

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА
МАТЕМАТИКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ПРОГРАМНОЇ
ІНЖЕНЕРІЇ

**Аналіз доходності фінансових інструментів з використанням
машинного навчання з підкріпленням для адаптивного управління
диверсифікованим інвестиційним портфелем**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти “магістр”

Виконав: здобувач 2 курсу 12-261М групи
Спеціальності «126 Інформаційні системи та
технології»

Освітньо-професійної програми
«Інформаційні системи та технології»

Фоменка Леоніда Сергійовича

Керівник: доктор економічних наук,
професор Кобець В. М.

Рецензент: Іванов О. Ю.

Старший розробник ІТ-компанії TurnKey
Labs

Івано-Франківськ – 2025

ЗМІСТ

ВСТУП	
РОЗДІЛ 1. Теоретичні основи формування інвестиційного портфелю	
1.1. Класичні підходи до оптимізації інвестиційного портфелю.....	
1.1.1. Теорія Марковіца та оптимізація середньої дисперсії.....	
1.1.2. Традиційні метрики оцінки ефективності портфелю	
1.1.3. Обмеження класичних підходів.....	
1.2. Сучасні методи оцінки ризиків інвестиційного портфелю.....	
1.2.1. Аналіз різних мір ризику (коефіцієнти Шарпа, Сортіно, Стерлінга)	
1.2.2. Концепція максимальної просадки (MDD) та її значення	
1.2.3. Коефіцієнт Кальмара як інтегрована метрика ризик-прибутковості	
1.2.4. Висновки до підрозділу	
РОЗДІЛ 2. ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ В УПРАВЛІННІ ІНВЕСТИЦІЙНИМ ПОРТФЕЛЕМ	
2.1. Рекурентне навчання з підкріпленням (RRL)	
2.2. Динамічне управління портфелем.....	
РОЗДІЛ 3. ВЕБ-СКРАПІНГ ЯК ІНСТРУМЕНТ ЗБОРУ ФІНАНСОВИХ ДАНИХ	
3.1. Методи збору фінансової інформації	
3.2. Обробка та аналіз зібраних даних	
3.3. Візуалізація даних та виявлення трендів.....	
РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ОПТИМІЗАЦІЇ ПОРТФЕЛЮ	
4.1. Функціональні можливості програмного забезпечення	
4.1.1 Джерело та структура даних	
4.1.2. Формування фінансових ознак.....	
4.1.3. Поділ вибірки на навчальну та тестову	
4.1.4. Використані технології та середовище виконання	
4.1.5. Висновок до підрозділу	
4.2. Впровадження алгоритмів аналізу та оптимізації	
4.3. Впровадження алгоритму машинного навчання з підкріпленням.....	
4.3.1. Підготовка даних алгоритму машинного навчання з підкріпленням.....	
4.3.2. Середовище PortfolioEnv	
4.3.3. Алгоритм навчання агента	
4.3.4. Rolling backtest та оцінка ефективності	
4.3.5. Візуалізація результатів	

4.3.6. Висновки навчання агента.....	
4.4. Оцінка ефективності моделі.....	
4.4.1. Порівняння результатів класичної та RL-моделі.....	
4.4.2. Аналіз результатів навчання RL-агента	
Висновки	
Джерела	

ВСТУП

Фінансовий ринок є складною та динамічною системою, що перебуває під впливом численних внутрішніх і зовнішніх факторів. Зміни у глобальній економіці, політичні процеси, зростання ролі інформаційних технологій та швидке поширення цифрових інструментів визначають нові умови для інвесторів. У таких обставинах прийняття ефективних рішень щодо управління активами потребує використання сучасних методів аналізу та моделювання.

Традиційні підходи до формування інвестиційного портфелю базуються на теорії Марковіца та подальших розробках у сфері оптимізації співвідношення прибутковості та ризику. Вони дали можливість закласти фундамент для створення портфельних стратегій і тривалий час залишалися основою для практичного інвестування. Проте зі зростанням обсягів фінансових даних та ускладненням взаємозв'язків між активами виявилися обмеження класичних моделей. Вони не завжди враховують нелінійність процесів, особливості поведінки ринку у кризові періоди та вплив додаткових факторів, які не піддаються простій статистичній оцінці.

