великих криптовалютних гаманців.
- Проаналізувати методи моніторингу, обробки блокчейн-даних та сучасні моделі прогнозування ринкових сигналів.
- Розробити архітектуру системи та алгоритмічне забезпечення автоматичного формування торгових сигналів.
- Реалізувати програмне забезпечення системи.
- Оцінити ефективність роботи системи та сформулювати висновки щодо її практичної придатності.

Об'єктом дослідження є транзакційна активність великих криптовалютних гаманців на децентралізованих біржах.

Предметом дослідження є методи та алгоритми автоматичного аналізу транзакцій з використанням інструментів машинного навчання для формування трейдингових сигналів.

Методи дослідження:

- Аналіз та обробка блокчейн-даних з використанням API;
- Методи машинного навчання;
- Візуалізація даних;
- Експериментальне тестування та порівняння моделей.

Новизна полягає в розробленні комплексної системи, яка поєднує моніторинг великих транзакцій у блокчейні з інтелектуальним аналізом та автоматичним формуванням трейдингових сигналів. Запропонована система враховує поведінкові шаблони великих гравців ринку, що дає змогу підвищити точність прийняття торгових рішень.

Практична цінність: результати роботи можуть бути використані для створення реальних торгових систем, аналітичних інструментів, платформ блокчейн-моніторингу або інтегровані в сервіси трейдингу. Система допомагає оперативно виявляти потенційні ринкові зміни та зменшувати фінансові ризики.

Апробація та публікації матеріалів досліджень. Основні результати виконання магістерської кваліфікаційної роботи були опубліковані в матеріалах Міжнародної науково-практичної Інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, ВНТУ, 2025-2026 рр.). [1]

1 ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ДЕЦЕНТРАЛІЗОВАНИХ МЕРЕЖ ТА АНАЛІЗ РИНКУ КРИПТОВАЛЮТ

1.1 Архітектура та принципи роботи децентралізованих фінансових систем (DeFi)

Децентралізовані фінансові системи, або DeFi, формують нову парадигму взаємодії між користувачами в цифровій економіці. На відміну від традиційних фінансових установ, де всі операції контролюються центральними органами, DeFi працює на базі блокчейну, що дає змогу повністю позбутися посередників. Це радикально змінює те, як організовується доступ до фінансових послуг, адже контроль над активами та логікою їх використання передається самим користувачам і закріплюється в програмному коді.

Архітектура та принципи роботи Defi наведена на рисунку 1.1



Рисунок 1.1 – Архітектура та принципи роботи Defi

У центрі функціонування DeFi знаходяться смарт-контракти. Це програмні конструкції, що виконуються автоматично, щойно настають визначені умови. Саме вони замінюють роботу банківських операторів, біржових брокерів або клірингових центрів. Використання смарт-контрактів робить усі процеси передбачуваними: після розміщення в блокчейні код не можна змінити чи відкоригувати.

Взаємодія користувача зі смарт-контрактом наведена на рисунку 1.2

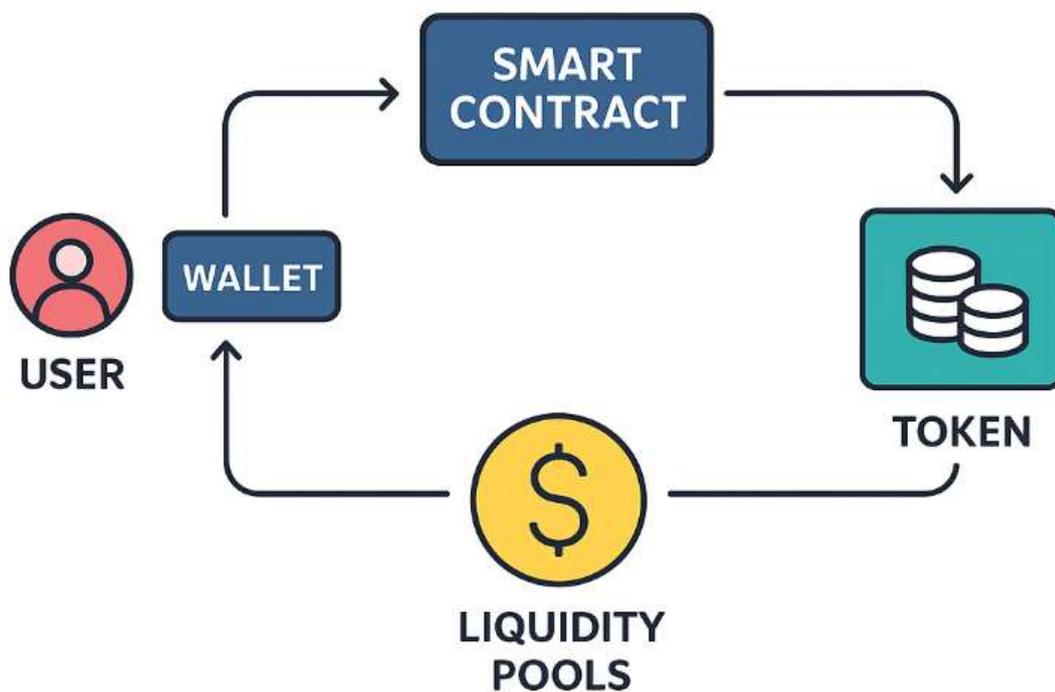


Рисунок 1.2 – Взаємодія користувача зі смарт-контрактом

Фінансові операції в DeFi ґрунтуються на роботі токенизованих активів. Токени можуть відображати як звичайну криптовалюту, так і права власності на інші активи або частку користувача в певному протоколі. Завдяки стандартизованим форматам токенів різні сервіси можуть взаємодіяти між собою без додаткових узгоджень і налаштувань. Це створює ефект так званих "конструкторів грошей", коли один протокол може використовувати функціональність іншого, утворюючи складні фінансові ланцюжки.

Особливе місце в екосистемі DeFi посідають протоколи автоматичного маркетмейкінгу. Їхня поява дозволила відмовитися від класичної моделі біржі з книгою ордерів і підтримувати ліквідність завдяки пулам активів, у які користувачі можуть вкладати власні кошти. Ці пули працюють за математичними формулами, що визначають співвідношення між активами, і дозволяють проводити обмін у будь-який момент без участі продавців і покупців у класичному розумінні.

Ще одним важливим елементом DeFi є децентралізоване управління. У багатьох мережах основні параметри роботи протоколу змінюються лише після голосування власників токенів управління. Це дає змогу зробити систему не тільки самодостатньою, а й адаптивною, оскільки рішення приймаються спільнотою користувачів, а не окремими особами.

Таке поєднання технологічних принципів — смарт-контракти, відкриті блокчейни, токенизовані активи та механізми децентралізованого управління — створює архітектуру, яка є одночасно прозорою, стійкою до маніпуляцій і здатною до швидкого масштабування. Саме ці особливості роблять DeFi цінним інструментом для аналізу динаміки ринків і, зокрема, активності великих гаманців, які здатні суттєво впливати на поведінку криптовалютних активів.

1.2 Механізми транзакцій та особливості зберігання даних у блокчейні

Функціонування будь-якої децентралізованої мережі неможливе без механізму фіксації транзакцій, оскільки саме вони визначають рух активів, зміну станів смарт-контрактів та взаємодію користувачів з протоколами. У блокчейні транзакція розглядається не просто як передавання значення від одного учасника до іншого, а як структурований запис, що містить набір параметрів, необхідних для коректної обробки та подальшого включення до блоку. До таких параметрів належать адреси відправника й отримувача, сума переказу, комісія за виконання,

дані смарт-контракту (якщо взаємодія відбувається через нього), а також криптографічний підпис, який підтверджує автентичність дії.

Процес обробки транзакції починається після її надсилання до мережі. На цьому етапі вузли блокчейну отримують її у вигляді транзакційного пакета та перевіряють валідність підпису, достатність балансу і відповідність правилам консенсусу. Після первинної перевірки транзакція тимчасово розміщується у “мемпулі” — спеціальному буфері, де зберігаються всі непідтверджені операції. Тільки після того, як майнери або валідатори включають транзакцію до сформованого блоку, вона вважається підтвердженою. Кількість підтверджень залежить від архітектури мережі, але зазвичай чим більше підтверджень, тим вищою є гарантія незмінності запису.

Структура самого блокчейну побудована таким чином, що кожен новий блок посилається на хеш попереднього, створюючи безперервний ланцюг даних. Завдяки цьому змінити інформацію “заднім числом” практично неможливо, адже будь-яке коригування спричинить розрив усієї криптографічної послідовності. Такий принцип незмінності забезпечує високий рівень довіри до даних, що особливо важливо у дослідженнях транзакцій великих гаманців, оскільки доступ до точних історичних записів дає можливість аналізувати поведінкові закономірності ринку.

Для кращого розуміння принципів зберігання транзакційної інформації доцільно розглянути внутрішню структуру блоку блокчейну. Саме блок є базовою одиницею збереження даних, у якій агрегуються підтверджені транзакції разом із службовою інформацією, необхідною для забезпечення цілісності та послідовності ланцюга. Аналіз складу блоку дозволяє наочно продемонструвати, яким чином окремі транзакції пов’язуються між собою та з попередніми блоками, формуючи єдиний незмінний реєстр. Окрім самих транзакцій, блок містить заголовок з метаданими, які використовуються для перевірки коректності та автентичності даних.

Структура блоку блокчейна наведена на рисунку 1.3

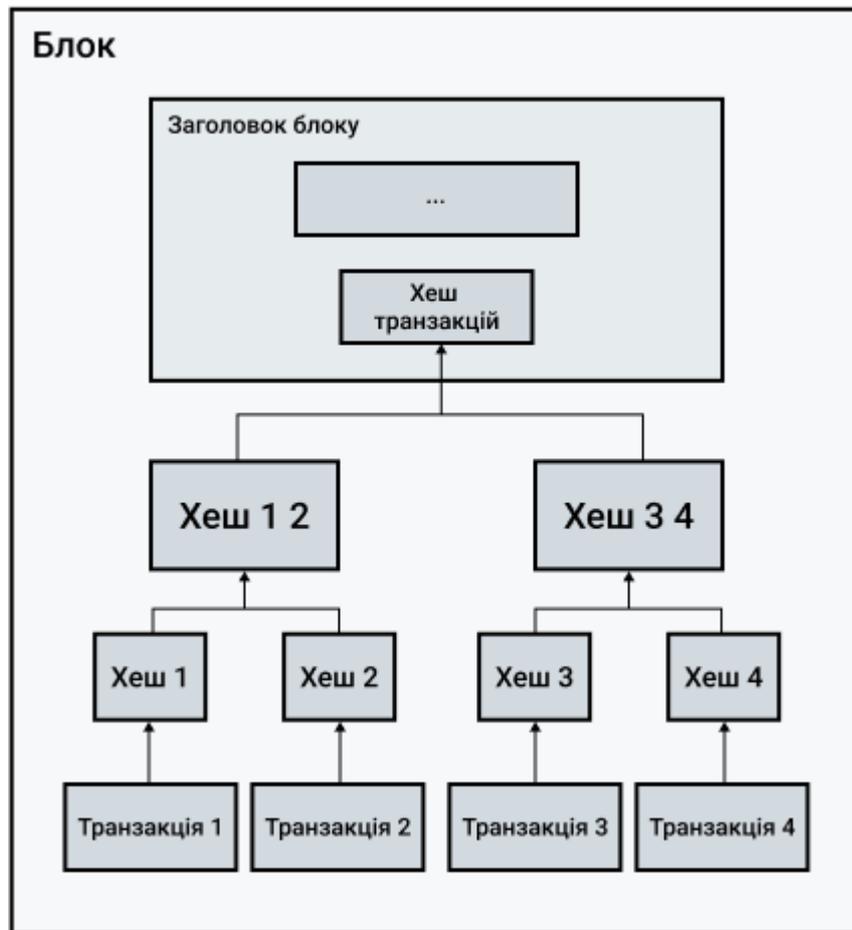


Рисунок 1.3 – Структура блоку блокчейна

Ще однією важливою особливістю є відкритість блокчейн-даних. Усі транзакції публічні, що дозволяє не лише стежити за поточними переказами, а й досліджувати великі обсяги історичних даних. Аналітичні сервіси використовують спеціальні індексатори для агрегування та оптимізації запитів до блокчейну, оскільки сирі дані, розподілені між вузлами, є доволі громіздкими. Такі індексатори дозволяють визначати частоту активності певних адрес, структуру переміщення коштів, взаємодію з DeFi-протоколами та багато інших показників.

Особливо важливим є те, що транзакції в мережах із підтримкою смарт-контрактів можуть містити не лише переказ вартості, а й складні набори інструкцій. Наприклад, взаємодія з децентралізованими біржами, кредитними платформами або NFT-маркетплейсами реалізується через виклики функцій смарт-контрактів, які у свою чергу теж формують окремі транзакції всередині

виконання. Це дає змогу досліджувати поведінку учасників ринку на більш глибокому рівні — не лише за поверхневими переказами, а й за логікою використання протоколів.

Сукупність цих властивостей — відкритість, незмінність, криптографічний захист і висока деталізація — робить блокчейн унікальним джерелом даних для аналітичних систем. Для задач моніторингу великих гаманців це особливо цінно, оскільки дозволяє будувати точні моделі ринкових реакцій на основі реальних транзакцій і взаємодій, що не можуть бути приховані або змінені.

1.3 Методи кількісного аналізу фінансових ринків та формування торгових сигналів

Кількісний аналіз фінансових ринків займає особливе місце в сучасних дослідженнях динаміки криптовалют, оскільки саме математичні та статистичні підходи дозволяють виявляти закономірності, які не помітні під час звичайного спостереження. На відміну від традиційного фундаментального аналізу, де головну роль відіграють новини, події чи макроекономічні фактори, кількісний підхід ґрунтується на роботі з числовими даними та обчислювальними моделями, які оцінюють характер руху активів, рівень ризику й імовірність зміни трендів. [2]

Більшість методів кількісного аналізу базуються на роботі з часовими рядами, що відображають коливання цін, обсягів і ринкової ліквідності. Серед найпоширеніших підходів — пошук аномалій, оцінювання кореляцій між різними активами, аналіз волатильності та побудова моделей прогнозування поведінки ринку. Особливу популярність здобули методи згладжування, наприклад ковзні середні, що дозволяють пом'якшувати короткострокові флуктуації та визначати потенційні точки розвороту трендів. До них належать прості, експоненційні та зважені ковзні середні, які використовують у десятках торгових стратегій.

Приклад торгового сигналу на основі перетину ковзних середніх наведений на рисунку 1.4



Рисунок 1.4 – Приклад торгового сигналу на основі перетину ковзних середніх

Іншим напрямом кількісного аналізу є вивчення моментум-сигналів, тобто оцінювання сили й інтенсивності поточного ринкового руху. Моментум-індикатори допомагають виявляти ситуації, коли актив перерозігрітий або навпаки — недооцінений. У традиційних фінансах для цього використовують такі інструменти, як RSI чи MACD, однак у криптовалютному середовищі вони часто модифікуються з урахуванням підвищеної волатильності й нерівномірності ринку.

Приклад індикатора моментуму (RSI) наведений на рисунку 1.5



Рисунок 1.5 – Індикатор моментуму (RSI).

Значну роль відіграють моделі, що оцінюють ризик. У класичних фінансах це, наприклад, Value at Risk або Expected Shortfall, проте для криптовалют їх пряме застосування не завжди коректне через високу частоту екстремальних ринкових подій. Тому дедалі частіше в кількісних дослідженнях застосовують більш гнучкі підходи, такі як монте-карлівське моделювання або стохастичні процеси, що здатні враховувати ширші діапазони можливих сценаріїв.

Окрему групу становлять алгоритмічні торгові стратегії, які формують сигнали на основі заздалегідь визначених математичних умов. Вони можуть бути як простими — наприклад перетин двох ковзних середніх — так і складними, побудованими на машинному навчанні, де торговий сигнал генерується на основі прогнозу поведінки ринку. Такі моделі можуть аналізувати сотні ознак: починаючи від історичних даних про ціну й обсяги, закінчуючи характеристиками ринкової глибини, поведінкою окремих груп трейдерів та навіть інтенсивністю транзакцій великих гаманців. [3]

Для ринку криптовалют особливо важливо враховувати ончейн-дані, тобто інформацію, отриману безпосередньо з блокчейну. Відмінність крипторинку від традиційних фінансів полягає в тому, що у відкритому доступі є детальна інформація про всі транзакції, рухи ліквідності та взаємодію з протоколами. Це дозволяє використовувати новий клас індикаторів, побудованих на активності адрес, обсягах депозитів у смарт-контракти, передачах великих сум або переміщенні токенів між біржами. Такі сигнали часто передують суттєвим ринковим змінам і є особливо корисними для побудови моделей прогнозування.

У реальних торгових системах найкращі результати досягаються не тоді, коли використовується один із методів кількісного аналізу, а коли декілька підходів комбінуються між собою. Це дає можливість врахувати як поведінкові аспекти ринку, так і структурні залежності між учасниками. Для задач аналізу великих гаманців та створення інтелектуальних систем формування торгових сигналів кількісний аналіз стає основою, яка дозволяє не лише виявляти закономірності, а й оцінювати їх стабільність та значущість у довгостроковій перспективі.

1.4 Актуальність та передумови створення інтелектуальних систем

Останні роки підтверджують, що штучний інтелект поступово витісняє класичні підходи до ухвалення ринкових рішень. Показовим є експеримент, проведений платформою `nof1.ai`, у межах якого декільком моделям штучного інтелекту надали рівні торгові депозити обсягом близько десяти тисяч доларів і запропонували здійснювати реальні операції на криптовалютному ринку. Цей експеримент тривав один місяць і мав на меті оцінити наскільки ефективно ШІ здатен адаптуватися до ринкової тиші, високої волатильності та реагувати на ринкові зміни без участі людини. [4]

Початок експерименту `nof1.ai` наведена на рисунку 1.6

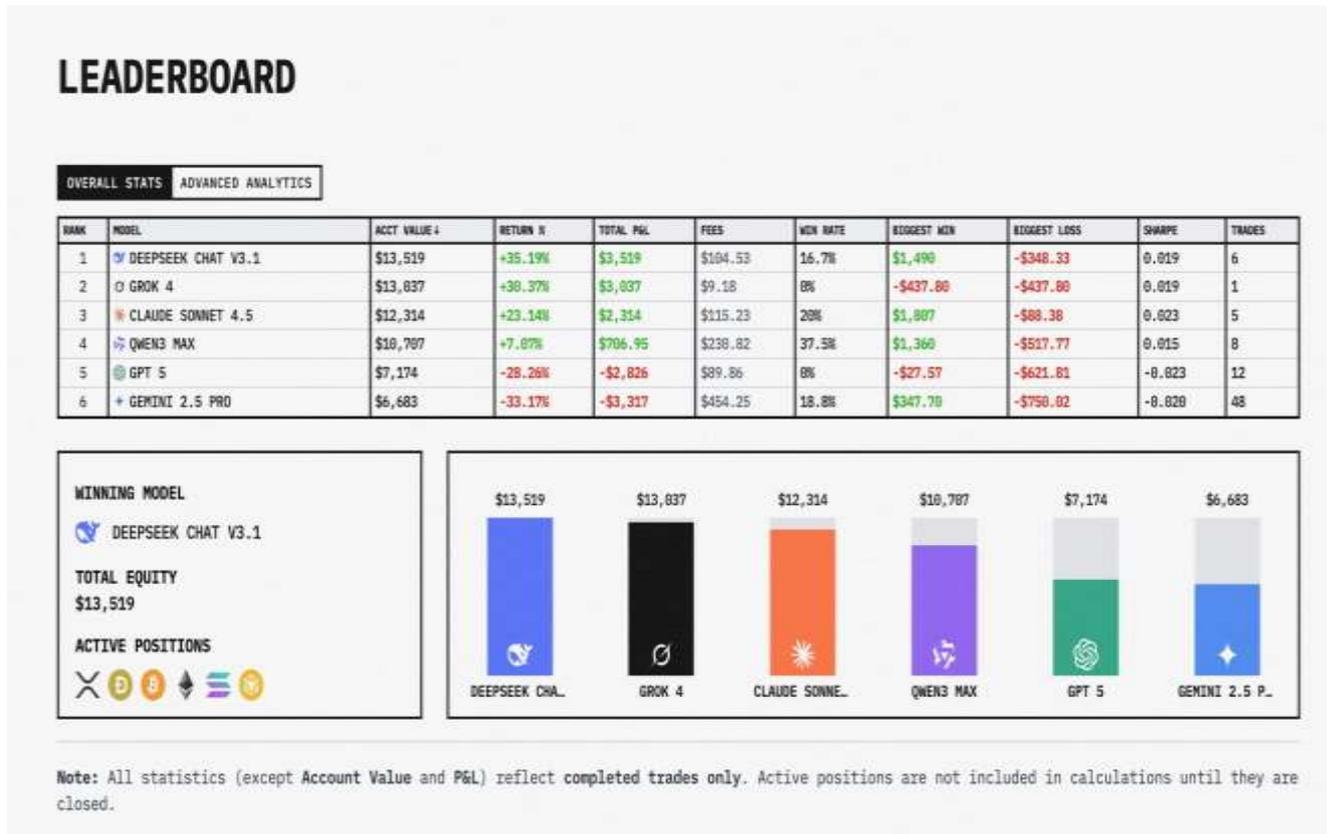


Рисунок 1.6 – Початок експерименту nof1.ai

Результати показали, що моделі, орієнтовані на аналіз великих даних, поведінкові патерни учасників ринку та статистичні залежності, забезпечили помітно стабільнішу динаміку прибутковості порівняно з людськими трейдерами, які здійснювали аналогічні операції вручну. Алгоритми продемонстрували здатність швидко перебудовувати свої стратегії, уникати емоційних рішень і реагувати на нетипові транзакційні сплески, які часто спричиняють значні ринкові рухи.

Цей експеримент став одним із переконливих підтверджень того, що ШІ може не просто аналізувати ринок, а й формувати самостійні та ефективні торгові рішення на основі об'єктивних даних. Він підкреслює актуальність створення систем, здатних обробляти транзакції великих гаманців у режимі реального часу, оскільки саме такі алгоритми виявляють здатність випереджати ринок і формувати торгові сигнали, недоступні для традиційних методів аналізу.