Ці виклики зумовили активне впровадження у фінансову сферу методів машинного навчання. Особливе місце серед них посідає навчання з підкріпленням, що дозволяє будувати адаптивні моделі управління портфелем. Такі системи здатні навчатися на основі попереднього досвіду, враховувати зміни середовища та вдосконалювати прийняття рішень у процесі роботи. Це відкриває перспективи для підвищення ефективності управління активами та кращого врахування ризиків.

Важливою складовою сучасних досліджень є питання збору та обробки даних. Оперативність та якість інформації визначають успішність будь-якої аналітичної моделі. Використання технологій веб-скрапінгу робить можливим систематичний збір фінансових показників із відкритих джерел, що дає змогу створювати бази даних для подальшого аналізу. Поєднання таких технологій із алгоритмами машинного навчання забезпечує більш комплексний підхід до прогнозування та оптимізації портфелю.

Об'єктом дослідження у цій роботі є методи машинного навчання, які застосовуються у фінансовій сфері. Предметом

дослідження виступає процес формування інвестиційного портфелю із використанням алгоритмів навчання з підкріпленням.

Метою роботи є розробка підходу до формування інвестиційних портфелів для інвесторів із різними цілями на основі методів навчання з підкріпленням.

Для досягнення цієї мети передбачено **виконання таких завдань**:

- здійснити аналіз теоретичних основ формування портфелю та сучасних підходів до оцінки його ефективності
- дослідити можливості застосування алгоритмів машинного навчання для управління портфелем
- розглянути методи збору та підготовки фінансових даних для подальшого аналізу
- реалізувати програмне забезпечення, що поєднує інструменти збору даних та алгоритми машинного навчання з підкріпленням
- провести тестування розробленого рішення та порівняти його результати з класичними методами оптимізації

Наукова новизна роботи полягає у застосуванні навчання з підкріпленням для управління портфелем, що дозволяє враховувати динамічні зміни ринку та поєднувати сучасні алгоритми з класичними методами оцінки ризику.

Практичне значення полягає у можливості використання розробленого підходу для побудови інвестиційних стратегій, здатних враховувати індивідуальні потреби інвесторів. Створене програмне забезпечення може бути застосоване як інструмент для підтримки рішень у сфері управління фінансовими активами.

Таким чином, робота спрямована на вирішення важливої наукової та практичної задачі - підвищення ефективності формування інвестиційного портфелю за допомогою сучасних технологій машинного навчання та інструментів збору фінансових даних.

РОЗДІЛ 1. Теоретичні основи формування інвестиційного портфелю

Формування інвестиційного портфеля є однією з ключових задач фінансового менеджменту. У науковій літературі накопичено значний досвід дослідження класичних підходів до оптимізації портфеля, які становлять основу сучасних методів управління фінансовими активами. Розуміння цих теоретичних основ є необхідним для оцінки ефективності новітніх інструментів та алгоритмів, зокрема методів машинного навчання, що набули поширення у фінансовій практиці останніх років.

1.1. Класичні підходи до оптимізації інвестиційного портфелю

1.1.1. Теорія Марковіца та оптимізація середньої дисперсії

Теорія портфельного інвестування, запропонована Гаррі Марковіцем у 1952 році, стала основою сучасного підходу до управління активами. Вона ґрунтується на моделі середньої дисперсії, яка передбачає формування оптимального портфеля шляхом балансування між очікуваною прибутковістю та ризиком. У даній концепції прибутковість визначається як середнє значення доходностей активів, а ризик оцінюється через їх дисперсію або стандартне відхилення. Важливим елементом цієї моделі є врахування коваріацій між активами, що дозволяє знижувати загальний ризик портфеля завдяки диверсифікації [1].

Оптимізація за Марковіцем передбачає побудову ефективної множини портфелів. Ця множина включає портфелі, що забезпечують мінімальний ризик за заданого рівня очікуваної прибутковості або максимальну прибутковість за певного рівня ризику. Таким чином, інвестор отримує можливість обирати портфель, який відповідає його індивідуальним уподобанням щодо співвідношення ризику і доходу. Теорія Марковіца започаткувала системний підхід до інвестування та дала підґрунтя для подальшого розвитку моделей, таких як CAPM (модель оцінки капітальних активів) [1].