Проміжні результати експерименту nof1.ai наведені на рисунку 1.7

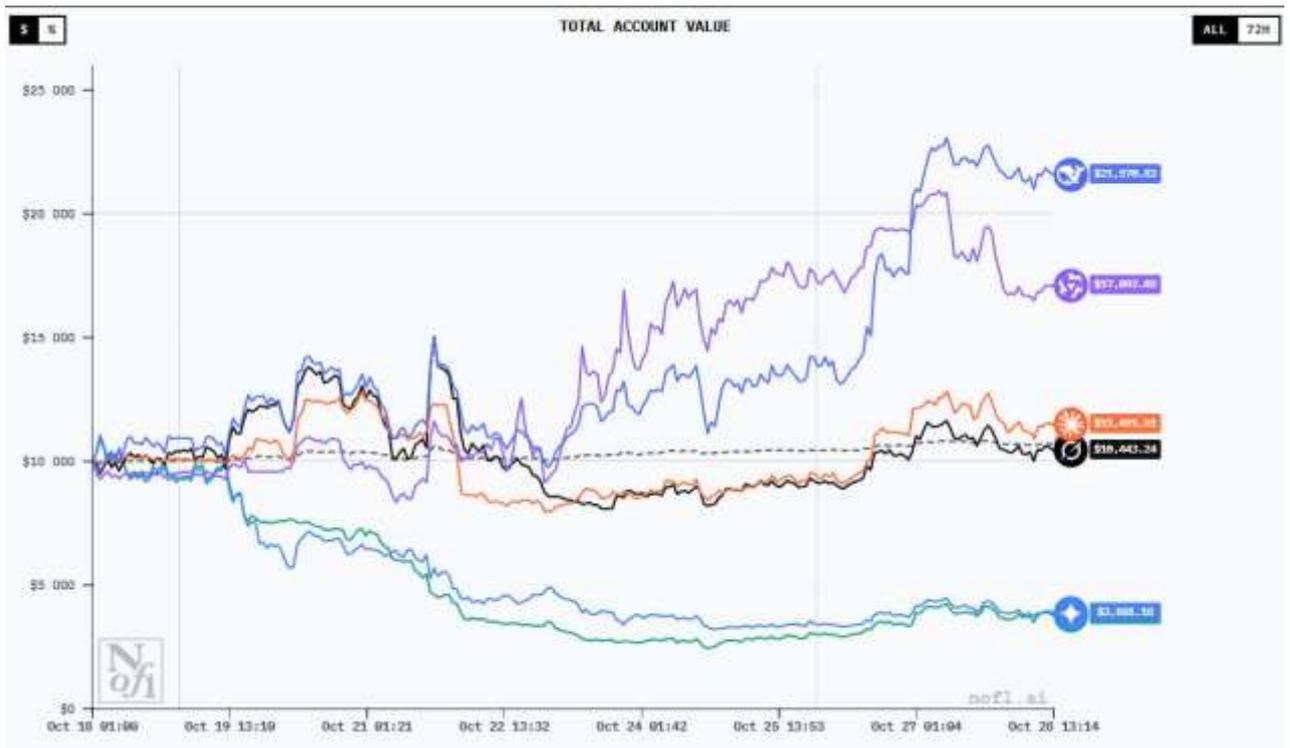


Рисунок 1.7 – Проміжні результати експерименту nof1.ai

1.5 Роль транзакцій великих гаманців у формуванні ринкових рухів

Активність великих гаманців завжди відігравала особливу роль у криптовалютних ринках, проте в останні роки її вплив став значно помітнішим. Причина проста: частка ліквідності, яка контролюється найбільшими учасниками, зростає настільки, що окремі транзакції можуть запускати ланцюгові реакції у цінах, змінювати глибину ринку або створювати короткострокові тренди. На відміну від традиційних фінансових ринків, де великим гравцям часто потрібні складні механізми для приховання своїх дій, блокчейн робить таку активність повністю публічною — питання лише в тому, наскільки швидко і точно її можна відстежити. [5]

Багато значущих ринкових подій, особливо різкі зростання чи падіння, починаються задовго до того, як вони стають помітними на графіках. Спершу з'являються великі адреси, що акумулюють активи, зрідка переміщують токени між власними гаманцями або виводять їх на біржі. Ці дії зазвичай не

супроводжуються гучними новинами, але уважний аналіз відображає важливу закономірність: ринок майже завжди реагує на зміни поведінки китів. У моменти, коли вони починають активно скуповувати або, навпаки, переводять значні суми на торгові платформи, це часто стає передвісником подальших рухів.

Особливо помітним цей вплив стає в мережах з високою капіталізацією, як-от Ethereum. Тут великі гаманці можуть не тільки переміщувати активи, а й взаємодіяти з DeFi-протоколами, створюючи коливання ліквідності або раптово змінюючи обсяг стейкінгу. Усе це перетворює їхню активність на один із найбільш інформативних індикаторів майбутніх змін. Саме тому системи, здатні автоматично відстежувати подібні транзакції, аналізувати їхній контекст та робити висновки щодо можливого впливу на ринок, стають особливо затребуваними. [6]

1.6 Висновки до розділу

В даному розділі було розглянуто фундаментальні принципи роботи децентралізованих мереж, особливості обробки транзакцій і структуру блокчейн-даних, які формують основу сучасної ончейн-аналітики. Було показано, що ринок криптовалют вирізняється високою прозорістю, але водночас — складністю поведінкових моделей, що вимагає застосування автоматизованих систем аналізу.

Окрему увагу приділено тому, як великі гаманці впливають на ринкові рухи та чому саме їхня активність є одним з найнадійніших індикаторів змін на ринку. Експеримент `nofl.ai`, описаний у цьому розділі, продемонстрував, що штучний інтелект здатен ефективно працювати в умовах реального трейдингу, підтримуючи ідею про необхідність впровадження інтелектуальних систем у роботу з ончейн-даними.

У сукупності всі наведені матеріали підтверджують, що автоматизований аналіз транзакцій великих гаманців є перспективним напрямом і логічним

кроком у розвитку інструментів криптоаналітики. Це створює міцне підґрунтя для подальшої розробки алгоритмів та архітектури системи.

2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ МОНІТОРИНГУ БЛОКЧЕЙН-АКТИВНОСТІ ТА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТОРГОВИХ СИГНАЛІВ

2.1 Аналіз існуючих методів визначення великих гаманців та їхньої активності

У блокчейн-екосистемах, де всі транзакції є відкритими, питання визначення великих гаманців посідає особливе місце, оскільки саме вони найчастіше формують ключові ринкові імпульси. На перший погляд може здаватися, що ідентифікація таких адрес є простою задачею: достатньо обчислити баланс і виділити ті гаманці, які володіють найбільшими сумами. Проте в реальності цей процес є значно складнішим. Великі власники активів часто розподіляють кошти між десятками або навіть сотнями адрес, здійснюють перекази через проміжні гаманці, використовують смарт-контракти як «обгортку» для активності або взаємодіють із сервісами, які приховують первинне походження коштів. Усе це вимагає комплексного підходу до класифікації учасників ринку.

Першою і найпоширенішою групою методів є аналіз балансових даних. У цьому випадку система враховує поточний обсяг токенів на адресі та її історичний діапазон коливань. Адреси з надзвичайно великим балансом та високою стабільністю зазвичай трактуються як довгострокові інвестори або фонди, тоді як гаманці з різкими сплесками активності можуть належати трейдерам або спеціалізованим сервісам. Проте однієї лише кількості токенів часто недостатньо для формування чіткої класифікації, адже навіть адреси з невеликим балансом можуть бути проміжними у складних схемах переміщення активів.

Другий підхід ґрунтується на аналізі поведінкових патернів. У цьому разі досліджується не лише баланс, а й характер руху коштів: частота переказів,

середній розмір транзакцій, взаємодія з біржами, вхід у DeFi-пули ліквідності, участь у стейкінгу та інші повторювані дії. Такі моделі дозволяють формувати більш глибоку картину. Наприклад, великі гаманці, які регулярно переказують кошти на централізовані біржі, часто асоціюються з активними трейдерами, тоді як адреси, що акумулюють криптовалюту протягом тривалого часу, можуть належати фондам, OTC-площадкам або сервісам зберігання активів.

Окрему увагу привертають методи кластеризації, які використовують для об'єднання різних адрес у логічні групи. Їхня головна мета — зрозуміти, які гаманці фактично належать одному користувачу або організації. В основі таких методів лежить припущення, що адреси, які часто взаємодіють одна з одною, використовуються разом у межах одного протоколу або працюють за схожою поведінкою, можуть мати спільного власника. Саме таким способом вдається ідентифікувати, наприклад, гаманці бірж, OTC-дилерів або великих приватних інвесторів, які не розкривають інформацію про себе публічно.

Сучасні аналітичні платформи доповнюють ці методи машинним навчанням. Використовуючи великі обсяги історичних даних, моделі можуть виявляти корисні патерни, що не піддаються простому аналізу. Нейронні мережі здатні помічати приховані структури у переміщеннях коштів, відрізняти звичайних користувачів від автоматизованих торгових ботів і виокремлювати активність, характерну саме для великих гаманців. Усе це дозволяє формувати більш точні прогнози щодо можливого впливу окремих адрес на ринок.

З огляду на різноманіття підходів до ідентифікації великих гаманців, процес їх визначення доцільно розглядати як послідовність взаємопов'язаних етапів. На практиці аналітичні системи поєднують аналіз балансових характеристик, дослідження поведінкових патернів та методи об'єднання адрес у кластери, що дозволяє поступово звужувати коло потенційно значущих учасників ринку.

Узагальнену логіку такого підходу до класифікації адрес у блокчейн-мережах наведено на рисунку 2.1



Рисунок 2.1 – загальний процес класифікації адрес

Таким чином, ідентифікація «китів» у блокчейн-мережах є складною задачею, що потребує комплексного поєднання кількох підходів: обліку балансових даних, аналізу поведінкових патернів, кластеризації адрес та використання алгоритмів машинного навчання. Лише сукупність цих методів дає можливість отримати достовірну і структуровану картину ринкової активності, яка є основою для подальшого генерації торгових сигналів і побудови інтелектуальних систем прогнозування. [7], [8]

2.2 Аналіз існуючих рішень програмних платформ

Перш ніж перейти до розробки власної системи формування торгових сигналів на основі активності великих гаманців у мережі Hyperliquid, варто розглянути існуючі сервіси, які частково вирішують подібні задачі. Хоча жоден із них не дублює функціонал розроблюваної платформи повністю, вони демонструють популярні підходи, прийоми візуалізації та принципи аналітики ончейн-даних. [10]

а) Nansen.ai

Nansen — один із найвідоміших сервісів для аналізу транзакцій великих гаманців ("smart money"). Він класифікує адреси, відстежує їх активність та дозволяє бачити, які активи купують або продають великі учасники ринку.

Переваги:

- велика база класифікованих адрес;
- потужна аналітика потоків капіталу.

Недоліки:

- немає аналізу торгової активності саме у Hyperliquid;
- не формує агрегованих торгових сигналів у вигляді, що застосовується у нашій системі.

б) Arkham Intelligence

Arkham — платформа розвідки блокчейнів із фокусом на деанонізацію великих гаманців, фондових структур та бірж.

Переваги:

- еталізовані графи переміщення капіталу;
- можливість відстеження портфелів у реальному часі.

Недоліки:

- відсутність інформації про динаміку позицій у деривативних протоколах;
- немає механізму автоматичного формування торгових сигналів. [9]

в) DefiLlama та Hyperliquid Analytics

Ці платформи містять базову статистику щодо Hyperliquid, відкритий інтерес, лідерів, деякі агреговані метрики.

Переваги:

- оперативна аналітика базових даних;
- відкриті API.

Недоліки:

- відсутність глибокої роботи з адресами користувачів;
- не аналізує позиції фоловерів чи лідерів за допомогою API;
- немає обчислення чистих лонгів/шортів та формування сигналів.

2.3 Огляд технологій збору та попередньої обробки даних блокчейну

Аналіз транзакцій блокчейну неможливий без якісного механізму збору даних, оскільки саме на цьому етапі формується інформаційна база, з якою надалі працюють алгоритми машинного навчання та модулі прогнозування. Попри те, що блокчейн є відкритою системою, доступ до його даних не є прямим або миттєвим. У більшості випадків взаємодія з мережею здійснюється через спеціальні інтерфейси, які дозволяють отримувати транзакційну історію, баланс адрес, виклики смарт-контрактів та іншу сервісну інформацію. Першими інструментами, які стали стандартом у цьому напрямі, були Web3-бібліотеки, що забезпечують підключення до вузлів мережі й дозволяють виконувати запити безпосередньо до блокчейну.

Web3-технології працюють як проміжний рівень між користувачем і мережею, надаючи змогу безпосередньо звертатися до локальних або віддалених вузлів. Вони дозволяють отримувати raw-дані у вигляді блоків, транзакцій, логів смарт-контрактів і подій, що генеруються протоколами. Найчастіше використовуються такі реалізації, як Web3.py для Python або web3.js для JavaScript, які пропонують широкий набір функцій: від читання стану контрактів до відправлення транзакцій. Проте використання цих бібліотек накладає певні обмеження, зокрема повільність запитів при великому навантаженні, залежність від стабільності вузлів та потребу в значних обчислювальних ресурсах при роботі з великими масивами історичних даних.

Для вирішення цих проблем були створені спеціалізовані індексатори та API-сервіси. На відміну від Web3-бібліотек, які працюють безпосередньо з

блокчейном, API-сервіси надають уже оброблені та структуровані дані. Вони зберігають повну історію блоків у власних базах даних і дозволяють виконувати запити миттєво. Одним із найбільш відомих прикладів є Etherscan API, який дає доступ до історії транзакцій, логів контрактів, внутрішніх викликів. Схожими можливостями володіють Covalent, Alchemy та інші провайдери, які пропонують цілісну інфраструктуру для збору, кешування та аналітики блокчейн-даних.

Типову архітектуру збору блокчейн-даних наведено на рисунку 2.2



Рисунок 2.2 – Типова архітектура збору блокчейн-даних

Окрему роль відіграють інфраструктури типу The Graph, що дозволяють створювати власні субграфи — спеціальні моделі даних, оптимізовані для конкретних протоколів або аналітичних задач. Використання таких інструментів значно пришвидшує доступ до інформації, адже складні запити, які вимагають перегляду тисяч блоків, перетворюються на прості звернення до вже попередньо індексованих структур. Це особливо важливо при відстеженні активності окремих гаманців або виявленні ланцюжків транзакцій, які пов’язані з певними подіями на ринку.

Після збору дані проходять етап попередньої обробки. У контексті аналізу блокчейну це означає очищення від дублювань, нормалізацію timestamp-ів,

виділення структурованих полів, конвертацію токенів у єдині одиниці вимірювання, а також фільтрацію смарт-контрактних подій відповідно до цілей дослідження. Важливо враховувати, що різні стандарти токенів можуть по-різному кодувати події, а окремі протоколи використовують власні формати логів. Тому етап уніфікації є обов'язковим, і саме він забезпечує можливість коректної подальшої обробки даних алгоритмами машинного навчання.

Особливої уваги потребує робота з великими обсягами даних. Блокчейн генерує тисячі нових транзакцій щохвилини, і їхня повна обробка вимагає оптимізації процесів. У таких випадках застосовують потокову обробку, кешування та механізми асинхронної обробки, що дозволяють зменшити затримки та уникнути перевантаження системи. Додатково інколи використовують механізми попереднього агрегування даних, як-от створення індексів активності або зведених таблиць, що зменшують обсяг повторних обчислень.

У сукупності ці технології утворюють багаторівневу інфраструктуру, яка забезпечує ефективний доступ до блокчейн-даних та їхню підготовку до аналітичної обробки. Саме якісний збір і очищення інформації визначають точність моделей прогнозування та можливість відстеження поведінкових патернів великих гаманців, що є центральним елементом інтелектуальної системи, розробка якої здійснюється в межах цієї роботи.

2.4 Аналіз методів машинного навчання та нейронних мереж, що застосовуються в трейдингу

Використання машинного навчання в аналізі фінансових ринків стало одним із найпомітніших трендів останнього десятиліття, оскільки алгоритми здатні знаходити приховані закономірності у даних, які не піддаються традиційному математичному моделюванню. Особливо це стосується криптовалютного ринку, де висока волатильність та нестабільність часто

роблять класичні індикатори недостатньо точними. Методи машинного навчання дозволяють опрацьовувати великі масиви історичних і ончейн-даних, виділяти характерні патерни поведінки учасників ринку й формувати більш точні прогнози щодо майбутніх цінових рухів.

Одними з перших моделей, які почали застосовувати у фінансовому прогнозуванні, були методи регресії. Вони дозволяють оцінювати ймовірність зростання або падіння ціни, ґрунтуючись на наборі пояснювальних змінних. Проте для криптовалют така модель часто виявляється занадто простою, оскільки ринки характеризуються нелінійною динамікою. Саме це стало причиною переходу до алгоритмів, здатних працювати зі складнішими залежностями: дерев рішень, ансамблевих методів, випадкових лісів та градієнтного бустингу. Ці моделі дозволяють аналізувати десятки або сотні ознак одночасно, проте їхня здатність прогнозувати час, коли саме відбудеться зміна тренду, обмежена.

Поява глибинного навчання відкрила новий етап розвитку алгоритмічного трейдингу. Нейронні мережі, побудовані на принципі багаторівневого перетворення ознак, здатні автоматично виділяти корисну інформацію з сирих даних і будувати математичні представлення, які максимально відповідають структурі часових рядів. Серед них особливо виділяються рекурентні архітектури, такі як LSTM та GRU. Ці моделі враховують послідовність подій, пам'ятають попередні стани ринку та здатні встановлювати залежності на довгих часових проміжках. Для криптовалютних ринків це має критичне значення, оскільки вплив певних подій або транзакцій може проявлятися не миттєво, а через кілька хвилин чи годин. [11]

Для наочного пояснення принципів роботи рекурентних нейронних мереж доцільно розглянути їхню типову архітектуру. Саме структура мережі визначає здатність моделі зберігати інформацію про попередні стани та враховувати часову залежність даних.

Архітектуру LSTM-моделі наведено на рисунку 2.3.

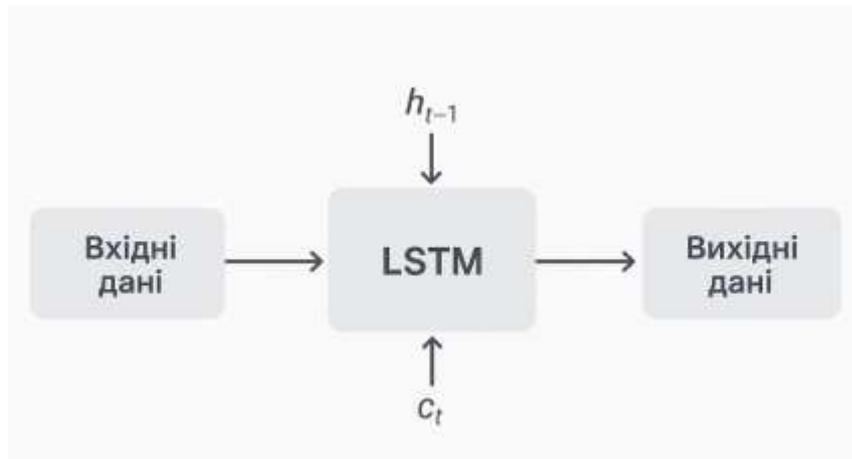


Рисунок 2.3 – Архітектура LSTM-моделі

Ще більш сучасними є трансформерні моделі, які останніми роками активно використовуються не лише у роботі з текстами, а й у фінансовому прогнозуванні. Трансформери працюють за принципом механізму уваги, що дозволяє моделі визначати, які частини історії ринку найбільше впливають на поточний стан. Це дає змогу аналізувати складні нелінійні патерни, що часто виникають унаслідок дій великих гаманців або синхронних переміщень у межах DeFi-протоколів. Завдяки цьому трансформери можуть формувати більш точні прогнози порівняно з класичними RNN-архітектурами. [12]

Не менш важливим напрямом є використання методів кластеризації та нелінійного зниження розмірності, таких як t-SNE або UMAP. Вони дозволяють виявляти приховані групи гаманців або патерни ринкової активності, які не можна побачити безпосередньо. Іноді такі моделі використовують не для прогнозування ціни, а для виявлення аномалій, інформаційних піків або різких стрибків у поведінці учасників ринку. У контексті аналізу великих гаманців ці методи стають особливо корисними, оскільки дозволяють простежити незвичні транзакційні ланцюжки або підозрілі рухи коштів, які можуть передувати значним змінам на ринку. [13]

Незважаючи на високу ефективність, застосування машинного навчання у трейдингу пов'язане з певними викликами. Моделі часто чутливі до шумів, які є характерною рисою криптовалютного ринку. Крім того, виникає проблема

коли мережа занадто точно підлаштовується під історичні дані та втрачає здатність прогнозувати майбутнє. Саме тому особливу увагу приділяють регуляризації, валідації та тестуванню моделей на реальних даних.