1.1.2. Традиційні метрики оцінки ефективності портфелю

Після поширення портфельної теорії виникла потреба у створенні метрик, які дозволяють оцінювати ефективність сформованих портфелів. Одним із найпоширеніших показників є коефіцієнт Шарпа. Він вимірює відношення надлишкової прибутковості портфеля (понад безризикову

ставку) до його стандартного відхилення. Високе значення коефіцієнта Шарпа свідчить про ефективне співвідношення ризику та доходу [2].

Іншим важливим показником є коефіцієнт Трейнора, що оцінює прибутковість портфеля відносно його систематичного ризику, вимірюваного за допомогою бета-коефіцієнта. На відміну від коефіцієнта Шарпа, який враховує загальний ризик, показник Трейнора дає змогу оцінювати дохідність у контексті лише ринкового ризику [3].

Коефіцієнт Дженсена (або альфа Дженсена) також використовується для аналізу ефективності управління портфелем. Він показує, наскільки фактична прибутковість перевищує очікувану відповідно до моделі САРМ. Додатне значення альфи свідчить про те, що портфель перевершує ринок, тоді як від'ємне - про його відставання [1].

Застосування цих метрик дозволяє інвесторам проводити порівняльний аналіз різних стратегій та приймати рішення на основі кількісних показників, що значно підвищує об'єктивність процесу управління портфелем.

1.1.3. Обмеження класичних підходів

Незважаючи на фундаментальне значення теорії Марковіца та похідних від неї моделей, вони мають низку обмежень, що знижують їх практичну ефективність.

По-перше, ключове припущення про нормальний розподіл доходностей активів рідко відповідає реальності. На фінансових ринках часто спостерігаються асиметричні розподіли, наявність "важких хвостів" та ефект кластеризації волатильності, що не враховуються у класичних моделях [4].

По-друге, застосування середньо-дисперсійного підходу передбачає використання історичних даних для оцінки очікуваної прибутковості та ризику. Однак минулі показники не завжди є надійним індикатором майбутньої динаміки, особливо в умовах кризових явищ чи структурних змін на ринку [4].

По-третє, у класичних моделях недостатньо враховуються додаткові фактори, що мають суттєве значення для інвестора. Це стосується ліквідності активів, обмежень на обсяг угод, податкових

аспектів та транзакційних витрат. Такі елементи здатні істотно вплинути на реальну ефективність портфеля.

Крім того, традиційні підходи здебільшого ігнорують поведінкові фактори, пов'язані з психологією інвестора та впливом ринкових настроїв. Унаслідок цього класичні моделі не завжди відображають повну складність фінансових ринків і можуть бути недостатньо ефективними у високоволатильних або нестабільних умовах.

Таким чином, класичні підходи до оптимізації портфеля, незважаючи на свою значущість, потребують удосконалення. Виявлені обмеження стимулювали розвиток нових методів, які більш повно враховують сучасні тенденції ринку та можливості обчислювальних технологій [4].

1.2. Сучасні методи оцінки ризиків інвестиційного портфелю

Оцінка ризику є одним із ключових завдань при управлінні інвестиційними портфелями. Якщо класичні моделі здебільшого ґрунтувалися на аналізі дисперсії доходностей, то сучасні підходи враховують ширший спектр факторів і дозволяють більш точно відображати реальну ризиковість активів. Це сприяє формуванню більш надійних стратегій управління портфелем, здатних забезпечити стійкість у мінливих ринкових умовах [4].

1.2.1. Аналіз різних мір ризику (коефіцієнти Шарпа, Сортіно, Стерлінга)

Коефіцієнт Шарпа тривалий час залишався одним із найпоширеніших показників ефективності інвестицій. Він визначає співвідношення між надлишковою прибутковістю портфеля та його стандартним відхиленням. Хоча цей показник широко використовується на практиці, він має суттєвий недолік: стандартне відхилення враховує як позитивні, так і негативні відхилення від середнього значення, тоді як інвестор здебільшого зацікавлений у мінімізації саме негативних результатів [5].

Для усунення цієї проблеми було запропоновано коефіцієнт Сортіно. Він базується на концепції "downside risk", тобто оцінює лише негативні коливання доходності відносно певного порогового рівня. Завдяки цьому коефіцієнт Сортіно є більш інформативним для

інвесторів, оскільки акцентує увагу на тих ситуаціях, коли прибутковість портфеля виявляється нижчою від очікуваної або бажаної [6].