2.5 Обґрунтування вибору методів та інструментальних засобів для розробки інтелектуальної системи

Загальну архітектуру інтелектуальної системи наведено на рисунку 2.4

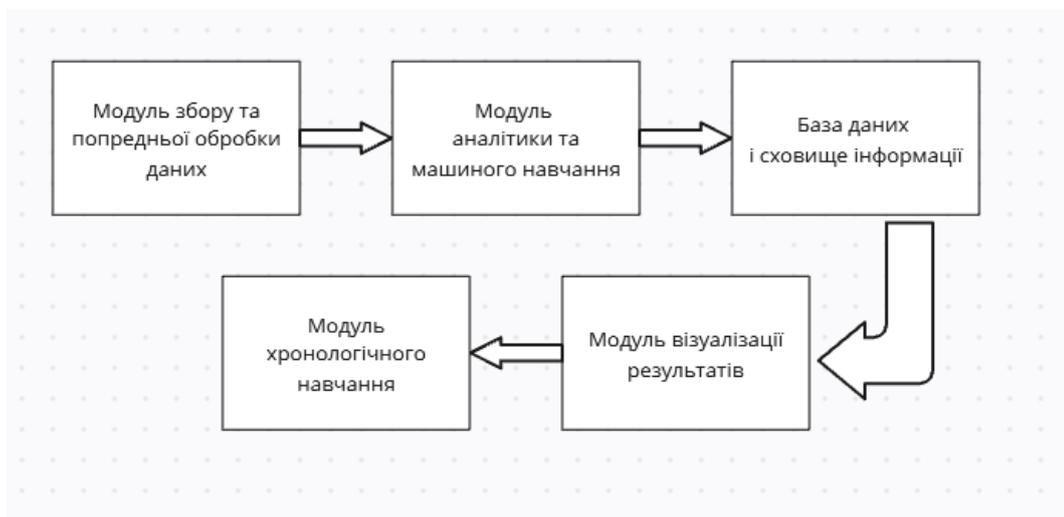


Рисунок 2.4 – Загальна архітектура інтелектуальної системи

Створення інтелектуальної системи, вимагає поєднання технологій, що забезпечують швидкість обробки, точність прогнозування та надійність роботи в умовах постійно змінюваного ринку. Вибір конкретних інструментальних засобів визначається не лише їхніми технічними характеристиками, а й тим, наскільки вони відповідають специфіці блокчейн-даних, їхньому обсягу, структурі та способу отримання. [14]

Першим аспектом є мова програмування та програмне середовище. Більшість сучасних систем ончейн-аналітики реалізується на Python, що й стало базовим вибором у цій роботі. Python має розвинену екосистему бібліотек для

машинного навчання, обробки даних та взаємодії з блокчейном. Наявність Web3.py спрощує доступ до вузлів Ethereum та аналогічних мереж, тоді як Pandas і NumPy дають змогу ефективно працювати з великомасштабними наборами транзакцій. Мова також має низку готових інструментів для створення прототипів моделей, що дозволяє швидко перевіряти гіпотези без залучення складних або спеціалізованих рішень. [15]

Другим важливим елементом є бібліотеки для побудови моделей машинного навчання. У межах цього проєкту доцільним є використання таких фреймворків, як PyTorch або TensorFlow, які забезпечують необхідну гнучкість і високу продуктивність. Ці інструменти дозволяють реалізувати як класичні рекурентні мережі, так і більш сучасні трансформерні архітектури, що особливо корисно в аналізі часових рядів та складних ринкових патернів. Перевагою є широкий вибір оптимізаторів, регуляризаторів та механізмів валідації, що дозволяє адаптувати моделі під конкретні типи даних і задач.

Окремої уваги потребує питання збору та зберігання даних. Оскільки блокчейн генерує величезні обсяги інформації, для системи важливо мати високопродуктивне сховище, яке може обробляти великі масиви транзакцій без значних затримок. У цьому контексті доцільним є використання PostgreSQL або подібних реляційних баз даних, які добре працюють із структурованими даними й дозволяють легко виконувати складні запити. Водночас для швидкого доступу до агресивних транзакційних потоків можуть використовуватися кеш-системи або окремі сервіси попередньо індексованих даних. Такий підхід дозволяє поєднувати точність блокчейн-інформації з ефективністю її обробки. [16]

Не менш важливим є вибір засобів для інтеграції та автоматизації. Для системи, яка працює з реальними даними, необхідно забезпечити стабільну обробку потоків транзакцій, регулярне оновлення моделей та контроль працездатності модулів. З цією метою використовуються інструменти контейнеризації на кшталт Docker або робочі середовища з підтримкою серверних скриптів, які дозволяють автоматизувати оновлення, збір даних, періодичне навчання моделі та тестування системи. Це дає змогу підтримувати

проект у актуальному стані та уникати проблем, пов'язаних із несинхронністю даних або застарілими версіями модулів. [17]

Вибір саме цих інструментів обумовлений тим, що вони забезпечують баланс між продуктивністю, гнучкістю та доступністю. Python і сучасні ML-фреймворки дозволяють створювати моделі будь-якої складності, PostgreSQL забезпечує стабільне зберігання даних, а Web3-сервіси дозволяють підключитися до блокчейну без необхідності власного повного вузла. Усе це створює умови для побудови інтелектуальної системи, здатної працювати з реальними ринковими даними, адаптуватися до їхніх змін та забезпечувати формування релевантних торгових сигналів.

2.6 Висновки до розділу

В даному розділі було розглянуто підходи, інструменти та методи, які становлять теоретичну й технологічну основу для створення інтелектуальної системи аналізу ончейн-даних. Дослідження існуючих способів ідентифікації великих гаманців показало, що ринкова поведінка так званих «китів» є багаторівневим явищем, яке не можна пояснити лише абсолютними значеннями балансів. Вирішальне значення мають поведінкові патерни, характер взаємодії з протоколами та структурна близькість адрес, що зумовлює необхідність використання як кластеризаційних підходів, так і моделей машинного навчання для побудови точних профілів активності.

Огляд технологій збору блокчейн-даних дав змогу визначити, що ефективність роботи всієї системи значною мірою залежить від коректного вибору інструментів доступу до блокчейну та архітектури зберігання інформації. Виявилось, що поєднання Web3-бібліотек, API-сервісів та індексованих структур забезпечує оптимальний баланс між швидкістю обробки й точністю отримуваних даних, а етап попереднього очищення та уніфікації інформації відіграє ключову роль у формуванні якісної вибірки для подальшого аналізу.

Дослідження методів машинного навчання показало, що сучасні фінансові моделі, особливо трансформерні архітектури та рекурентні мережі, здатні краще відтворювати нелінійну динаміку ринку, порівняно з класичними алгоритмами. Їхня здатність працювати з великими часовими інтервалами та складними патернами відкриває можливість формування більш точних прогнозів і попередження значних ринкових рухів, які часто зумовлені активністю великих гаманців.

Обґрунтований вибір інструментальних засобів дав змогу сформувати технологічний фундамент майбутньої системи. Використання Python, PyTorch або TensorFlow, PostgreSQL та Web3-сервісів дозволяє забезпечити гнучкість, масштабованість і високу продуктивність системи, а також гарантує, що її модулі можуть працювати зі справжніми потоками даних у реальному часі.

У сукупності проведені дослідження створюють цілісну картину методологічних та технологічних рішень, необхідних для проектування інтелектуальної системи аналізу ончейн-активності. Цей розділ формує теоретичне підґрунтя для подальшої розробки архітектури системи та реалізації алгоритмів у наступному розділі.

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА АЛГОРИТМІВ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Загальна концепція та функціональні вимоги до системи

Розроблена система створювалась для автоматизації процесу, який зазвичай потребує багато часу: збирання адрес учасників ринку, отримання їхніх торгових позицій та формування узагальнених сигналів. У криптовалютних мережах обсяг доступних даних величезний, але він подається у сирому вигляді, без структури та логічного зв'язку. Тому головна ідея системи полягає в тому, щоб перетворити цей потік інформації на зрозумілий набір показників, що дійсно відображають поведінку великих гравців. [18]

Основна концепція передбачає, що користувач надає лише початковий список адрес у звичайному текстовому файлі, а всі наступні етапи — пошук фоловерів, обробка транзакцій, розрахунок позицій та формування сигналів — виконуються автоматично. Важливим вимогами було те, щоб система не висувала жорстких умов до формату файлу: адреси можуть бути впорядковані як завгодно, записані у різних стилях або містити дублікати. Програма повинна сама привести їх до єдиного вигляду.

Ще одна ключова вимога — здатність працювати з великими вибірками. Оскільки деякі адреси мають десятки фоловерів, а кожен з них — власний набір позицій, система повинна обробляти сотні запитів до API без повільного завантаження чи зависань. Тому логіка роботи базується на асинхронних запитах і поступовому оновленні індикаторів прогресу, щоб користувач одразу бачив перебіг обробки. [19]

Серед функціональних вимог особливо важливими були точність і структурованість підсумкових даних. Система не лише збирає позиції, а й виконує їх агрегацію: об'єднує дрібні трейди, підраховує сумарні значення для кожної монети, визначає чисту позицію (лонг чи шорт) та формує висновок у

вигляді торгового сигналу. При цьому результати подаються в табличній формі, де кожна монета має чітке позначення напрямку, обсягу та коментаря.

У підсумку загальна концепція системи полягає у створенні інструменту, здатного замінити ручний збір даних і надати трейдеру якісно підготовлену аналітику. Система слугує “містком” між блокчейн-даними та процесом прийняття торгових рішень, надаючи користувачу впорядкований набір метрик, який легко інтерпретувати. [20]

3.2 Структура програмного забезпечення та взаємодія основних модулів

Створюючи систему, яка має автоматично збирати адреси користувачів, звертатись до API та формувати підсумкові торгові сигнали, важливо було побудувати архітектуру так, щоб вона залишалася простою для підтримки, але водночас гнучкою й достатньо швидкою при роботі з великими обсягами даних. Тому вся логіка застосунку була поділена на кілька незалежних модулів, кожен з яких виконує своє завдання та мінімально впливає на інші частини системи.

Першим модулем є блок завантаження та нормалізації адрес. Він працює безпосередньо з текстовим файлом, який користувач додає в інтерфейсі. Модуль автоматично зчитує файл, “чистить” дані від зайвих пробілів, переводить адреси у потрібний формат та видаляє дублікати. Від якості цієї частини залежить подальша робота всієї системи, тому вона орієнтована на максимально коректне формування початкового списку адрес. [21]

Другим ключовим елементом є модуль взаємодії з API Hyperliquid. Саме він відповідає за отримання всіх даних про трейдерів — фоловерів, лідерів та їхні торгові позиції. Робота з API організована асинхронно: поки система отримує дані для однієї адреси, вона вже може починати обробку іншої. Це дозволяє досягати стабільної швидкості навіть тоді, коли список містить сотні або тисячі

адрес. Програма паралельно оновлює індикатори прогресу, щоб користувач бачив, як просувається збір інформації.

Схему загальної архітектури платформи наведено на рисунку 3.1

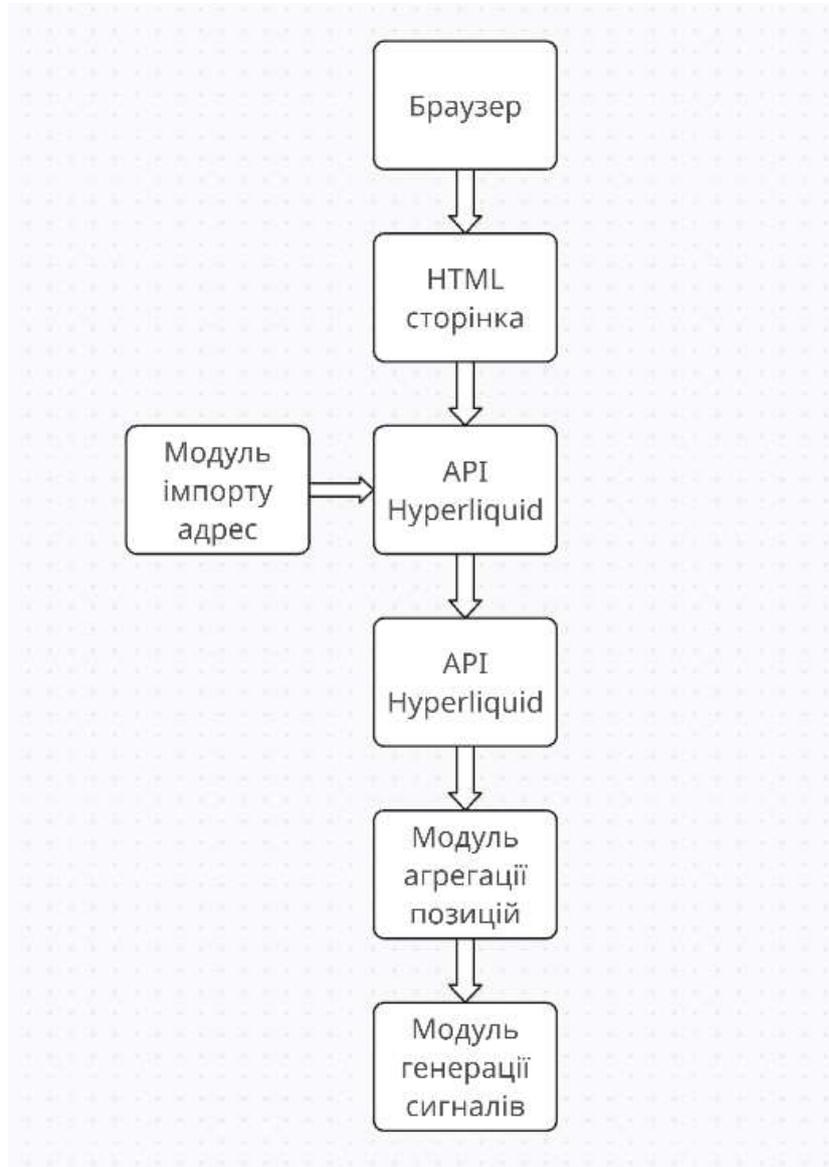


Рисунок 3.1 – Схема загальної архітектури платформи

Після отримання сирих даних система передає їх у модуль агрегації позицій. Тут відбувається ключова частина обробки: позиції для кожної монети групуються, відокремлюються лонги від шортів, розраховується нетто-розмір позиції, загальний обсяг і середні значення. Дрібні угоди автоматично зливаються у групи **All_small** Long/Short, що дозволяє прибрати ринковий “шум”

та не перевантажувати таблиці непотрібними деталями. У результаті формується чітка картина, де видно, в який бік зміщується баланс сили для кожного активу.

Далі дані надходять у модуль генерації сигналів. Він порівнює нетто-позицію з загальним обсягом торгів по монеті та визначає, чи є статистично значима перевага лонгів або шортів. Цей модуль також формує текстовий коментар до кожного сигналу — від сильної домінації до слабкої переваги, залежно від заданих порогових значень.

Завершальний етап — візуалізація. Модуль інтерфейсу формує таблиці лідерів, позицій та торгових сигналів, оновлюючи їх одразу після завершення розрахунків. Оскільки всі обчислення виконуються всередині браузера, користувач отримує результат миттєво, без сторонніх серверів і сторонніх залежностей. [22]

Запропонована модульна архітектура забезпечує не лише зручність супроводу та розширення системи, а й підвищує її надійність у процесі експлуатації. Оскільки кожен модуль виконує чітко визначену функцію, можливі зміни або вдосконалення окремих компонентів не потребують повного перепроєктування платформи. Такий підхід дозволяє адаптувати систему до змін у структурі API, розширювати перелік аналізованих активів або інтегрувати додаткові аналітичні блоки без порушення загальної логіки роботи.

Крім того, обрана архітектура створює основу для подальшого розвитку інтелектуальних можливостей системи. Зокрема, у майбутньому можливе доповнення модуля генерації сигналів методами машинного навчання або використанням додаткових ончейн-індикаторів для підвищення точності оцінок. При цьому збереження чіткої межі між етапами збору, обробки та візуалізації даних дозволяє масштабувати рішення та застосовувати його в інших торгових або аналітичних сценаріях.

У підсумку структура системи — це набір чітко окреслених модулів, які разом утворюють цілісний інструмент: від завантаження простого текстового файлу до формування повного аналітичного звіту про ринкові позиції й готових торгових сигналів.

3.3 Механізм імпорту та нормалізації адрес

Щоб система могла аналізувати торгову активність різних гаманців, їй спочатку необхідно отримати вихідний список адрес, з якими надалі працюватиме алгоритм. На практиці джерелом таких даних є текстові файли, сформовані користувачем або зовнішніми сервісами. Вхідні файли можуть мати різний формат — від простого набору адрес, розділених комами, до списків, сформованих у стовпчик. Тому перший етап роботи модуля полягає у тому, щоб коректно зчитати файл, розпізнати всі вміщені в ньому індекси та привести їх до єдиного вигляду.

Імпорт здійснюється безпосередньо в інтерфейсі веб-програми: користувач завантажує файл через форму, після чого текст обробляється за допомогою вбудованого механізму FileReader. На цьому етапі система усуває зайві пробіли, переходи на новий рядок, а також неправильно введені символи, які могли з'явитися під час ручного копіювання. Усі записи автоматично нормалізуються до стандартної форми адреси — у форматі **0x...**, довжиною 42 символи. [23]

Після очищення та нормалізації виконується ще один важливий крок — усунення дублікатів. Часто в реальних списках трапляються повтори, або одна й та сама адреса може бути одночасно вказана у файлі та введена вручну в окремому полі. Завдяки автоматичному використанню структури множини дублікати видаляються ще до основної аналітичної обробки. Це дозволяє уникнути спотворення даних, яке могло б виникнути, наприклад, через повторне обчислення позицій одного й того самого трейдера. [24]

Оскільки процес імпорту та нормалізації адрес складається з кількох послідовних кроків, доцільно представити його у вигляді наочної схеми. Це дозволяє чітко відобразити логіку обробки вхідних даних — від моменту завантаження файлу користувачем до формування уніфікованого списку адрес, готового до подальшого аналізу.

Узагальнений алгоритм роботи модуля імпорту та нормалізації адрес наведено на рисунку 3.2.

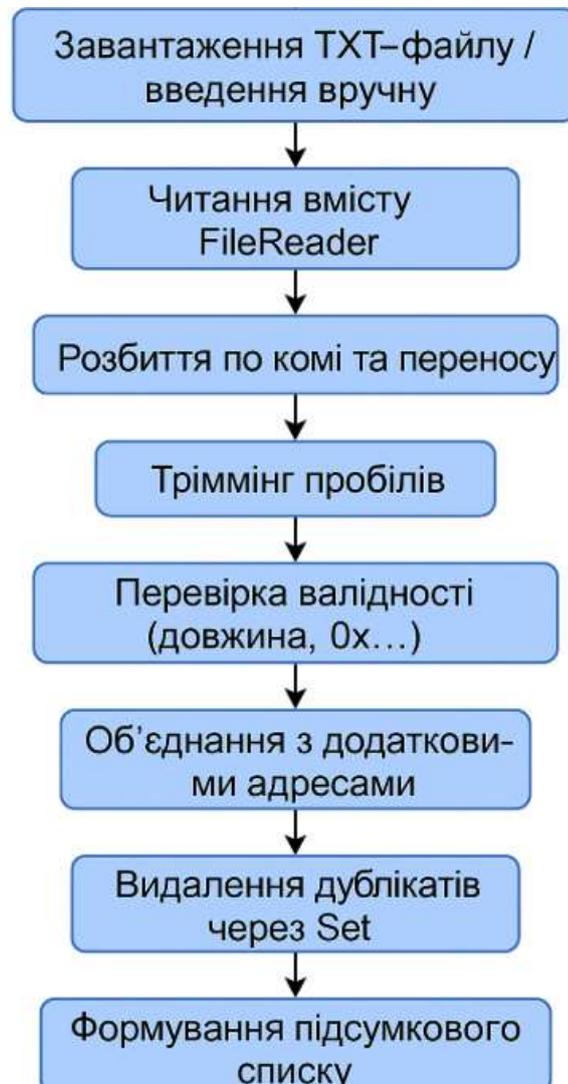


Рисунок 3.2 – Схема обробки адрес (flowchart)

Такий попередній етап нормалізації робить подальший аналіз значно надійнішим. Адреси, які не відповідають формату — коротші, довші або без префікса **0x**, — також відсіюються, щоб у систему не потрапляли некоректні записи. У результаті модуль формує чистий, структурований та уніфікований список унікальних адрес, який далі передається в API-запити для завантаження реальних торгових даних кожного користувача. [25]

3.4 Інтеграція інтелектуальної системи з API Hyperliquid

Щоб система могла працювати не лише як автономний інструмент, а як повноцінний аналітичний модуль, їй потрібне постійне підключення до актуальних ринкових даних. У цьому проєкті основним джерелом інформації виступає API платформи Hyperliquid, яке надає відкритий доступ до стану користувачьких позицій, даних торгових гаманців та інформації про динаміку відкритих контрактів. Саме завдяки цьому API система може “бачити” ринкові зміни практично в реальному часі, фрагмент API-запиту наведено на рисунку 3.3.

```
await fetch('https://api.hyperliquid.xyz/info', {  
  method: 'POST',  
  body: JSON.stringify({ type: 'clearinghouseState', user: address })  
});
```

Рисунок 3.3 – Фрагмент API-запиту

Інтеграція побудована максимально просто й надійно. Після того як користувач завантажує файл з адресами, система формує список гаманців, включаючи як основні адреси, так і всіх їхніх фоловерів. Для кожної окремої адреси надсилається запит типу **clearinghouseState**, який повертає поточні позиції: символи активів, напрям (лонг або шорт), розмір, середню ціну входу, нереалізований PnL та рівень ліквідації. Це та інформація, яка дозволяє будувати “зріз” ринку в конкретний момент часу. [26]

Окремо використовуються запити **vaultDetails**, оскільки вони розкривають структуру підписників певних великих трейдерів. Такий підхід дає змогу розширювати вибірку не вручну, а автоматично — система знаходить усіх, хто повторює торгівлю ключових гаманців, і додає їх у подальший аналіз. Це спрощує роботу користувача й підвищує достовірність даних, оскільки охоплення торгуючих учасників стає ширшим.

Уся логіка інтеграції побудована так, щоб система могла коректно працювати як із десятком адрес, так і з великими масивами даних. Для цього використовується асинхронний збір інформації: кожен запит виконується незалежно, а результати послідовно об'єднуються у спільний пул позицій. Після того як усі запити завершені, дані автоматично агрегуються, очищуються та переходять на наступний етап — аналіз обсягів, розрахунок чистих позицій та формування торгових сигналів. [27]

Таким чином, інтеграція з API Hyperliquid стала ключовою частиною всієї системи. Вона дає доступ до фактичної поведінки ринку, мінімізує затримки оновлень і забезпечує той рівень деталізації, який неможливо отримати зі звичайних агрегованих біржових індикаторів.

3.5 Алгоритм формування торгових сигналів

Після того як система збирає й упорядковує всі позиції від вибраних адрес, починається найважливіший етап — перетворення сирих даних на зрозумілі ринкові сигнали. Цей модуль фактично відповідає за інтелектуальну частину всієї розробки, адже саме він визначає, куди в сукупності “дивляться” великі гравці — у лонг чи в шорт — і з якою силою. [28]

Алгоритм працює поетапно. Спершу для кожної монети обчислюється сукупний чистий обсяг (**Net size**): сума всіх лонгів мінус сума всіх шортів. Поруч із цим рахується і загальний обсяг позицій (**Total volume**), тобто всі відкриті позиції незалежно від напрямку. Ці два значення дозволяють визначити співвідношення, яке у програмі відображене як частка абсолютного **Net size** від **Total volume**. Чим ця частка більша, тим сильніший ринковий перекис.

На базі цих показників система застосовує порогові рівні. Якщо чистий обсяг суттєво переважає ($> 30\%$), то формується сигнал з коментарем про домінування певної сторони — як правило, "Домінує чистий лонг" або "Домінує чистий шорт". Якщо ж перевага є, але не така значна, тобто ($<30\%$), то сигнал

уже обережніший і характеризується як слабка або легка перевага, і не відображається. [29]

Фінальний сигнал складається з кількох елементів: назви активу, напрямку (**LONG / SHORT / NEUTRAL**), розміру чистого перекосу, загального обсягу та короткого текстового висновку. Після цього все потрапляє у таблицю торгових сигналів на сайті. Завдяки повністю автоматичній логіці система видає сигнал саме тоді, коли з'являється реальний зсув у поведінці великих гравців, а не просто через випадкові коливання.

3.6 Логіка агрегації позицій та формування інтегрованої картини ринку

У цьому пункті розглядається фінальна частина внутрішнього алгоритму системи — той самий механізм, який перетворює розрізнені позиції сотень або навіть тисяч трейдерів у цілісне ринкове зведення. Сама процедура на практиці виявилась однією з найскладніших, адже дані, що надходять від API, значно відрізняються за масштабом та рівнем деталізації. Наприклад, в одному гаманці може бути одна велика позиція, тоді як інший користувач розбиває торгівлю на десятки дрібних ордерів. Якщо їх не нормалізувати, сигнал втрачає сенс. [30]

Логіка працює в кілька кроків. Після того як усі адреси пройшли через модуль завантаження й фільтрації, окремі позиції починають групуватися за торговими парами та напрямком (лонг / шорт). Такий підхід дозволяє оцінити реальний обсяг сили на ринку: скільки капіталу підтримує зростання ціни та який обсяг працює на зниження. Позиції, що не проходять розмірний поріг, не викидаються, а переносяться в узагальнені категорії All_small Long та All_small Short. Це важливо, бо дрібні трейдери все одно впливають на ринок, хоча й не так суттєво, як великі гаманці.

Коли всі дані “зібрані під спільним дахом”, система обчислює три ключові величини:

- total volume — загальний обсяг торгівлі по кожній монеті;
- net size — чистий баланс між лонгами та шортами;
- direction strength — відносна перевага одного боку ринку, що вимірюється часткою net size у загальному обсязі.

Логіку агрегації позицій наведено на рисунку 3.4

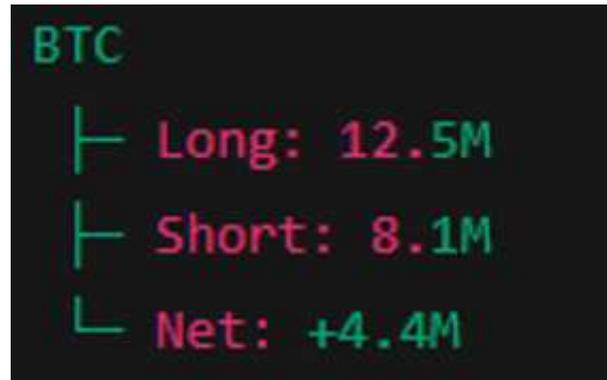


Рисунок 3.4 – Логіка агрегації позицій

Саме на цьому етапі формується фінальна картина. Якщо чиста позиція суттєво зміщується в один бік, монета потрапляє до таблиці сигналів, яку бачить користувач. Якщо дисбаланс помірний — коментар змінюється на більш стриманий, що дозволяє інтерпретувати поведінку ринку з урахуванням невизначеності. Таким чином, система не просто відображає сухі числа, а “розкладає” ринок на зрозумілі тенденції, показуючи, хто наразі сильніший — покупці чи продавці, і наскільки саме.

3.7 Висновки до розділу

У цьому розділі була сформована цілісна архітектура інтелектуальної системи та описані всі ключові механізми, які забезпечують її роботу — від первинного збору даних і нормалізації адрес до інтеграції з API та побудови алгоритмів, що визначають торгові сигнали. Структура програмного

забезпечення виявилася достатньо гнучкою, щоб працювати з великим масивом нестандартних даних і, водночас, досить простою для розширення новими модулями.

Окремо було розглянуто логіку агрегації позицій та методи формування чистих значень (Net size) для кожного активу, адже саме ці розрахунки визначають кінцевий зміст торгового сигналу. Завдяки поєднанню індивідуальних позицій великих гаманців та агрегованих показників дрібних учасників система здатна давати цілісне уявлення про баланс сил — ситуації, де на ринку домінують покупці, продавці чи відсутня чітка перевага.

Побудований алгоритмічний ланцюжок дозволяє автоматично проходити через усі етапи — від імпорту адрес до фінального аналізу — що робить систему самодостатнім інструментом оперативного моніторингу ринкової активності. У підсумку, розроблена архітектура показує, що така система може слугувати надійною основою для подальшої імплементації: автоматичного попередження трейдерів, побудови моделей прогнозування, а згодом — повноцінних торгових стратегій на основі поведінки великих гравців. [31]

4 ОПИС РЕАЛІЗАЦІЇ СИСТЕМИ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ

4.1 Технологічний стек та середовище розробки

Розроблена платформа складається з веб-інтерфейсу та серверної частини, які працюють разом для збору, обробки й відображення даних. Основна логіка виконується безпосередньо у браузері користувача, що значно спрощує розгортання системи — достатньо статичної HTML-сторінки, яка може працювати без окремого бекенду. [32]

Основні технології:

а) HTML5: використовується для структурування інтерфейсу. Сторінка містить елементи вводу файлів, таблиці для відображення лідерів, позицій та торгових сигналів, а також панель прогресу. HTML став основою всієї взаємодії користувача з програмою.

б) CSS3: відповідає за візуальний стиль системи: таблиці, відступи, кольорове виділення прибуткових/збиткових позицій, індикатори прогресу. Хоч система не використовує складних UI-фреймворків, базової стилізації достатньо для комфортної роботи та читабельності даних.

в) JavaScript: є ключовою технологією системи. Він забезпечує:

- імпорт та парсинг файлу з адресами;
- запити до API Hyperliquid;
- збір та агрегацію позицій сотень/тисяч користувачів;
- обчислення Net size, Long/Short спредів, співвідношення сил;
- формування торгових сигналів;
- динамічне оновлення таблиць без перезавантаження сторінки.

Уся логіка виконується локально, що дозволяє уникнути серверних затримок і спрощує експлуатацію.

г) Hyperliquid API: система інтегрується з офіційним API біржі Hyperliquid.

Через нього отримуються:

- дані про фоловерів вказаних адрес;
- відкриті позиції кожного користувача;
- обсяг, напрямок (Long/Short), плече та інші параметри.
- API є основним джерелом ринкової інформації для розрахунку сигналів.

г) JSON: використовується як формат передачі даних між JavaScript і API Hyperliquid. Завдяки цьому всі відповіді легко обробляти та конвертувати у зручні таблиці.

Обраний технологічний стек дозволив створити інструмент, який:

- працює автономно та не потребує серверної інфраструктури;
- може обробляти великі масиви даних без додаткових витрат;
- має простий інтерфейс, який легко розширювати;
- взаємодіє з зовнішніми сервісами в реальному часі.

Такий підхід робить систему максимально простою в розгортанні, надійною, швидкою та зручною у використанні. [34]

4.2 Реалізація модулів системи

Реалізація програмної системи відбувалася поетапно, відповідно до структури, визначеної у третьому розділі. Кожний функціональний модуль був розроблений таким чином, щоб забезпечувати максимальну автономність, простоту тестування та можливість подальшого масштабування.

Ключові модулі:

а) Модуль завантаження та нормалізації адрес

Перший етап роботи системи — це отримання списку адрес від користувача. Реалізовано два варіанти введення: завантаження **.txt**-файлу або

ручне введення у текстове поле. Модуль обробляє файл, розбиває текст за комами або перенесеннями рядків, очищує значення від пробілів і створює уніфікований масив адрес. На цьому етапі також виконується перевірка коректності форматів (довжина та префікс 0x) та видалення дублікатів.

Модуль завантаження та нормалізації адрес наведено на рисунках 4.1 – 4.4

```
document.getElementById('addressFile').addEventListener('change', function(e) {
  const file = e.target.files[0];
  const fileNameDisplay = document.querySelector('.file-name');
  if (file) {
    fileNameDisplay.textContent = file.name;
    const reader = new FileReader();
    reader.onload = function(e) {
      const text = e.target.result;
      const addresses = text
        .split(/,|\n/)
        .map(addr => addr.trim())
        .filter(addr => addr.length > 0);
      fetchData(addresses);
    };
    reader.readAsText(file);
  } else {
    fileNameDisplay.textContent = 'No file chosen';
  }
});
```

Рисунок 4.1 – Модуль завантаження файла

```
function saveAddressesFile() {
  const vaultAddresses1 = document.getElementById('vaultAddresses').value
    .split(',')
    .map(addr => addr.trim())
    .filter(addr => addr.length > 0);

  const vaultAddresses2 = document.getElementById('vaultAddresses2').value
    .split(',')
    .map(addr => addr.trim())
    .filter(addr => addr.length > 0);

  const uniqueAddresses = [...new Set([...vaultAddresses1, ...vaultAddresses2])];

  if (uniqueAddresses.length === 0) {
    alert('No addresses to save!');
    return;
  }
}
```

Рисунок 4.2 – Модуль нормалізації адрес (частина 1)

```

const fileContent = uniqueAddresses.join(',\n');

const blob = new Blob([fileContent], { type: 'text/plain;charset=utf-8' });
const url = window.URL.createObjectURL(blob);
const a = document.createElement('a');
a.href = url;
a.download = 'unique_addresses.txt';
document.body.appendChild(a);
a.click();
document.body.removeChild(a);
window.URL.revokeObjectURL(url);
}

```

Рисунок 4.3 – Модуль нормалізації адрес (частина 2)

Upload Address File: No file chosen

Рисунок 4.4 – Інтерфейс модуля завантаження

б) Модуль взаємодії з API Hyperliquid

Другим основним елементом є асинхронний модуль роботи з API. Він виконує два типи запитів:

– Отримання інформації про vault та його підписників (**vaultDetails**).

Система збирає фоловерів для кожної з введених адрес і формує їх повний список.

– Отримання позицій кожного користувача (**clearinghouseState**).

Для кожної адреси запитуються активні лонг- та шорт-позиції по всіх торгових парах. [35]

Щоб уникнути навантаження на API та зависань інтерфейсу, всі запити виконуються асинхронно, із поетапним оновленням статусу на екрані.

Модуль взаємодії з API Hyperliquid наведено на рисунку 4.5

```

async function getUserPositions(address) {
  const sizeFilter = parseFloat(
    document.getElementById('positionSizeFilter').value.replace(/\s/g, '')
  ) || DEFAULT_SIZE_FILTER;

  try {
    const res = await fetch('https://api.hyperliquid.xyz/info', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({
        type: 'clearinghouseState',
        user: address
      })
    });

    const data = await res.json();

    return (data.assetPositions || []).map(pos => {
      const position = {
        address,
        coin: pos.position.coin,
        dir: parseFloat(pos.position.szi) > 0 ? 'Long' : 'Short',
        value: parseFloat(pos.position.positionValue),
        leverage: parseFloat(pos.position.leverage?.value || 1).toFixed(0),
        entryPx: formatPrice(pos.position.entryPx),
        unrealizedPnl: parseFloat(pos.position.unrealizedPnl || 0).toFixed(0),
        liquidationPx: formatPrice(pos.position.liquidationPx || 0)
      };

      if (Math.abs(position.value) < sizeFilter) {
        position.address = `All_small ${position.dir}`;
      }

      return position;
    });
  } catch {
    return [];
  }
}

```

Рисунок 4.5 – Модуль взаємодії з API Hyperliquid

в) Модуль агрегації позицій

Отримані по кожній адресі позиції об'єднуються в одну структуру даних, де:

- суми long і short значень підраховуються окремо;
- дрібні позиції (< 10 млн) групуються у категорію All_small;
- для кожного активу фіксується загальний обсяг торгів та чиста позиція (нетто-розмір).

Цей модуль відповідає за формування фінальної картини розподілу ринкової сили між покупцями та продавцями.

Модуль агрегації позицій наведено на рисунках 4.6 – 4.8

```
function aggregateSmallPositions(allPositions, sizeFilter) {
  const grouped = {};
  const aggregated = [];

  allPositions.forEach(pos => {
    const key = `${pos.coin}|${pos.dir}`;

    if (Math.abs(pos.value) < sizeFilter) {
      if (!grouped[key]) {
        grouped[key] = {
          coin: pos.coin,
          dir: pos.dir,
          value: 0,
          unrealizedPnl: 0,
          totalValue: 0,
          totalCoins: 0,
          sumValueOverLeverage: 0
        };
      }
    }
  })
}
```

Рисунок 4.6 – Модуль агрегації позицій (частина 1)

```
if (Math.abs(pos.value) < sizeFilter) {  
  if (!grouped[key]) {  
    grouped[key] = {  
      coin: pos.coin,  
      dir: pos.dir,  
      value: 0,  
      unrealizedPnl: 0,  
      totalValue: 0,  
      totalCoins: 0,  
      sumValueOverLeverage: 0  
    };  
  }  
  
  grouped[key].value += pos.value;  
  grouped[key].unrealizedPnl += parseFloat(pos.unrealizedPnl);  
  grouped[key].totalValue += pos.value;  
  grouped[key].totalCoins += pos.value / parseFloat(pos.entryPx.replace(/\\/s/g, '').replace(',', ''));  
  grouped[key].sumValueOverLeverage += pos.value / parseFloat(pos.leverage || 1);  
  
} else {  
  aggregated.push(pos);  
}  
});
```

Рисунок 4.7 – Модуль агрегації позицій (частина 2)

```

for (const key in grouped) {
  const g = grouped[key];
  const entryPx = g.totalCoins > 0 ? g.totalValue / g.totalCoins : 0;
  const leverage = g.sumValueOverLeverage !== 0
    ? Math.abs(g.value / g.sumValueOverLeverage).toFixed(1)
    : '';

  aggregated.push({
    address: `All_small ${g.dir}`,
    coin: g.coin,
    dir: g.dir,
    value: g.value,
    leverage,
    entryPx: formatPrice(entryPx),
    unrealizedPnl: g.unrealizedPnl.toFixed(0),
    liquidationPx: ''
  });
}

return aggregated;
}

```

Рисунок 4.8 – Модуль агрегації позицій (частина 3)

г) Модуль формування торгових сигналів

На основі сукупності позицій для кожного токена розраховується відношення `net-position` до загального обороту. Якщо абсолютна перевага однієї сторони перевищує визначений поріг (30%), система формує торговий сигнал:

- LONG, якщо домінують покупці;
- SHORT, якщо переважають продавці;
- або не додає актив у список сигналів, якщо баланс сил нейтральний.

Цей модуль генерує фінальний набір сигналів, а також супроводжуючі коментарі (домінує лонг/шорт, слабка перевага тощо). [36]

Модуль формування торгових сигналів наведено на рисунку 4.8

```
function generateSignals(coinNetSizes, coinTotalSizes) {
  const signals = [];
  const LONG_THRESHOLD = 0.3;
  const SHORT_THRESHOLD = 0.3;

  for (const coin in coinNetSizes) {
    const net = coinNetSizes[coin] || 0;
    const total = coinTotalSizes[coin] || 0;
    if (total === 0) continue;

    const netAbs = Math.abs(net);
    const ratio = netAbs / total;

    let direction = 'NEUTRAL';
    let comment = 'Балансований попит і пропозиція';

    if (ratio >= LONG_THRESHOLD && net > 0) {
      direction = 'LONG';
      comment = 'Домінує чистий лонг, перевага покупців';
    } else if (ratio >= SHORT_THRESHOLD && net < 0) {
      direction = 'SHORT';
      comment = 'Домінує чистий шорт, перевага продавців';
    } else {
      continue;
    }

    signals.push({
      coin,
      direction,
      net,
      total,
      comment
    });
  }

  return signals;
}
```

Рисунок 4.8 – Модуль формування торгових сигналів

г) Модуль візуалізації результатів

Усі дані відображаються у таблицях HTML, які генеруються динамічно.

Основні блоки інтерфейсу:

- таблиця ТОП-5 фоловерів;
- таблиця всіх позицій із розподілом по монетах;
- таблиця торгових сигналів.

Цей модуль не тільки показує результати, але й забезпечує коректне форматування чисел, кольорове виділення long/short та оновлення даних у процесі завантаження. [37]

Модуль візуалізації результатів наведено на рисунку 4.9

```
const signalsTableBody = document.querySelector('#signalsTable tbody');
const signalsCountElement = document.getElementById('signalsCount');
signalsTableBody.innerHTML = '';
const signalsFragment = document.createDocumentFragment();

signals.forEach(sig => {
  const row = document.createElement('tr');
  const dirClass = sig.direction === 'LONG' ? 'positive'
    : sig.direction === 'SHORT' ? 'negative'
    : '';
  row.innerHTML = `
    <td>${sig.coin}</td>
    <td class="${dirClass}">${sig.direction}</td>
    <td>${Math.round(Math.abs(sig.net)).toLocaleString('uk-UA')}</td>
    <td>${Math.round(Math.abs(sig.total)).toLocaleString('uk-UA')}</td>
    <td>${sig.comment}</td>
  `;
  signalsFragment.appendChild(row);
});

signalsTableBody.appendChild(signalsFragment);
signalsCountElement.textContent = `(${signals.length} signals)`;

const leadersTableBody = document.querySelector('#topLeadersTable tbody');
leadersTableBody.innerHTML = '';
leadersTableBody.appendChild(leadersFragment);

positionsTableBody.innerHTML = '';
positionsTableBody.appendChild(positionsFragment);

positionCountElement.textContent = `(${sortedPositions.length} positions)`;
progressElement.textContent = '100%';
```

Рисунок 4.9 – Модуль візуалізації результатів

Інтерфейси роботи всіх модулів наведено на рисунках 4.10 – 4.17

Top 5 Leader Followers (All-Time) (4774 addresses) 100%

#	Address	Total Profit	Current PNL	Equity	Days
1	0x2e3d94...a8dd14	43 926 806	-4 576 926	1 000 000	843
2	0xb0a55f...6e9540	12 047 377	11 532 283	30 000 000	224
3	0x010461...1c703a	5 986 999	3 688 411	141 821 872	940
4	0x8196e0...6b5836	1 475 104	748 427	3 772 360	816
5	0x3a648f...4e52d9	934 974	914 722	8 914 722	307

Рисунок 4.10 – Інтерфейс роботи модуля для виведення ТОП-5 фоловерів

Trading signals (6 signals)

Symbol	Direction	Net size (USD)	Total volume (USD)	Comment
HYPE	LONG	212 365 278	448 204 499	Домінує чистий лонг, перевага покупців
BTC	SHORT	380 792 021	977 195 433	Домінує чистий шорт, перевага продавців
ETH	SHORT	269 009 482	716 184 025	Домінує чистий шорт, перевага продавців
SOL	SHORT	120 614 685	215 634 871	Домінує чистий шорт, перевага продавців
PAXG	SHORT	13 653 671	20 032 882	Домінує чистий шорт, перевага продавців
BNB	SHORT	10 322 002	14 186 731	Домінує чистий шорт, перевага продавців

Рисунок 4.11 – Інтерфейс роботи модуля формування торгових сигналів

Positions 100% (85 positions) size greater than:

Address	Symbol	Type	Size	Leverage	Entry	PnL	Liquidation
0x9eec98...81daab	ETH	Long	158 847 146	15	3 201	-8 739 787	2 344
All_small Short	ETH	Short	93 705 115	15.0	3 035	582 966	
0x5b5d51...98c060	ETH	Short	87 320 832	10	3 523	14 074 040	4 344
All_small Long	ETH	Long	78 351 495	9.5	3 053	-982 946	
0x880ac4...aa311c	ETH	Short	35 026 447	12	3 110	870 570	3 936
0x7fdafd...c517d1	ETH	Short	34 483 856	20	3 410	4 259 537	6 092
0xbadbb1...039ee6	ETH	Long	29 740 060	20	3 116	-803 785	1 176
0x020ca6...a35872	ETH	Long	27 890 720	25	2 982	455 700	2 890
0xd47587...151a91	ETH	Short	23 949 154	15	4 015	7 740 309	7 628
0x45d26f...ed4029	ETH	Short	20 019 120	20	2 952	-538 162	6 962
0x856c35...0e910d	ETH	Short	19 414 159	20	3 269	1 496 514	7 186

Рисунок 4.12 – Інтерфейс роботи модуля агрегації позицій (частина 1)

0x175e70...b577b4	ETH	Long	10 967 145	2	4 477	-5 232 398	2 358
0x20c2d9...da44f5	ETH	Short	10 940 916	25	3 402	1 339 818	3 976
0x9c89f5...e6600e	ETH	Long	10 608 500	20	3 085	-189 100	1 230
0xa52090...a5a575	ETH	Long	10 380 746	25	3 522	-1 683 129	2 107
0x208db9...ace716	ETH	Short	10 048 483	20	2 991	-139 615	4 853
Total ETH Short 268 624	ETH		735 093 903			18 940 383	

Рисунок 4.13 – Інтерфейс роботи модуля агрегації позицій (частина 2)

All_small Long	HYPE	Long	115 204 120	6.3	36,06	-8 662 410	
All_small Short	HYPE	Short	82 248 534	6.0	34,56	5 646 493	
0x5b5d51...98c060	HYPE	Short	44 079 601	5	42	10 713 870	60,81
0x7fdafd...c517d1	HYPE	Short	42 011 328	5	37,7	4 859 268	59,7
0xbadbb1...039ee6	HYPE	Long	31 182 814	5	32,86	854 717	12,79
0xb83de0...7d6e36	HYPE	Short	23 871 284	5	42,87	6 416 420	50,22
0x880ac4...aa311c	HYPE	Short	22 995 722	5	37,33	2 410 091	47,98
0x549e6d...dc0533	HYPE	Long	20 289 300	5	33,6	116 841	29,2
0x856c35...0e910d	HYPE	Short	14 083 040	3	34,3	206 003	93,54
0x007d76...d967a0	HYPE	Long	13 520 853	10	37,5	-1 494 493	18,4
0x45d26f...ed4029	HYPE	Short	12 836 780	10	27,63	-2 337 706	97,92
0x76c216...155aa0	HYPE	Short	10 749 751	10	31,32	-782 733	63,54
Total HYPE Short 72 678 954	HYPE		433 073 127			17 946 361	

Рисунок 4.14 – Інтерфейс роботи модуля агрегації позицій (частина 3)

0x9eec98...81daab	XRP	Long	85 404 652	10	2,292	-3 607 752	1,253
All_small Short	XRP	Short	68 428 929	17.6	2,178	-489 457	
0xecb63c...b82b00	XRP	Short	16 270 276	10	2,123	-567 524	5,749
All_small Long	XRP	Long	11 578 148	7.2	2,254	-323 567	
Total XRP Long 12 283 595	XRP		181 682 005			-4 988 300	

Рисунок 4.15 – Інтерфейс роботи модуля агрегації позицій (частина 4)

Total SOL Short 25 497 921	SOL		214 231 288			8 011 181	
All_small Long	ZEC	Long	37 103 480	6.2	495,9	-6 748 976	
All_small Short	ZEC	Short	36 724 186	6.1	505,9	6 782 961	
0xd47587...151a91	ZEC	Short	28 203 733	5	419,2	-1 220 455	960,6

Рисунок 4.16 – Інтерфейс роботи модуля агрегації позицій (частина 5)

All_small Short	FARTCOIN	Short	23 651 838	6.1	0,3393	3 539 420	
All_small Long	FARTCOIN	Long	14 598 748	5.3	0,2686	2 366 237	
Total FARTCOIN Short 9 053 090	FARTCOIN		38 250 586			5 905 657	
All_small Short	XPL	Short	18 907 422	5.5	0,2763	13 777 642	
All_small Short	ENA	Short	18 585 463	7.2	0,327	5 951 306	
All_small Short	MON	Short	14 487 113	3.3	0,03672	3 553 869	
All_small Short	BNB	Short	11 393 916	6.9	932,3	714 786	
All_small Short	DOGE	Short	10 644 017	9.5	0,157	751 312	
0x856c35...0e910d	PAXG	Short	10 467 371	5	4 077	-415 929	14 233
All_small Short	UNI	Short	10 432 977	10.0	7,228	2 038 678	

Рисунок 4.17 – Інтерфейс роботи модуля агрегації позицій (частина б)

4.3 Методика тестування та проведення експериментів

Після реалізації основних модулів системи постало завдання перевірити, наскільки стабільно вона працює під час взаємодії з реальними даними. Тестування проводилося у кілька етапів, кожен з яких дозволяв виявити окремий клас потенційних проблем.