Ще однією сучасною мірою ефективності є коефіцієнт Стерлінга. Він визначається як відношення середньої річної прибутковості до середнього значення максимальної просадки. Цей показник дозволяє оцінити, наскільки ефективно портфель компенсує втрати, які виникають у процесі інвестування. На відміну від коефіцієнта Шарпа, який абстрагується від часових аспектів, коефіцієнт Стерлінга враховує характер динаміки втрат і тому вважається більш наближеним до реальних інвестиційних умов [7].

1.2.2. Концепція максимальної просадки (MDD) та її значення

Максимальна просадка (Maximum Drawdown, MDD) є одним із найбільш інтуїтивно зрозумілих та практично значущих показників ризику. Вона визначається як найбільше падіння вартості портфеля від локального максимуму до наступного мінімуму за певний період часу [8].

Значення цього показника полягає в тому, що він безпосередньо відображає можливі втрати інвестора у найгіршому сценарії. Інвестори часто орієнтуються саме на MDD, оскільки він показує рівень зниження капіталу, який може бути психологічно або фінансово неприйнятним. Наприклад, портфель із високою прибутковістю, але значною просадкою може виявитися менш привабливим, ніж портфель із помірною дохідністю та стабільною динамікою [9].

Максимальна просадка також активно використовується в алгоритмічних стратегіях і під час тестування торгових систем. Вона дозволяє виявити слабкі місця алгоритму та визначити стійкість стратегії до несприятливих ринкових умов [10].

1.2.3. Коефіцієнт Кальмара як інтегрована метрика ризик-прибутковості

Коефіцієнт Кальмара поєднує у собі підхід до оцінки прибутковості та максимальних втрат. Він розраховується як відношення середньої річної прибутковості портфеля до його максимальної просадки. Таким чином, даний показник інтегрує у собі як

дохідність, так і ризик, виражений у вигляді найбільших можливих втрат [11].

Високе значення коефіцієнта Кальмара свідчить про те, що портфель здатен генерувати стабільну прибутковість без надмірних падінь капіталу. Ця метрика є особливо цінною для інвесторів, які прагнуть не лише максимізувати дохідність, а й мінімізувати ймовірність значних втрат. У практиці управління активами коефіцієнт Кальмара часто використовується для порівняння різних стратегій або фондів, оскільки він відображає збалансованість між доходом та ризиком [12].

1.2.4. Висновки до підрозділу

Сучасні методи оцінки ризику розширюють можливості інвесторів у порівнянні з класичними підходами, дозволяючи більш точно оцінювати стійкість портфеля в різних ринкових умовах. Використання таких показників, як коефіцієнти Сортіно, Стерлінга та Кальмара, а також аналіз максимальної просадки, забезпечує комплексне бачення ризиків і дозволяє формувати інвестиційні стратегії, що відповідають індивідуальним цілям і обмеженням інвестора.

РОЗДІЛ 2. ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ В УПРАВЛІННІ ІНВЕСТИЦІЙНИМ ПОРТФЕЛЕМ

У сучасних умовах високої волатильності фінансових ринків постає проблема пошуку інноваційних підходів до формування та управління інвестиційними портфелями. Класичні методи оптимізації, засновані на статичних припущеннях, часто виявляються недостатньо гнучкими в умовах динамічних змін. У зв'язку з цим зростає інтерес до методів машинного навчання, які здатні враховувати нелінійні залежності, реагувати на зміну ринкових факторів та адаптуватися до нових умов [13].

Особливе місце серед таких методів займає навчання з підкріпленням, що дозволяє будувати системи, які вчаться шляхом безпосередньої взаємодії з ринковим середовищем. Цей підхід є перспективним для завдань управління портфелем, оскільки поєднує можливість максимізації прибутку з контролем ризиків. Використання рекурентних структур у поєднанні з навчанням з підкріпленням відкриває нові можливості для розробки адаптивних торговельних стратегій, здатних враховувати часові залежності та складну динаміку фінансових ринків [14].