Першим було функціональне тестування. Перевірялося коректне завантаження файлів з адресами, правильність отримання фоловерів через API Hyperliquid, відображення таблиць та поведінка інтерфейсу під час обробки великої кількості записів. На цьому етапі також спеціально застосовувалися некоректні та неповні файли, щоб переконатися, що система адекватно реагує на помилки у вхідних даних. [38]

Наступним кроком було навантажувальне тестування. Система запускалася зі значно збільшеними списками адрес, а також у різних браузерях і на різних пристроях. Це дозволило визначити моменти, де виникають затримки у роботі, а також виявити залежність швидкості від продуктивності API. В результаті було оптимізовано частину логіки агрегації позицій, щоб уникнути «заморожування» інтерфейсу при великих обсягах даних.

Окремо перевірялася точність формування торгових сигналів. Для цього проміжні розрахунки — суми лонгів і шортів, чиста позиція, процентне співвідношення — порівнювалися з контрольними вручну підготовленими значеннями. Такий підхід підтвердив коректність формули та стабільність результатів при зміні порогів.

Фінальний етап мав характер користувацького тестування. Перевірялося, наскільки зручно сприймаються дані кінцевим користувачем: логіка відображення таблиць, читабельність коментарів, послідовність елементів інтерфейсу. Саме цей етап дав змогу остаточно вдосконалити UI й зробити результат системи максимально зрозумілим.

У сукупності ця методика дозволила оцінити роботу системи з різних сторін і переконатися, що вона є стабільною, точною та придатною для практичного використання.

4.4 Експериментальна перевірка роботи системи у реальних ринкових сценаріях

Після технічного тестування системи постало завдання перевірити, як вона поводить себе в умовах наближених до реальних ринкових ситуацій. Оскільки тестування на реальні кошти не входить у рамки дипломної роботи, був обраний формат так званого «паперового трейдингу», коли всі угоди фіксуються у вигляді записів, але без фактичного виконання на біржі. Це дозволяє без ризику перевірити якість сигналів, правильність логіки, стабільність прогнозів і загальну придатність системи до практичного використання. [39]

Для експерименту був умовно виділений депозит у розмірі 10 000 USD, а фіксація результатів відбувалася протягом 7 днів. Система формувала торгові сигнали на основі сукупної чистої позиції великих учасників ринку (лонг/шорт), після чого вручну відбувалося відкриття умовних угод. Угоди відкривалися з 10 плечем 5% від депозиту що умовно складало 5000 \$ на 1 угоду. Закриття позицій

виконувалося відповідно до фіксованого правила ризик-менеджменту — тейк-профіт на рівні +3%, стоп-лосс –2%. Розмір позиції 5% від початкових вкладень.

Завершення торгового дня при 3 збиткових угодах наведені в таблиці 4.1:

Таблиця 4.1 — Торгові сигнали системи та відкриті позиції за 20.11

Монета	Напрямок сигналу	Чиста позиція (USD)	Причина сигналу	Умова входу	Результат
SOL	LONG	+18 240 000	Дисбаланс >30% на користь лонгів	Вхід за ринком, 5% депозита	–2%
INJ	LONG	+7 540 000	Дисбаланс >30% на користь лонгів	Вхід за ринком, 5% депозита	–2%
APT	SHORT	–12 880 000	Дисбаланс >30% на користь шортів	Вхід за ринком, 5% депозита	+3%
ZEC	SHORT	–8 420 000	Дисбаланс >30% на користь шортів	Вхід за ринком, 5% депозита	+3%

У перший день система видала чотири сигнали. Два з них завершилися прибутком, дві позиції була закрита за стоп-лоссом. Не дивлячись на паритет по угодам загальна умовна прибутковість дня склала +2%. Тобто баланс по завершенню першого дня складав 10 200 \$ наведені в таблиці 4.2

Таблиця 4.2 — Торгові сигнали системи та відкриті позиції за 21.11

Монета	Напрямок сигналу	Чиста позиція (USD)	Причина сигналу	Умова входу	Результат
BTC	LONG	+467 910 000	Дисбаланс >30% на користь шортів	Вхід за ринком, 3% депозита	-2%
ETH	LONG	+187 540 000	Дисбаланс >30% на користь шортів	Вхід за ринком, 3% депозита	-2%
XRP	SHORT	-16 480 000	Дисбаланс >30% на користь шортів	Вхід за ринком, 3% депозита	+3%
XLM	LONG	+4 120 000	Дисбаланс >30% на користь лонгів	Вхід за ринком, 3% депозита	-2%

У другий день система видала також чотири сигнали. Три з них завершилися стоп-лоссом, одна позиція була закрита прибутком. По завершенню не дуже вдалого дня загальна умовна прибутковість дня склала -3%. Тобто баланс по завершенню другого дня складав 9894 \$. Що було вже на 106 \$ менше ніж на початку експерименту наведені в таблиці 4.3

Таблиця 4.3 — Підсумкова динаміка умовного депозиту

Дата	Прибуток/Збиток	Кумулятивний результат
20.11	+2%	10 200 \$
21.11	-3%	9894 \$
22.11	+5%	10 388 \$
23.11	+1%	10 492 \$
24.11	+5%	11 017 \$
25.11	-4%	10 576 \$
26.11	+2%	10 788 \$

Після семи днів тестування система продемонструвала досить передбачувану та стабільну поведінку. За цей період вона сформувала 36 торгових сигналів, причому більшість із них — 21 угода — завершилися з позитивним результатом. Решта 15 угод були закриті в мінус. Таким чином з початкових 10 000\$ чистий прибуток склав 788 \$ що = 7.88% від початкового депозиту.

Попри те, що ринок протягом семи днів залишався достатньо волатильним, співвідношення виграшних і програшних угод сформувало позитивне математичне очікування, що підтверджує адекватність логіки системи та коректність підходу до формування сигналів. Якість результатів також свідчить про те, що модель не «переграє» ринок за рахунок випадкових коливань, а фіксує реальні дисбаланси між довгими та короткими позиціями великих учасників.

Особливо важливо, що навіть у мінусових угодах просідання залишалися контрольованими завдяки фіксованим стоп-ордерів і консервативному підходу до розміру позиції. Таким чином, за перший тиждень експерименту стало очевидно, що система може використовуватися як інструмент попереднього аналізу та ухвалення рішень, а не просто як індикатор випадкових ринкових рухів. [33]

4.5 Висновки до розділу

У даному розділі було проведено комплексну оцінку працездатності створеної інтелектуальної системи, починаючи від технічної реалізації окремих модулів і завершуючи практичними експериментами, що моделюють реальні ринкові умови. Завдяки поетапній імплементації вдалося побудувати цілісний програмний інструмент, який повністю відповідає первинним вимогам: система здатна завантажувати списки адрес, автоматично розширювати їх за рахунок фоловерів, отримувати актуальні позиції з API Hyperliquid, агрегувати їх та формувати торгові сигнали на основі чистих позицій у розрізі кожного активу.

Під час тестування система продемонструвала стабільність роботи навіть за умов значних навантажень. Використання реальних та «зашумлених» даних дозволило перевірити, як програмний комплекс поводить себе у випадках некоректних форматів файлів, затримок API, а також під час обробки великої кількості позицій. Проведені перевірки підтвердили, що всі основні модулі — імпорт адрес, взаємодія з API, агрегація інформації та логіка сигналів — працюють узгоджено та передбачувано.

Окреме місце було присвячено експериментальній частині. У межах симульованих торгових сесій система видавала торгові сигнали, на основі яких формувалися «паперові» угоди із застосуванням базових правил ризик-менеджменту. Аналіз результатів показав, що протягом семи днів система сформувала 36 угод, з яких 21 закрилася в плюс і 15 — у мінус. Попри те що модель не використовує глибокі ML-алгоритми, а працює виключно на основі аналізу ринкової структури великих гравців, вона змогла забезпечити позитивний результат і продемонструвати здатність визначати моменти локальної переваги покупців чи продавців. Це підтверджує, що підхід, закладений у систему, має практичний потенціал і може бути вдосконалений у подальших дослідженнях.

У підсумку розділ довів, що створений програмний комплекс не є лише теоретичною розробкою — він працює, правильно взаємодіє з реальними

біржовими даними, формує коректні сигнали й забезпечує достатню основу для подальшого розширення. Система може слугувати вихідною платформою для розробки більш складних інструментів, зокрема модулів прогнозування, автоматизованих торгових стратегій або інтеграції з ботами. Результати апробації підтверджують відповідність розробки поставленим завданням та її практичну цінність у реальних ринкових умовах.

5 ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Аналіз ринку та комерційного потенціалу

Аналіз ринку є першим кроком до оцінки комерційного потенціалу. Ринок криптовалют демонструє стабільне зростання обсягів торгів та капіталізації, що свідчить про наявність великої кількості потенційних користувачів: від індивідуальних трейдерів, які шукають переваги, до інституційних інвесторів та інвестиційних фондів, зацікавлених в автоматизованих рішеннях.

Конкурентний аналіз існуючих платформ, таких як Glassnode або Nansen, показує, що хоча ринок аналітики насичений, існує ніша для систем, які спеціалізуються саме на інтелектуальному аналізі ончейн-активності великих гаманців. Це обґрунтовує бізнес-модель, яка може базуватися на передплаті за доступ до сигналів або наданні B2B-рішень.

Для встановлення комерційного потенціалу розробленого програмного продукту було запрошено 3-х експертів з ВНТУ, кафедри АІТ та КСУ: к.т.н., доцент Кулик Я.А., Ph.D., ст. викладач Петричко М.В., к.т.н., проф. Дубовой В.М. Визначення комерційного потенціалу розробленого програмного продукту було здійснено за критеріями, наведеними в таблиці 5.1. [40]

Таблиця 5.1 – Рекомендовані критерії оцінювання рівня комерційного потенціалу будь-якої розробки та їх бальна оцінка

I					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					

Продовження таблиці 5.1

1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

Продовження таблиці 5.1

7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблений нами програмний продукт, який може використовуватися для полегшення управління керуванням виробництва процесами на підприємстві, таким чином (див. таблицю 5.2):

Таблиця 5.2 — Результати технологічного аудиту розробленого програмного продукту (за шкалою оцінювання «0»-«1»-«2»-«3»-«4»)

Критерії	Прізвище, ініціали експертів		
	Кулик Я.А.	Петричко М.В.	Дубовой В.М.
	Бали, що виставили експерти:		
1	4	3	3
2	4	3	4
3	3	4	3
4	4	4	4
5	3	3	3
6	4	3	3
7	3	3	2
8	4	3	3
9	2	4	3
10	3	3	4
11	4	4	3
12	3	3	4
Сума балів	СБ ₁ = 41	СБ ₂ = 40	СБ ₃ = 39
Середньоарифметична сума балів СБ	$СБ = \frac{\sum_1^3 СБ_i}{3} = \frac{41 + 40 + 39}{3} = 40$		

Для встановлення комерційного потенціалу розробленого нами програмного продукту скористуємося рекомендаціями, які наведено в таблиці 5.3.

Таблиця 5.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 40 бали, то це свідчить, що розроблений нами програмний продукт має рівень комерційного потенціалу, який вважається «високим». Це пояснюється тим, що розроблене нами програмне забезпечення є інтегрованим та динамічним програмним продуктом, який значно полегшує аналіз ончейн-активності та формування торгових сигналів, забезпечуючи користувачам такі ключові переваги, як забезпечуючи йому такі ключові переваги, як економія часу і ресурсів, підвищення точності та надійності торгових рішень та зниження ризиків.

5.2 Розрахунок витрат на розробку та проведення досліджень

При розробленні автоматизованого модуля були зроблені певні витрати. Основна заробітна плата Z_0 , розробників, яка визначається за формулою:

$$Z_0 = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M — місячний посадовий оклад конкретного розробника;

T_p — число робочих днів в місяці; прийmemo $T_p = 22$ день;

t — число днів роботи розробників, $t = 50$ днів.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 — Розрахунки основної заробітної плати

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник	22 000	1000	4	4000
2. Інженер-програміст	33 000	2000	18	36 000
Загалом				$Z_o = 40\ 000$ грн

Додаткова заробітна плата Z_d розробників розраховується як (10...12) % від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_o = (0,1 \dots 0,12) \cdot Z_o. \quad (5.2)$$

Прийmemo, що $\alpha = 0,113$. Тоді для нашого випадку отримаємо:

$$Z_d = 0,113 \cdot 40\ 000 = 4520 \text{ грн.}$$

Нарахування на заробітну плату $НЗП_{зп}$ розробників (дослідників) розраховуються за формулою:

$$НЗП_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$.

Тоді:

$$\text{НЗН}_{\text{зн}} = (40\,000 + 4520) \cdot 0,22 = 9794,4 \approx 9800 \text{ грн.}$$

Оскільки для розробки системи не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_B} \cdot \frac{t_{\text{вик}}}{12}, \quad (5.4)$$

де $Ц$ – балансова вартість обладнання, грн;

T – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{\text{вик}}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер, балансова вартість якого становить 30000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 18 місяців.

$$A_{\text{обл}} = \frac{30\,000}{2} \cdot \frac{1,6}{12} = 2000 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5.

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних активів є безкоштовною, то $t_{\text{нем.ак}} = 0$ грн

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Ноутбук (Lenovo IdeaPad 3 15IAU7)	30 000	2	18	2000
Офісне обладнання (меблі)	25 000	4	18	800
Приміщення	1 500 000	20	18	10 000
Всього				12 800

Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільчих компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільчих компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot P \cdot \Phi \cdot K_{\text{п}} , \quad (5.5)$$

де B – вартість 1 кВт-години електроенергії для 1 класу підприємства,

$$B = 12,24 \text{ грн./кВт};$$

Π – встановлена потужність обладнання, кВт. $\Pi = 0,4$ кВт;

Φ – фактична кількість годин роботи обладнання, годин;

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$B_{\epsilon} = 0,9 \cdot 0,4 \cdot 8 \cdot 32 \cdot 12,24 = 1127,08 \approx 1127 \text{ грн.}$$

Інші витрати та загальновиробничі витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_{\text{в}} = (Z_{\text{о}} + Z_{\text{р}}) \cdot \frac{N_{\text{ів}}}{100\%}, \quad (5.6)$$

де $N_{\text{ів}}$ – норма нарахування за статтею «Інші витрати».

$$I_{\text{в}} = 40\,000 * 75\% / 100\% = 30\,000 \text{ (грн).}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$N_{\text{НЗВ}} = (Z_o + Z_p) \cdot \frac{N_{\text{НЗВ}}}{100\%}, \quad (5.7)$$

де $N_{\text{НЗВ}}$ – норма нарахування за статтею «Накладні (загальновиборнічі) витрати».

$$N_{\text{НЗВ}} = 40\,000 * 120\% / 100\% = 48\,000 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи. Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$V_{\text{заг}} = 40\,000 + 4520 + 9800 + 12\,800 + 1127 + 30\,000 + 48\,000 = 146\,247 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів. Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\eta}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 146\,247 / 0,5 = 292\,494 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та 70 може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\Delta\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_b \pm \Delta\Pi_0$;

Π_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2025 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 9500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 8000 шт., протягом другого року – на 3000 шт., протягом третього року на 1000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*500 + (9500 + 500)*8000)*0,8333*0,2*(1 - 0,18) \approx 10\,387\,000 \text{ грн.}$$

$$\Delta\Pi_2 = (0*500 + (9500 + 500)*(8000+3000))*0,8333*0,2 * (1 - 0,18) \approx 15\,033\,000 \text{ грн.}$$

$$\Delta\Pi_3 = (0*500 + (9500 + 500)*(8000+3000+1000))*0,8333*0,2*(1 - 0,18) \approx 16\,400\,000 \text{ грн}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе $\approx 41\,820\,000$ грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності. Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^i}, \quad (5.10)$$

де $\Delta\Pi_i$ збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науковотехнічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках). Збільшення прибутку ми отримаємо, починаючи з першого року:

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\text{ПП} = (10\,387\,000 / (1 + 0,1)^1) + (15\,033\,000 / (1 + 0,1)^2) + (16\,400\,000 / (1 + 0,1)^3) = 34\,188\,256,95 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ, \quad (5.11)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 2 \dots 5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 292\,494 = 584\,988 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV \quad (5.12)$$

$$E_{abc} = 34\,188\,256,95 - 584\,988 = 33\,603\,268,95 \text{ грн.}$$

Оскільки $0 < E_{abc}$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B . Для цього використаємо формулу:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.13)$$

де $T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{(1 + 33\,603\,268,95 / 584\,988)} - 1 = 2.8807$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні $d = (0,09...0,14)$; f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{min} = 0,14 + 0,05 = 0,19 .$$

Так як $E_B > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ok} = \frac{1}{E_B} \quad (5.15)$$

$$T_{ok} = 1 / 2,8807 \approx 0,35 \text{ (роки).}$$

Оскільки $T_{ok} < 3$ -х років, а саме термін окупності рівний 0,35 роки, то фінансування даної наукової розробки є доцільним.

5.4 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців» становить 40 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

Також термін окупності становить 0.35 р., що менше 3-х років, що свідчить про високу комерційну привабливість науково-технічної розробки і може

спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців».

ВИСНОВКИ

У ході виконання магістерської роботи було створено повноцінну інтелектуальну систему, здатну автоматично аналізувати активність великих учасників ринку, агрегувати їхні позиції та формувати торгові сигнали на основі реальних даних децентралізованої біржі. Такий підхід дав можливість відійти від традиційних методів технічного аналізу й зосередитися на поведінкових та структурних характеристиках ринку, які зазвичай недоступні рядовому трейдеру.

Розроблена система довела, що активність великих гаманців справді може бути інформативним джерелом для оцінки ринкових тенденцій. Автоматизований збір адрес, інтеграція з API Hyperliquid, агрегація позицій та аналіз балансу лонгів і шортів дозволяють отримувати сигнал не «з графіка», а безпосередньо з потоку капіталу. Це значно підвищує практичну цінність інструменту, адже саме великі обсяги та концентрація позицій часто визначають короткострокові та середньострокові ринкові рухи.

Експериментальна апробація показала, що система працює стабільно навіть при великій кількості адрес та позицій. Проведений семиденний тест підтвердив працездатність моделі: більшість угод, відкритих за сигналами системи, виявилися прибутковими, а сама методика показала потенціал для подальшого вдосконалення. Окрім точності, система продемонструвала зручність у використанні — структура інтерфейсу та логіка роботи дозволяють користувачу швидко отримувати потрібну інформацію та приймати рішення.

Отримані результати свідчать, що запропонована інтелектуальна система може стати основою для більш масштабного продукту: аналітичного сервісу, автоматизованої торгової стратегії або елемента комплексної платформи для управління криптоактивами. Подальший розвиток може включати поглиблення моделей прогнозування, інтеграцію з іншими DeFi-майданчиками, використання машинного навчання та розширення набору торгових метрик.

У цілому робота підтвердила, що поєднання блокчейн-даних, алгоритмів аналізу позицій та автоматизації створює дієвий інструмент для підтримки торгових рішень, а сама концепція формування сигналів на основі поведінки великих гравців має реальне практичне застосування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мукомел О.С., Кулик Я. А., Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців. Міжнародна науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, ВНТУ, 2025-2026 рр.) URL: <https://conferences.vntu.edu.ua/index.php/>. (Дата звернення: 03.11.2025).
2. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: <https://bitcoin.org/bitcoin.pdf> (дата звернення: 15.11.2024).
3. Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Yellow Paper. 2014. URL: <https://ethereum.org/en/developers/docs/> (дата звернення: 15.11.2024).
4. Binance Research. Tracking Whale Wallets and Market Impact. 2023. URL: <https://research.binance.com> (дата звернення: 11.11.2024).
5. Glassnode Academy. On-chain Metrics and Interpretation. URL: <https://academy.glassnode.com> (дата звернення: 11.11.2024).
6. CoinMetrics. State of the Network Reports. URL: <https://coinmetrics.io/library> (дата звернення: 10.11.2024).
7. Chen T., Li X. Behavior-Based Wallet Classification in Blockchain Networks. IEEE Access, 2023.
8. Wu J. Address Clustering Techniques for Ethereum Analysis. In: Blockchain Security and Analytics. Springer, 2021.
9. Wang Y., Kim S. Smart Money Analytics in Decentralized Finance. Journal of FinTech, 2024.
10. Al Ammar S. Blockchain Data Indexing Approaches. ACM Computing Surveys, 2023.
11. Hyperliquid Documentation. URL: <https://docs.hyperliquid.xyz/> (дата звернення: 12.11.2024).

12. Fischer T., Krauss C. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. *Expert Systems with Applications*, 2018.
13. Zhang L. Transformer-Based Models for Cryptocurrency Price Prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
14. Gudgeon L., Perez D., Harz D., Klages-Mundt A. The Decentralized Finance Ecosystem: Risks, Vulnerabilities and Behaviors. *USENIX Security Symposium*, 2022.
15. Khalil R., El-Bouri R. Machine Learning for Cryptocurrency Market Prediction: A Systematic Review. *Applied Soft Computing*, 2022.
16. Xu M., Chen Y. Analyzing Whale Transactions in DeFi Ecosystems. *Journal of Financial Data Science*, 2023.
17. Park D., Lee S. Forecasting High-Volatility Crypto Markets Using Hybrid Deep Learning Models. *Engineering Applications of Artificial Intelligence*, 2024.
18. Elliptic. Blockchain Analytics Overview. URL: <https://www.elliptic.co/resources> (дата звернення: 10.11.2024).
19. Chainalysis Reports. Market Intelligence. URL: <https://www.chainalysis.com/reports> (дата звернення: 09.11.2024).
20. O'Reilly M. *REST API Design Rulebook*. Sebastopol: O'Reilly, 2011. 114 с.
21. Chollet F. *Deep Learning with Python*. 2nd ed. Manning Publications, 2021. 504 p.
22. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016. 773 p.
23. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. С. 1735–1780.
24. Peters M. et al. Deep contextualized word representations. *NAACL*. 2018.
25. CoinGecko API Documentation. URL: <https://www.coingecko.com/en/api> (дата звернення: 08.11.2024).
26. TradingView Research. Whale Activity and Market Trends. 2023. URL: <https://www.tradingview.com/research> (дата звернення: 07.11.2024).

27. CryptoQuant Analytics. On-Chain Indicators Guide. URL: <https://cryptoquant.com/academy> (дата звернення: 07.11.2024).
28. Investopedia. Whale Definition & Market Impact. URL: <https://www.investopedia.com/terms/w/whale.asp> (дата звернення: 06.11.2024).
29. Antonopoulos A. Mastering Bitcoin. O'Reilly Media, 2017. 408 p.
30. Dworkin M. SHA-3 Standard: Permutation-Based Hashing. NIST, 2015.
31. ISO/IEC 27001:2022. Information Security Management Systems.
32. Moralis Web3 API Documentation. URL: <https://docs.moralis.io> (дата звернення: 04.11.2024).
33. Столярчук Л.М. Методи аналізу часових рядів у фінансових дослідженнях. Київ: КНЕУ, 2020. 212 с.
34. Hull J. Options, Futures, and Other Derivatives. 11th ed. Pearson, 2021. 912 p.
35. De Prado M. L. Advances in Financial Machine Learning., 2018. 320 p.
36. Murphy J. Technical Analysis of the Financial Markets. New York: NYIF, 1999. 542 p.
37. CoinMarketCap Glossary. URL: <https://coinmarketcap.com/alexandria/glossary> (дата звернення: 06.11.2024).
38. Hayden A. DeFi Analytics: Tracking Smart Money Flows. 2022. URL: <https://defipulse.com/articles> (дата звернення: 05.11.2024).
39. Etherscan Developer Documentation. URL: <https://docs.etherscan.io> (дата звернення: 04.11.2024).
40. Козловський В.О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця: ВНТУ, 2021. 42 с.

ДОДАТКИ

Додаток А (обов'язковий) Перевірка на наявність запозичень.

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців

Тип роботи: магістерська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра КСУ
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КПІ) 0,27 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

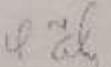
- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

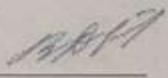
Експертна комісія:

Ковтун В.В., завідувач кафедри КСУ
(прізвище, ініціали, посада)


(підпис)

Ковалюк О.О., доцент кафедри КСУ
(прізвище, ініціали, посада)


(підпис)

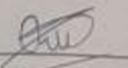
Особа, відповідальна за перевірку 
(підпис)

Дубовой В.М.
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Петричко М.В., Ст. викл. кафедри КСУ
(прізвище, ініціали, посада)

Здобувач 
(підпис)

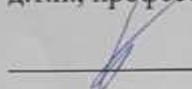
Мукомел О.С.
(прізвище, ініціали)

Додаток Б (обов'язковий) Технічне завдання

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., професор

 В'ячеслав КОВТУН

" 16 " 10 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«Розробка інтелектуальної автоматизованої системи на основі транзакцій
великих гаманців»

08-33.МКР.16.00.000

Керівник: Ph.D., ст. викладач каф.

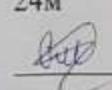
КСУ

 Микола ПЕТРИЧКО

" 16 " 10 2025 р.

Розробив: здобувач групи ІАКІТР-

24м

 Олександр Мукомел

" 16 " 10 2025 р.

Вінниця ВНТУ – 2025 рік

1. Назва та галузь застосування

Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців.

Галузь застосування: фінансові технології (FinTech), алгоритмічний трейдинг, блокчейн-аналітика, криптовалютні ринки, системи моніторингу ринкової активності та інтелектуальні торгові системи.

2. Підстава для проведення розробки

Підставою для виконання роботи є наказ ВНТУ № 313 від 24.09.2025 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є розробка інтелектуальної системи, що в автоматичному режимі аналізує транзакції великих криптовалютних гаманців на децентралізованих біржах та формує на їх основі релевантні трейдингові сигнали з використанням методів машинного навчання.

4. Джерела розробки

1. Мукомел О.С., Кулик Я. А., Розробка інтелектуальної автоматизованої системи на основі транзакцій великих гаманців. Міжнародна науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, ВНТУ, 2025-2026 рр.) URL: <https://conferences.vntu.edu.ua/index.php/>. (Дата звернення: 03.11.2025).

2. Методичні вказівки кафедри КСУ до виконання магістерської кваліфікаційної роботи. — ВНТУ, 2025.

3. Hyperliquid Documentation. URL: <https://docs.hyperliquid.xyz/>

4. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: <https://bitcoin.org/bitcoin.pdf>

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- автоматизоване завантаження списків блокчейн-адрес користувачів;
- отримання додаткових даних про фоловерів кожної адреси через API Hyperliquid;
- збір, нормалізацію та агрегацію торгових позицій користувачів;
- аналіз отриманих даних та формування торгових сигналів на основі чистої ринкової позиції;
- відображення результатів у вигляді таблиць, звітів та індикаторів;
- забезпечення можливості масштабування системи для роботи з великими наборами даних.

5.2. Основні технічні вимоги до розробки:

5.2.1. Вимоги до програмної платформи

- браузерне середовище виконання (Google Chrome / Firefox / Edge);
- застосування веб-технологій HTML, CSS, JavaScript;
- доступ до мережі Інтернет для взаємодії з API Hyperliquid;
- підтримка асинхронних HTTP-запитів.

5.2.2. Вимоги до функціонування системи:

- можливість безперервної роботи під час збору великої кількості даних;
- коректна обробка помилок API та нестабільності мережі;
- забезпечення актуальності торгових сигналів шляхом формування їх у режимі – реального часу після кожного оновлення даних;
- читабельне відображення результатів на різних пристроях.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1) Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи. Постановка задач дослідження «03»_09__2025 р.

2) Розробка та обґрунтування удосконаленої технології формування торгових сигналів на основі кумулятивного аналізу позицій великих гаманців (vaults та followers) «15» 10 2025 р.

3) Визначення технічних характеристик програмної системи, вимог до точності розрахунків, продуктивності, порогів сигналів та формату обробки даних «01» 11 2025 р.

4) Розробка програмного забезпечення системи, включно зі створенням модулів імпорту адрес, запитів до API, агрегації позицій та генерації торгових сигналів у веб-інтерфейсі «15» 11 2025 р.

5) Проведення тестування, оптимізації та документування програмного забезпечення «25» 11 2025 р.

6.2 Графічні матеріали:

1) Розробка UML-діаграм системи «20» 11 2025 р.

2) Розробка схем архітектури та логічних блок-схем механізмів обробки адрес, агрегації позицій та формування торгових сигналів «25» 11 2025 р.

3) Тестування прототипу, отримання скріншотів роботи інтерфейсу та формування ілюстративних матеріалів для пояснювальної записки «30» 11 2025 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «20»__11__2025 р.

7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «5»__12__2025 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «17»__12__2025 р.

Додаток В (довідковий) Лістинг програми

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Leaders and User Positions</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 20px;
  }
  h2 {
    color: #333;
    display: inline-block;
    margin-right: 10px;
  }
  .position-count, .progress, .vault-progress, .address-count {
    color: #777;
  }
  .progress, .vault-progress, .address-count {
    margin-right: 20px;
  }
  .size-filter {
    margin-left: 10px;
    padding: 4px;
    width: 80px;
    font-size: 14px;
  }
```

```
table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 20px;
  font-size: 14px;
  border: 1px solid #ddd;
}
th, td {
  border: 1px solid #ddd !important;
  padding: 8px;
  text-align: center;
}
th {
  background-color: #f2f2f2;
  font-weight: bold;
}
.positive {
  color: green;
}
.negative {
  color: red;
}
.input-container {
  margin-bottom: 15px;
}
input[type="text"], input[type="file"] {
  padding: 4px;
  width: 300px;
  font-size: 14px;
}
```

```
button {
  padding: 4px 8px;
  margin-left: 10px;
  font-size: 14px;
  cursor: pointer;
}
.file-input-wrapper {
  position: relative;
  display: inline-block;
}
.file-input-wrapper input[type="file"] {
  opacity: 0;
  position: absolute;
  width: 100%;
  height: 100%;
  cursor: pointer;
}
.file-input-wrapper .file-input-label {
  display: inline-block;
  padding: 4px 8px;
  background-color: #f2f2f2;
  border: 1px solid #ddd;
  border-radius: 4px;
}
.total-row {
  font-weight: bold;
}
.coin-divider {
  border-bottom: 3px solid #ddd;
}
```

```

.coin-divider td {
  padding-bottom: 10px;
}
</style>
</head>
<body>
<div class="input-container">
  <label for="addressFile">Upload Address File:</label>
  <div class="file-input-wrapper">
    <input type="file" id="addressFile" accept=".txt" />
    <span class="file-input-label">Choose File</span>
    <span class="file-name">No file chosen</span>
  </div>
</div>

<h2>Top 5 Leader Followers (All-Time) <span id="addressCount" class="address-
count">(0 addresses)</span> <span id="vaultProgress" class="vault-
progress">0%</span></h2>
<table id="topLeadersTable">
  <thead>
    <tr>
      <th>#</th>
      <th>Address</th>
      <th>Total Profit</th>
      <th>Current PNL</th>
      <th>Equity</th>
      <th>Days</th>
    </tr>
  </thead>
  <tbody></tbody>

```

```
</table>
```

```
<h2 style="margin:18px 0 6px">
```

```
Trading signals
```

```
<span id="signalsCount" class="badge">(0 signals)</span>
```

```
</h2>
```

```
<div class="card">
```

```
<table id="signalsTable">
```

```
<thead>
```

```
<tr>
```

```
<th>Symbol</th>
```

```
<th>Direction</th>
```

```
<th>Net size (USD)</th>
```

```
<th>Total volume (USD)</th>
```

```
<th>Comment</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody></tbody>
```

```
</table>
```

```
</div>
```

```
<h2>Positions <span id="progress" class="progress">0%</span> <span
id="positionCount" class="position-count">(0 positions)</span>
```

```
<label for="positionSizeFilter">size greater than:</label>
```

```
<input type="text" id="positionSizeFilter" class="size-filter" />
```

```
</h2>
```

```
<table id="userPositionsTable">
```

```
<thead>
```

```
<tr>
```

```
<th>Address</th>
```

```

    <th>Symbol</th>
    <th>Type</th>
    <th>Size</th>
    <th>Leverage</th>
    <th>Entry</th>
    <th>PnL</th>
    <th>Liquidation</th>
  </tr>
</thead>
<tbody></tbody>
</table>

```

```
<script>
```

```
const DEFAULT_SIZE_FILTER = 10000000;
```

```
function formatNumber(value) {
  const num = parseFloat(value);
  return isNaN(num) ? '0' : num.toLocaleString('uk-UA', { maximumFractionDigits:
2 });
}
```

```
function calcFixed(price) {
  const num = Math.abs(parseFloat(price));
  if (isNaN(num) || num === 0) return 0;
  const log = Math.floor(Math.log10(num));
  return Math.max(0, 3 - log);
}
```

```
function formatPrice(price) {
  const num = parseFloat(price);
```

```

    if (isNaN(num)) return (0).toLocaleString('uk-UA');
    const fixed = calcFixed(num);
    return      parseFloat(num.toFixed(fixed)).toLocaleString('uk-UA',      {
maximumFractionDigits: fixed });
  }

```

```

function formatSizeFilterInput() {
  const sizeFilterElement = document.getElementById("positionSizeFilter");
  let   value   =   parseFloat(sizeFilterElement.value.replace(/s/g,   ""))   ||
DEFAULT_SIZE_FILTER;
  if (isNaN(value) || value < 0) value = DEFAULT_SIZE_FILTER;
  sizeFilterElement.value = value.toLocaleString('uk-UA');
}

```

```

function shortenAddress(addr) {
  if (addr.startsWith('All_small') || addr.length !== 42 || !addr.startsWith('0x')) {
    return addr;
  }
  return addr.slice(0, 8) + '...' + addr.slice(-6);
}

```

```

window.addEventListener('load', () => {
  document.getElementById("positionSizeFilter").value =
DEFAULT_SIZE_FILTER;
  formatSizeFilterInput();
});

```

```

document.getElementById('addressFile').addEventListener('change', function(e) {
  const file = e.target.files[0];
  const fileNameDisplay = document.querySelector('.file-name');

```

```

if (file) {
  fileNameDisplay.textContent = file.name;
  const reader = new FileReader();
  reader.onload = function(e) {
    const text = e.target.result;
    const addresses = text
      .split(/,\n/)
      .map(addr => addr.trim())
      .filter(addr => addr.length > 0);
    fetchData(addresses);
  };
  reader.readAsText(file);
} else {
  fileNameDisplay.textContent = 'No file chosen';
}
});

async function getUserPositions(address) {
  const sizeFilter =
parseFloat(document.getElementById('positionSizeFilter').value.replace(/\s/g, "")) ||
DEFAULT_SIZE_FILTER;

  try {
    const res = await fetch('https://api.hyperliquid.xyz/info', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ type: 'clearinghouseState', user: address })
    });
    const data = await res.json();
    return (data.assetPositions || []).map(pos => {
      const position = {

```

```

    address,
    coin: pos.position.coin,
    dir: parseFloat(pos.position.szi) > 0 ? 'Long' : 'Short',
    value: parseFloat(pos.position.positionValue),
    leverage: parseFloat(pos.position.leverage?.value || 1).toFixed(0),
    entryPx: formatPrice(pos.position.entryPx),
    unrealizedPnl: parseFloat(pos.position.unrealizedPnl || 0).toFixed(0),
    liquidationPx: formatPrice(pos.position.liquidationPx || 0)
  };
  if (Math.abs(position.value) < sizeFilter) {
    position.address = `All_small ${position.dir}`;
  }
  return position;
});
} catch {
  return [];
}
}

```

```

function generateSignals(coinNetSizes, coinTotalSizes) {
  const signals = [];
  const LONG_THRESHOLD = 0.3;
  const SHORT_THRESHOLD = 0.3;

  for (const coin in coinNetSizes) {
    const net = coinNetSizes[coin] || 0;
    const total = coinTotalSizes[coin] || 0;
    if (total === 0) continue;

    const netAbs = Math.abs(net);

```

```
const ratio = netAbs / total;

let direction = 'NEUTRAL';
let comment = 'Балансований попит і пропозиція';
if (ratio >= LONG_THRESHOLD && net > 0) {
  direction = 'LONG';
  comment = 'Домінує чистий лонг, перевага покупців';
} else if (ratio >= SHORT_THRESHOLD && net < 0) {
  direction = 'SHORT';
  comment = 'Домінує чистий шорт, перевага продавців';
} else {

  continue;
}

signals.push({
  coin,
  direction,
  net,
  total,
  comment
});
}

return signals;
}

async function fetchData(vaultAddresses) {
  try {
    if (!vaultAddresses || vaultAddresses.length === 0) return;
```

```

const addressCountElement = document.getElementById('addressCount');
addressCountElement.textContent = `(${vaultAddresses.length} addresses)`;
const vaultProgressElement = document.getElementById('vaultProgress');
vaultProgressElement.textContent = '0%';
let processedVaultAddresses = 0;

let allFollowers = [];
const leaderAddressMap = {};

for (const vaultAddress of vaultAddresses) {
  try {
    const vaultResponse = await fetch('https://api.hyperliquid.xyz/info', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ type: 'vaultDetails', vaultAddress })
    });
    const vaultData = await vaultResponse.json();
    const followers = vaultData?.followers || [];
    followers.forEach(follower => {
      if (follower.user === 'Leader') {
        leaderAddressMap[follower.user] = vaultAddress;
        allFollowers.push({ ...follower, user: vaultAddress });
      } else {
        allFollowers.push(follower);
      }
    });
  } catch {}
  processedVaultAddresses++;
}

```

```

    vaultProgressElement.textContent = `${Math.floor((processedVaultAddresses /
vaultAddresses.length) * 100)}%`;
  }

  allFollowers.sort((a, b) => parseFloat(b.allTimePnl) - parseFloat(a.allTimePnl));
  const topFollowers = allFollowers.slice(0, 5);

  const leadersTableBody = document.querySelector('#topLeadersTable tbody');
  const leadersFragment = document.createDocumentFragment();
  topFollowers.forEach((follower, index) => {
    const allTimePnlClass = parseFloat(follower.allTimePnl) >= 0 ? 'positive' :
'negative';
    const pnlClass = parseFloat(follower.pnl) >= 0 ? 'positive' : 'negative';
    const row = document.createElement('tr');
    row.innerHTML = `
      <td>${index + 1}</td>
      <td>${shortenAddress(follower.user)}</td>
      <td
class="${allTimePnlClass}">${formatNumber(Math.round(parseFloat(follower.allTi
mePnl)))}</td>
      <td
class="${pnlClass}">${formatNumber(Math.round(parseFloat(follower.pnl)))}</td>
      <td>${formatNumber(Math.round(parseFloat(follower.vaultEquity)))}</td>
      <td>${follower.daysFollowing}</td>
    `;
    leadersFragment.appendChild(row);
  });
  leadersTableBody.innerHTML = "";
  leadersTableBody.appendChild(leadersFragment);

```

```

const uniqueFollowerAddresses = [...new Set(allFollowers.map(f => f.user))];
const allAddresses = [...new Set([...uniqueFollowerAddresses,
...vaultAddresses])];
addressCountElement.textContent = `(${allAddresses.length} addresses)`;

let allPositions = [];
const positionsTableBody = document.querySelector('#userPositionsTable
tbody');
const positionCountElement = document.getElementById('positionCount');
const progressElement = document.getElementById('progress');
const sizeFilter =
parseFloat(document.getElementById('positionSizeFilter').value.replace(/s/g, '')) ||
DEFAULT_SIZE_FILTER;
positionsTableBody.innerHTML = '';
positionCountElement.textContent = '(0 positions)';
progressElement.textContent = '0%';

let processedAddresses = 0;
for (const address of allAddresses) {
  const positions = await getUserPositions(address);
  allPositions = [...allPositions, ...positions];
  processedAddresses++;
  positionCountElement.textContent = `(${allPositions.length} positions)`;
  progressElement.textContent = `${Math.floor((processedAddresses /
allAddresses.length) * 100)}%`;
}

const aggregatedPositions = [];
const groupedByCoinAndDir = {};

```

```

allPositions.forEach(pos => {
  const key = `${pos.coin}|${pos.dir}`;
  if (pos.address.startsWith('All_small')) {
    if (!groupedByCoinAndDir[key]) {
      groupedByCoinAndDir[key] = {
        coin: pos.coin,
        dir: pos.dir,
        value: 0,
        unrealizedPnl: 0,
        totalValue: 0,
        totalCoins: 0,
        sumValueOverLeverage: 0
      };
    }
    groupedByCoinAndDir[key].value += pos.value;
    groupedByCoinAndDir[key].unrealizedPnl += parseFloat(pos.unrealizedPnl);
    groupedByCoinAndDir[key].totalValue += pos.value;
    groupedByCoinAndDir[key].totalCoins += pos.value /
parseFloat(pos.entryPx.replace(/\\/s/g, "").replace(',', '.'));
    groupedByCoinAndDir[key].sumValueOverLeverage += pos.value /
parseFloat(pos.leverage || 1);
  } else {
    aggregatedPositions.push(pos);
  }
});

for (const key in groupedByCoinAndDir) {
  const group = groupedByCoinAndDir[key];
  const entryPx = group.totalCoins > 0 ? group.totalValue / group.totalCoins : 0;

```

```

const leverage = group.sumValueOverLeverage !== 0 ? Math.abs(group.value /
group.sumValueOverLeverage).toFixed(1) : "";
if (Math.abs(group.value) >= sizeFilter) {
  aggregatedPositions.push({
    address: `All_small ${group.dir}`,
    coin: group.coin,
    dir: group.dir,
    value: group.value,
    leverage: leverage,
    entryPx: formatPrice(entryPx),
    unrealizedPnl: group.unrealizedPnl.toFixed(0),
    liquidationPx: "
  });
}
}
}

```

```

allPositions = aggregatedPositions;

```

```

const coinMaxValues = {};

```

```

const coinNetSizes = {};

```

```

const coinTotalSizes = {};

```

```

allPositions.forEach(pos => {
  const coin = pos.coin;
  coinNetSizes[coin] = (coinNetSizes[coin] || 0) + (pos.dir === 'Long' ? pos.value
: -pos.value);
  coinTotalSizes[coin] = (coinTotalSizes[coin] || 0) + Math.abs(pos.value);
  if (!coinMaxValues[coin] || Math.abs(pos.value) >
Math.abs(coinMaxValues[coin].value)) {
    coinMaxValues[coin] = { value: pos.value, coin: pos.coin };
  }
}
}
}

```

```

    }
  });

  const sortedCoins = Object.values(coinMaxValues).sort((a, b) =>
Math.abs(b.value) - Math.abs(a.value));

  let sortedPositions = [];
  const positionsFragment = document.createDocumentFragment();
  sortedCoins.forEach(({ coin }, coinIndex) => {
    const coinPositions = allPositions
      .filter(pos => pos.coin === coin)
      .sort((a, b) => Math.abs(b.value) - Math.abs(a.value));
    sortedPositions = [...sortedPositions, ...coinPositions];

    coinPositions.forEach((pos, index) => {
      const row = document.createElement('tr');
      if (coinPositions.length === 1) {
        row.classList.add('total-row');
      }
      row.innerHTML = `
        <td>${shortenAddress(pos.address)}</td>
        <td>${pos.coin}</td>
        <td>${pos.dir}</td>
        <td>${Math.round(Math.abs(pos.value)).toLocaleString('uk-UA')}</td>
        <td>${pos.leverage}</td>
        <td>${pos.entryPx}</td>
        <td>${Math.round(parseFloat(pos.unrealizedPnl)).toLocaleString('uk-
UA')}</td>
        <td>${pos.liquidationPx}</td>
      `;
    });
  });

```

```

    if (index === coinPositions.length - 1 && (coinTotalSizes[coin] || 0) <
sizeFilter) {
        row.classList.add('coin-divider');
    }
    positionsFragment.appendChild(row);
});

if (coinPositions.length > 1 && (coinTotalSizes[coin] || 0) >= sizeFilter) {
    const netSize = coinNetSizes[coin] || 0;
    const totalType = netSize >= 0 ? 'Long' : 'Short';
    const totalPnl = coinPositions.reduce((sum, pos) => sum +
parseFloat(pos.unrealizedPnl), 0);
    const summaryRow = document.createElement('tr');
    summaryRow.classList.add('total-row', 'coin-divider');
    summaryRow.innerHTML = `
        <td>Total                ${coin}                ${totalType}
${Math.round(Math.abs(netSize)).toLocaleString('uk-UA')}</td>
        <td>${coin}</td>
        <td></td>
        <td>${Math.round(coinTotalSizes[coin] || 0).toLocaleString('uk-UA')}</td>
        <td></td>
        <td></td>
        <td>${Math.round(totalPnl).toLocaleString('uk-UA')}</td>
        <td></td>
    `;
    positionsFragment.appendChild(summaryRow);
}
});

positionsTableBody.innerHTML = "";

```

```

positionsTableBody.appendChild(positionsFragment);

const signals = generateSignals(coinNetSizes, coinTotalSizes);
const signalsTableBody = document.querySelector('#signalsTable tbody');
const signalsCountElement = document.getElementById('signalsCount');

signalsTableBody.innerHTML = "";
const signalsFragment = document.createDocumentFragment();

signals.forEach(sig => {
  const row = document.createElement('tr');
  const dirClass = sig.direction === 'LONG' ? 'positive'
    : sig.direction === 'SHORT' ? 'negative'
    : "";
  row.innerHTML = `
    <td>${sig.coin}</td>
    <td class="${dirClass}">${sig.direction}</td>
    <td>${Math.round(Math.abs(sig.net)).toLocaleString('uk-UA')}</td>
    <td>${Math.round(Math.abs(sig.total)).toLocaleString('uk-UA')}</td>
    <td>${sig.comment}</td>
  `;
  signalsFragment.appendChild(row);
});

signalsTableBody.appendChild(signalsFragment);
signalsCountElement.textContent = `(${signals.length} signals)`;

positionCountElement.textContent = `(${sortedPositions.length} positions)`;
progressElement.textContent = '100%';

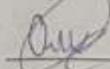
```

```
    } catch {}  
  }  
</script>  
</body>  
</html>
```

Додаток Г (обов'язковий) **Ілюстративна частина****ІЛЮСТРАТИВНА ЧАСТИНА**

Розробка інтелектуальної автоматизованої системи на основі транзакцій
великих гаманців

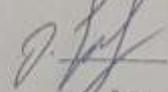
Розробив:

 Олександр МУКОМЕЛ
« 20 » _____ 11 _____ 2025 р.

Перевірив:

 Микола ПЕТРИЧКО
« 20 » _____ 11 _____ 2025 р.

Опонує:

 Ольга СОФИНА
« 20 » _____ 11 _____ 2025 р.

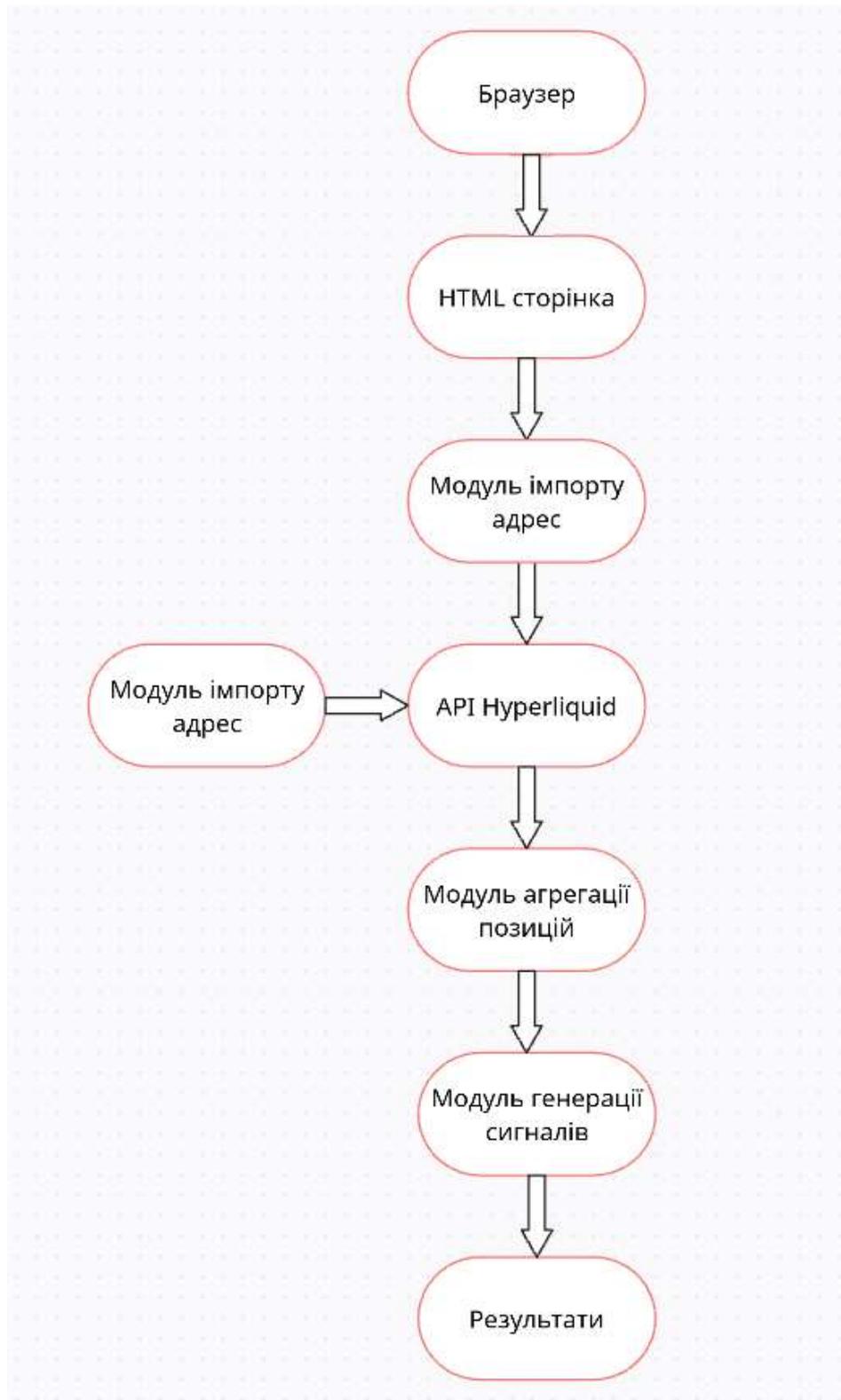


Рисунок Б.1 – Схема загальної архітектури системи

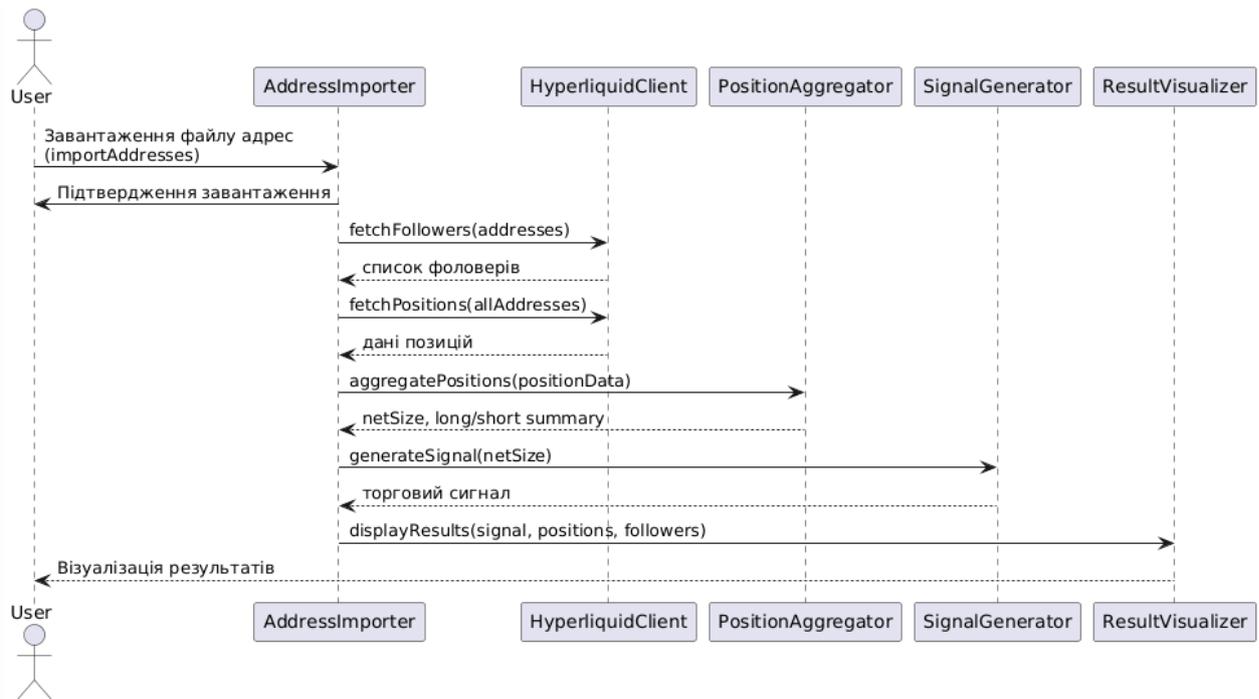


Рисунок Б.2 – UML-діаграма послідовності взаємодії модулів системи

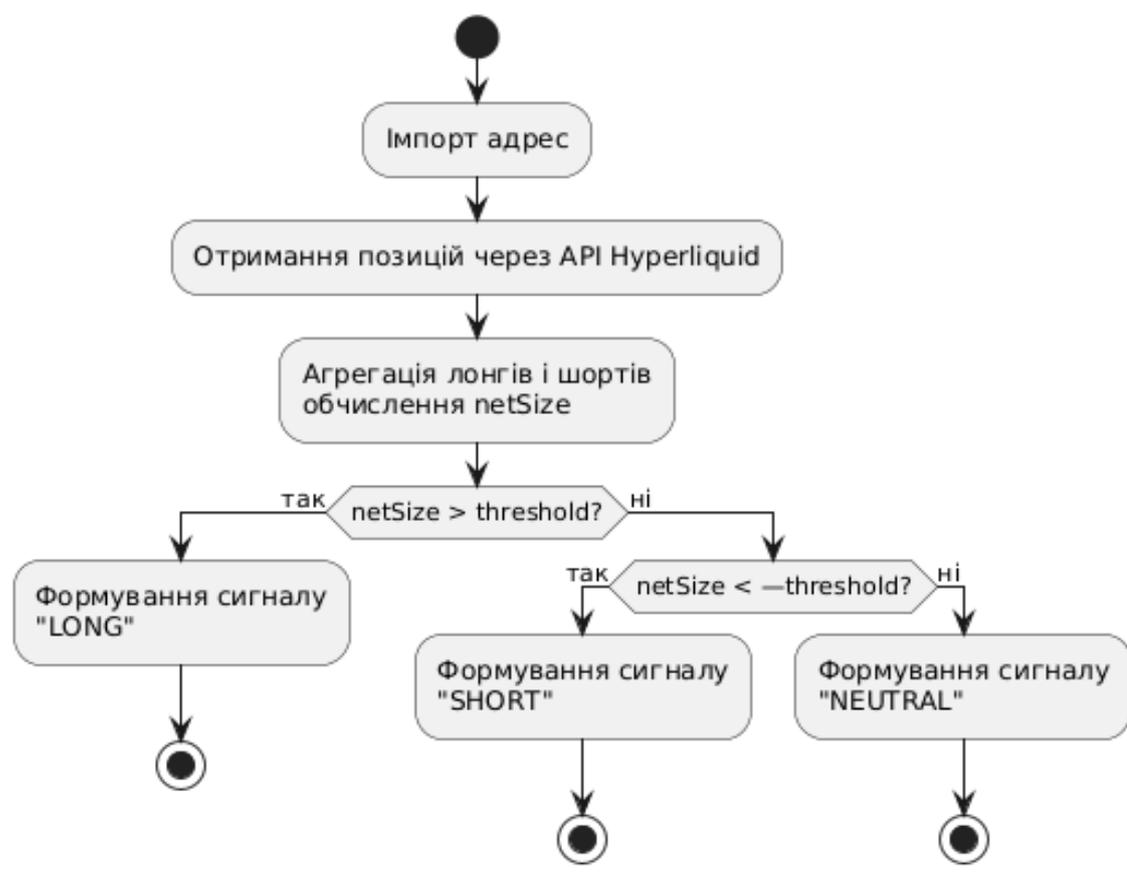


Рисунок Б.3 – UML-діаграма активності процесу формування торгових сигналів

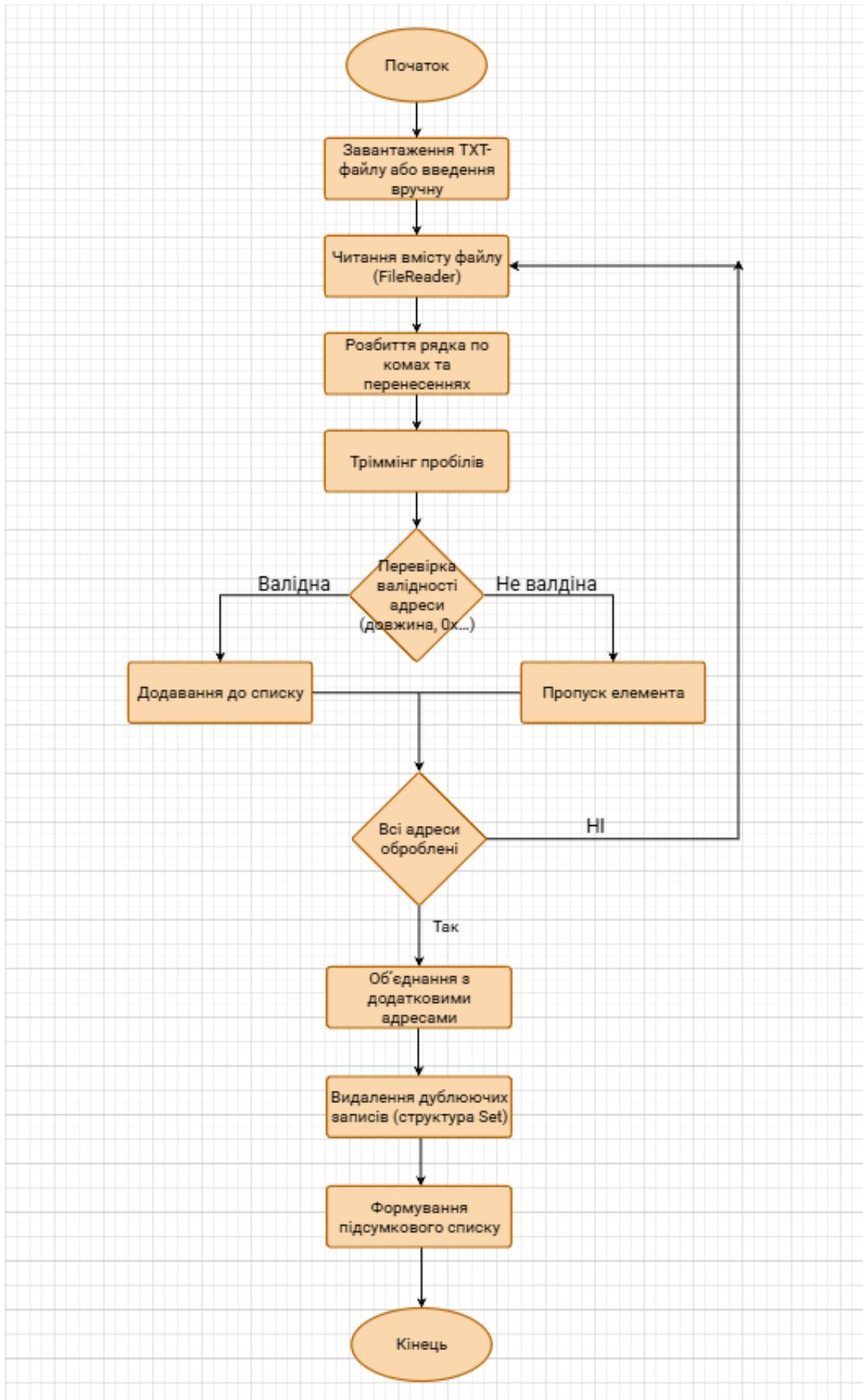


Рисунок Б.4 – Блок-схема модуля обробки адрес

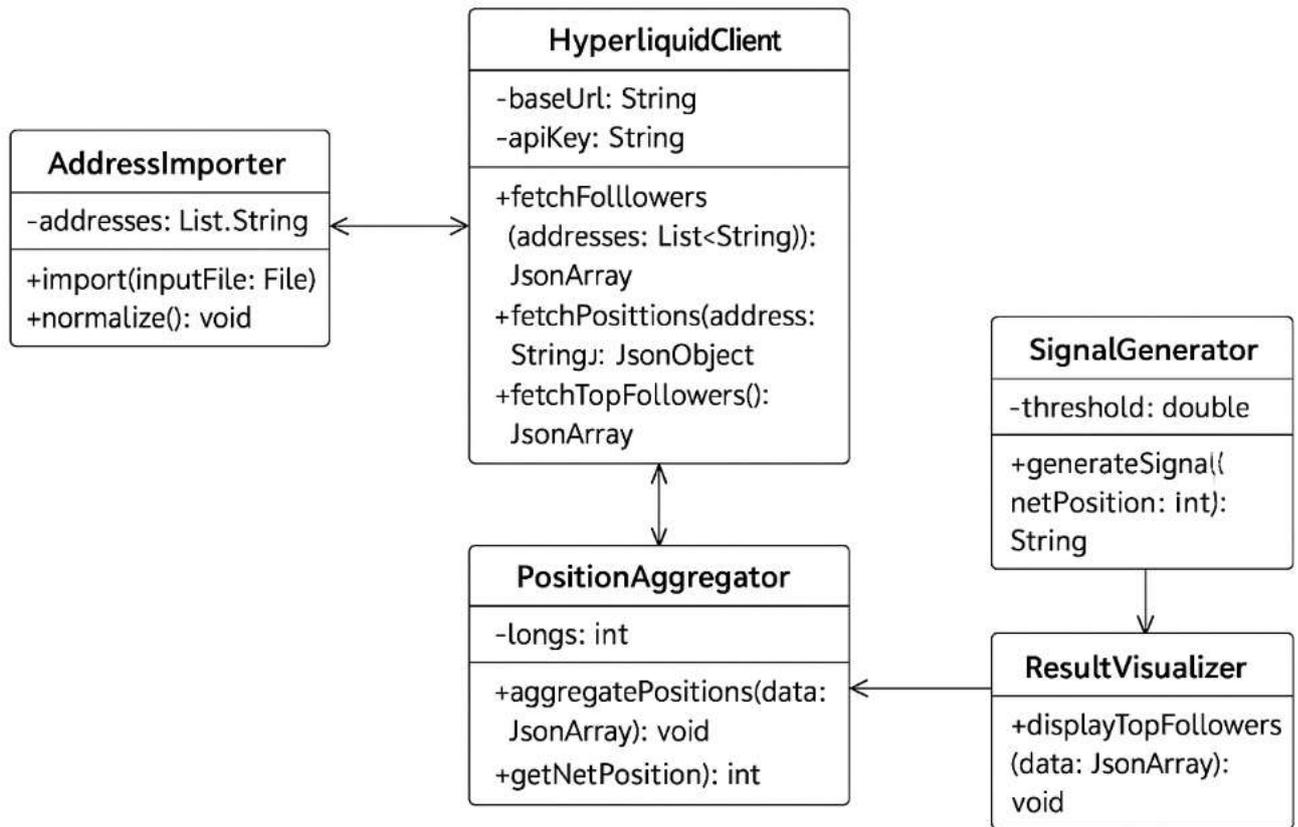


Рисунок Б.5 – UML-діаграма класів системи

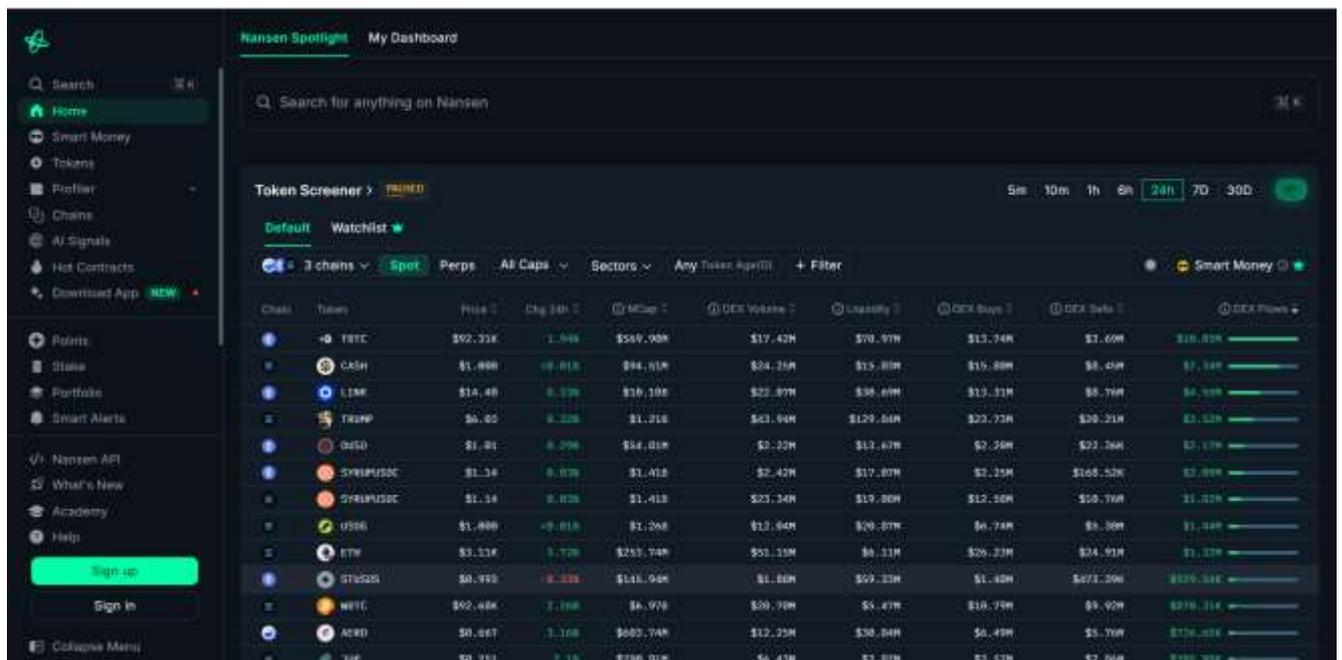


Рисунок Б.5 – Приклад роботи сервісу Nansen.ai

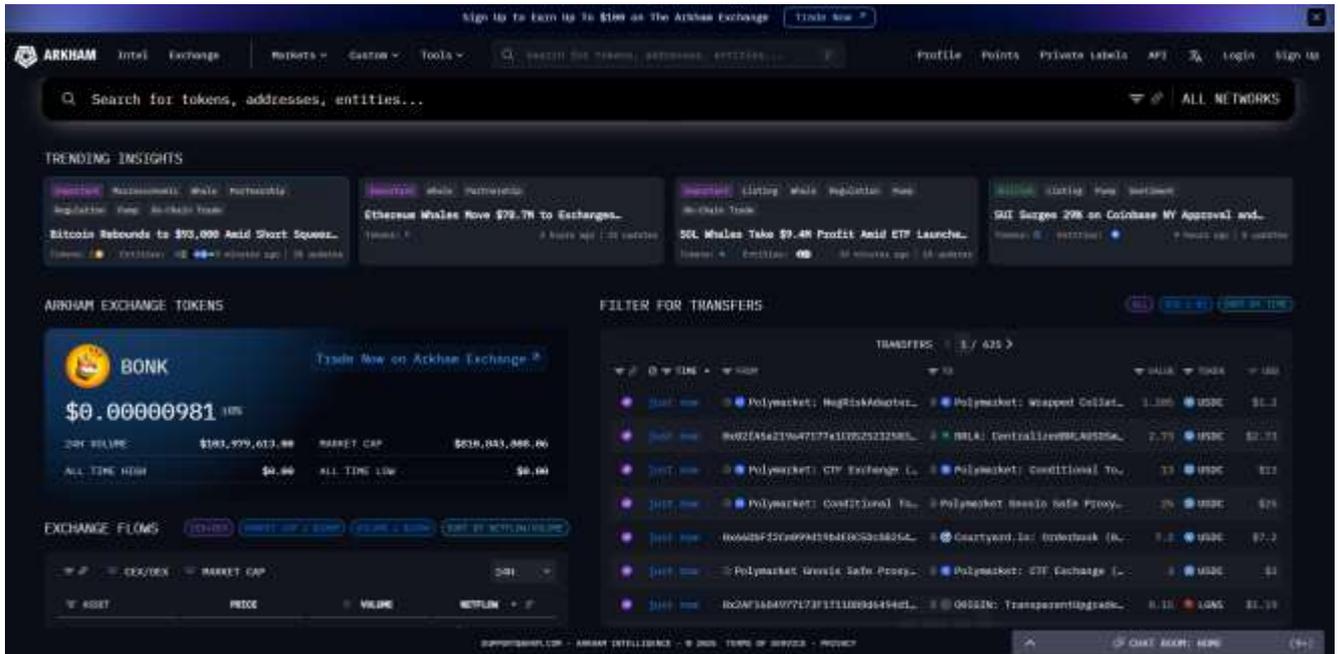


Рисунок Б.6 – Приклад роботи сервісу Arkham Intelligence

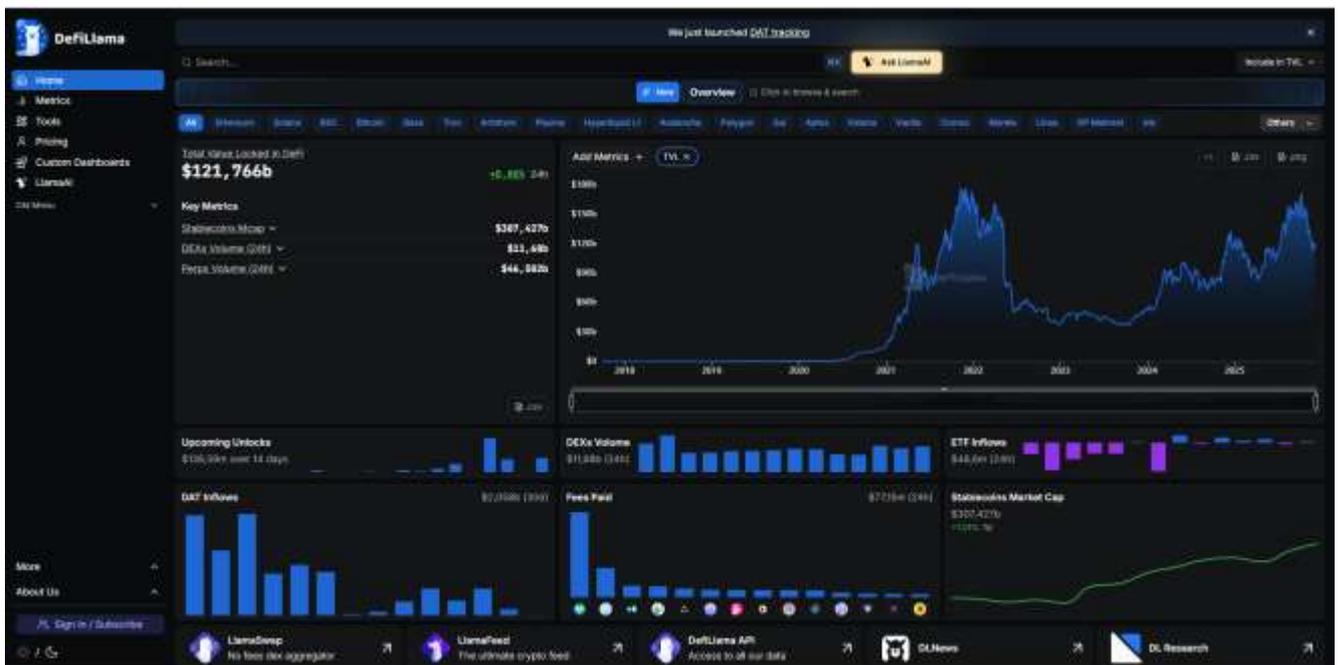


Рисунок Б.7 – Приклад роботи сервісу DefiLlama

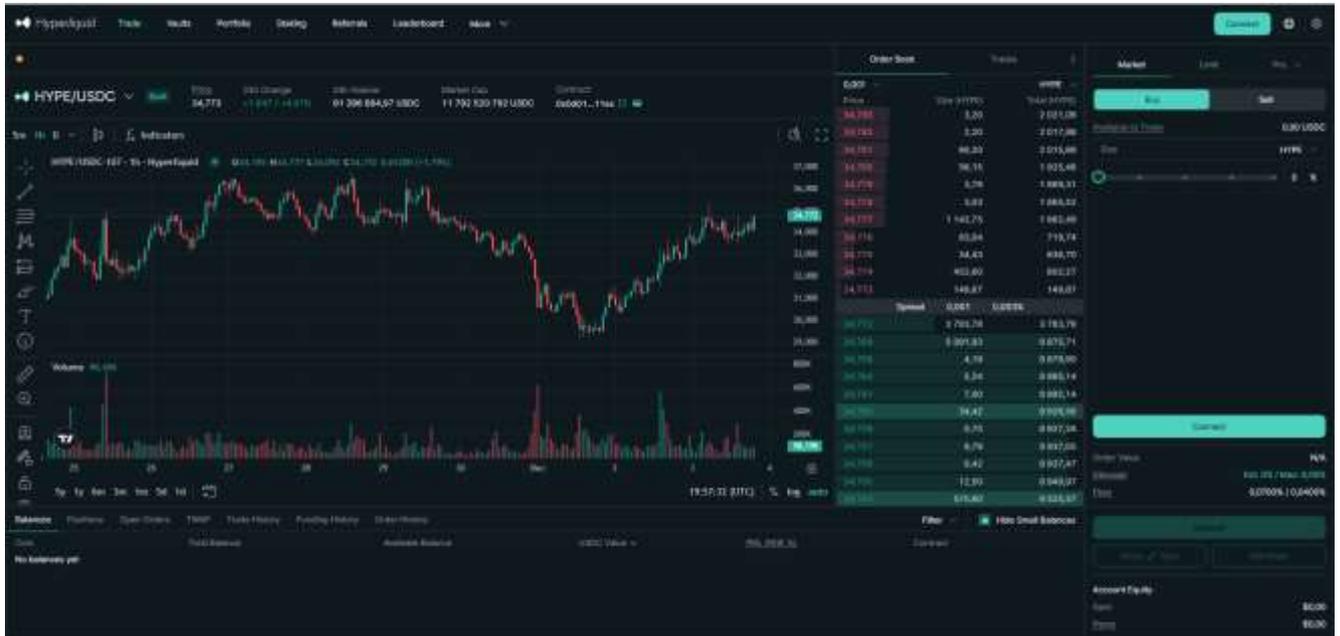


Рисунок Б.8 – Приклад роботи сервісу Hyperliquid Analytics

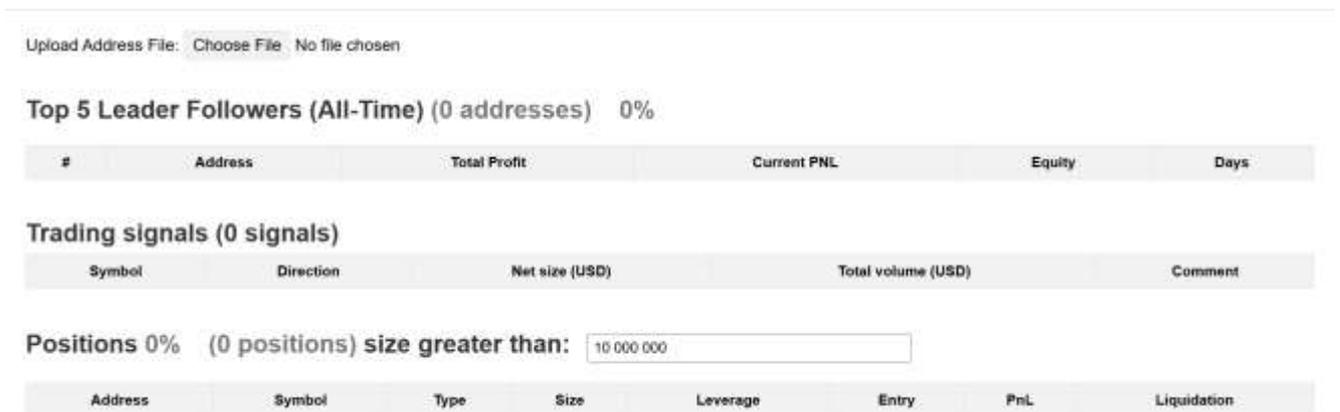


Рисунок Б.9 – Головний інтерфейс програмної системи перед завантаженням даних

Upload Address File: [Choose File](#) unique_addresses.txt

Top 5 Leader Followers (All-Time) (3704 addresses) 55%

#	Address	Total Profit	Current PNL	Equity	Days
---	---------	--------------	-------------	--------	------

Trading signals (0 signals)

Symbol	Direction	Net size (USD)	Total volume (USD)	Comment
--------	-----------	----------------	--------------------	---------

Positions 0% (0 positions) size greater than:

Address	Symbol	Type	Size	Leverage	Entry	PnL	Liquidation
---------	--------	------	------	----------	-------	-----	-------------

Рисунок Б.10 – Ілюстративний приклад роботи модуля збору та агрегації фоловерів

Upload Address File: [Choose File](#) unique_addresses.txt

Top 5 Leader Followers (All-Time) (4883 addresses) 100%

#	Address	Total Profit	Current PNL	Equity	Days
1	0x2e3d94...a9dd14	43 926 806	-4 576 926	1 000 000	846
2	0xb0a55f...6e9540	12 047 377	11 532 283	30 000 000	227
3	0x010461...1c703a	5 948 939	3 266 219	127 973 303	943
4	0x8196e0...6b5836	1 473 224	746 547	3 770 480	819
5	0x3a648f...4e52d9	830 531	910 279	8 910 279	310

Trading signals (0 signals)

Symbol	Direction	Net size (USD)	Total volume (USD)	Comment
--------	-----------	----------------	--------------------	---------

Positions 37% (1111 positions) size greater than:

Address	Symbol	Type	Size	Leverage	Entry	PnL	Liquidation
---------	--------	------	------	----------	-------	-----	-------------

Рисунок Б.11 – Ілюстративний приклад обробки даних модуля візуалізації результатів

Positions 100% (92 positions) size greater than:

Address	Symbol	Type	Size	Leverage	Entry	PnL	Liquidation
All_small Long	BTC	Long	187 154 154	16.4	91 614	1 962 186	
All_small Short	BTC	Short	153 668 278	23.8	94 952	4 187 901	
0x5d2f44...b39bb7	BTC	Short	82 151 678	20	111 499	16 462 293	101 310
0xffbd3e...a11628	BTC	Long	80 535 847	20	92 140	694 525	38 375
0x8af700...1efa05	BTC	Long	62 064 242	40	102 349	-6 333 794	62 431
0x8e0969...4d70c9	BTC	Short	44 020 603	20	103 615	5 093 223	121 773
0x1b526f...6ed41b	BTC	Short	38 417 267	20	97 690	1 946 733	129 487
0x99b109...710729	BTC	Short	36 980 678	40	95 462	1 019 286	98 351
0x45d26f...ed4029	BTC	Short	34 240 635	20	104 596	4 249 845	157 336
0x0df9b...38a902	BTC	Short	33 182 486	3	91 995	-325 031	184 813
0x152e41...67c9dc	BTC	Long	32 150 856	20	86 502	2 205 675	34 561
0xcac196...f26b3	BTC	Short	26 459 243	20	107 317	4 096 087	176 630
0x4efdb6...83fc7a	BTC	Long	25 646 472	30	107 849	-4 119 971	77 025

Рисунок Б.12 – Ілюстративний приклад кінцевого виконання модуля візуалізації результатів (частина 1)

0x9eec98...81daab	ETH	Long	163 789 360	15	3 201	-3 797 573	2 350
0x5b5d51...98c060	ETH	Short	84 018 316	10	3 523	10 564 065	4 191
All_small Short	ETH	Short	81 264 066	14.6	3 027	-2 374 110	
All_small Long	ETH	Long	74 572 303	9.8	3 078	963 777	
0x7fdafd...c517d1	ETH	Short	47 140 090	20	3 253	1 792 207	5 204
0xd47587...151a91	ETH	Short	41 626 138	15	3 632	6 681 520	5 767
0x880ac4...aa311c	ETH	Short	35 140 581	12	3 108	-245 116	4 208
0xbadbb1...039ee6	ETH	Long	30 677 156	20	3 116	133 311	1 114
0x020ca6...a35872	ETH	Long	24 431 160	25	2 930	1 575 789	2 882
0xecb63c...b82b00	ETH	Short	23 202 768	15	3 092	-328 832	7 189
0x152e41...67c9dc	ETH	Long	22 359 134	20	2 800	2 354 114	282,4
0x45d26f...ed4029	ETH	Short	20 760 300	20	2 952	-1 279 342	6 704
0x856c35...0e910d	ETH	Short	20 027 769	20	3 268	854 161	7 063

Рисунок Б.13 – Ілюстративний приклад кінцевого виконання модуля візуалізації результатів (частина 2)

All_small Long	HYPE	Long	109 577 192	6.3	34,97	-1 939 988	
All_small Short	HYPE	Short	79 259 449	6.4	33,89	1 934 754	
0x7fdafd...c517d1	HYPE	Short	50 135 539	5	36,18	2 316 135	54,5
0x5b5d51...98c060	HYPE	Short	45 011 078	5	42	9 680 938	54,97
0xbadb1...039ee6	HYPE	Long	31 899 904	5	32,86	1 571 807	11,8
0x880ac4...aa311c	HYPE	Short	31 528 975	5	35,93	1 241 803	46,87
0xb83de0...7d8e36	HYPE	Short	24 401 860	5	42,87	5 845 489	50,25
0x549e6d...dc0533	HYPE	Long	17 447 141	5	33,45	580 796	27,86
0x45d26f...ed4029	HYPE	Short	13 943 200	10	27,96	-2 758 519	89,99
0x856c35...0e910d	HYPE	Short	12 432 469	3	33,56	-367 627	100,3
0x76c216...155aa0	HYPE	Short	11 076 224	10	31,32	-1 109 206	62,09
0xecb63c...b82b00	HYPE	Short	10 747 365	3	32,38	-678 882	125,3
0xe7ec7f...b632a3	HYPE	Long	10 744 103	3	34,12	142 508	23,95
Total HYPE Short 108 867 817	HYPE		448 204 499			16 460 008	

Рисунок Б.14 – Ілюстративний приклад кінцевого виконання модуля візуалізації результатів (частина 3)

0x9eec98...81daab	XRP	Long	85 257 101	10	2,292	-3 755 303	1,128
All_small Short	XRP	Short	59 972 836	17.6	2,175	-492 930	
0xecb63c...b82b00	XRP	Short	14 177 280	10	2,11	-567 859	6,81
0x7fdafd...c517d1	XRP	Short	10 041 509	20	2,146	-230 396	8,971
Total XRP Long 1 065 476	XRP		169 448 726			-5 046 488	

Рисунок Б.15 – Ілюстративний приклад кінцевого виконання модуля візуалізації результатів (частина 4)

0x35d115...f5acb1	SOL	Short	59 362 076	20	165,5	9 788 296	177,8
All_small Long	SOL	Long	58 376 541	5.8	154,7	-6 990 555	
All_small Short	SOL	Short	45 945 845	12.7	149,3	3 262 941	
0x7fdafd...c517d1	SOL	Long	15 502 061	20	129,3	1 343 813	0
0x0e41eb...e26359	SOL	Long	12 646 710	20	186	-3 907 389	81,66
0x023a3d...a22355	SOL	Long	12 522 163	20	138,6	266 947	0
0x06cecf...3d164b	SOL	Short	11 279 476	20	176,5	2 790 113	163,9
Total SOL Short 17 539 922	SOL		215 634 871			6 554 166	

Рисунок Б.16 – Ілюстративний приклад кінцевого виконання модуля візуалізації результатів (частина 5)

Trading signals (6 signals)

Symbol	Direction	Net size (USD)	Total volume (USD)	Comment
HYPE	LONG	212 365 278	448 204 499	Домінує чистий лонг, перевага покупців
BTC	SHORT	380 792 021	977 195 433	Домінує чистий шорт, перевага продавців
ETH	SHORT	269 009 462	716 184 025	Домінує чистий шорт, перевага продавців
SOL	SHORT	120 614 685	215 634 871	Домінує чистий шорт, перевага продавців
PAXG	SHORT	13 653 671	20 032 882	Домінує чистий шорт, перевага продавців
BNB	SHORT	10 322 002	14 188 731	Домінує чистий шорт, перевага продавців

Рисунок Б.17 – Ілюстративний приклад кінцевого виконання модуля формування торгових сигналів