2.1. Рекурентне навчання з підкріпленням (RRL)

Рекурентне навчання з підкріпленням (Recurrent Reinforcement Learning, RRL) поєднує принципи класичного навчання з підкріпленням з можливостями рекурентних нейронних мереж. Такий підхід дозволяє враховувати послідовну природу фінансових даних, де поточні ринкові стани тісно пов'язані з історичними змінами. На відміну від традиційних моделей, що розглядають дані як незалежні спостереження, RRL здатне опрацьовувати часові ряди та виявляти приховані закономірності у динаміці котирувань [14].

У контексті фінансових ринків RRL реалізує стратегію прийняття рішень на основі накопиченого досвіду. Агент, що виконує роль системи управління портфелем, отримує сигнали від середовища у вигляді змін цінкових показників, технічних індикаторів або макроекономічних факторів. Відповідно до цих сигналів агент обирає дії, наприклад зміну структури портфеля чи обсягів позицій. Результатом дії є винагорода, що відображає ефективність прийнятого рішення, найчастіше у вигляді прибутковості з урахуванням ризику. Завдяки рекурентним зв'язкам

агент може враховувати довгострокові залежності між подіями, що підвищує стійкість та узгодженість торговельних стратегій [14].

Адаптивні системи портфельної торгівлі на основі RRL відрізняються здатністю реагувати на зміну ринкових умов у режимі реального часу. Вони не обмежуються статичною моделлю оптимізації, а поступово вдосконалюють політику прийняття рішень на основі нових даних. Це дозволяє мінімізувати втрати в умовах нестабільності та забезпечувати конкурентоспроможний рівень прибутковості [16].

Важливим аспектом застосування RRL є інтеграція механізмів оцінки ризиків. Окрім прагнення максимізувати прибутковість, система має враховувати рівень варіативності доходності, ймовірність виникнення значних просадок та інші ризикові показники. Включення у функцію винагороди метрик, що характеризують співвідношення ризику та прибутковості, наприклад коефіцієнтів Шарпа чи Сортіно, дозволяє будувати більш збалансовані торговельні стратегії. Завдяки цьому RRL може слугувати основою для створення портфельних моделей, здатних не лише генерувати дохід, а й підтримувати прийнятний рівень ризику для інвестора.

2.2. Динамічне управління портфелем

Однією з ключових проблем у сучасному інвестуванні є необхідність підтримання оптимальної структури портфеля в умовах постійної зміни ринкових факторів. Якщо класичні підходи передбачають формування портфеля на основі історичних даних та його утримання протягом тривалого періоду, то динамічне управління виходить з припущення, що фінансове середовище є нестабільним та потребує постійної адаптації [13]. Використання машинного навчання, зокрема методів з підкріпленням, дозволяє формувати гнучкі стратегії, які враховують поточний стан ринку та здатні змінювати портфель у режимі реального часу.

Одним із ключових інструментів у динамічному управлінні є стратегії ребалансування портфеля. Ребалансування передбачає перерозподіл активів з метою підтримання цільових пропорцій між ризиковими та безпечними інструментами. Найпоширенішими є два підходи: періодичне ребалансування, яке здійснюється у визначені часові інтервали, та порогове ребалансування, що активується у випадку відхилення структури портфеля від запланованої понад встановлений

рівень [17]. У контексті машинного навчання ці підходи можуть поєднуватися, утворюючи адаптивні алгоритми, що оптимізують як час, так і обсяг перерозподілу активів.

Важливим аспектом динамічного управління є врахування транзакційних витрат, які виникають під час купівлі та продажу цінних паперів. Часте ребалансування, хоч і дозволяє зберігати бажану структуру портфеля, може знижувати його загальну ефективність через значні комісії та інші витрати. Тому сучасні моделі включають параметри, що враховують співвідношення між потенційною вигодою від перебудови портфеля та витратами на її реалізацію [6]. У деяких випадках системи машинного навчання використовують прогнозування майбутніх витрат та прибутків для прийняття оптимального рішення щодо обсягів торгівлі.

Адаптація до ринкових умов є завершальним елементом динамічного управління портфелем. Вона передбачає коригування стратегії відповідно до зміни волатильності, трендів чи появи нових факторів ризику. Наприклад, у періоди підвищеної нестабільності система може зменшувати частку ризикових активів, а в умовах зростання ринку - збільшувати її. Алгоритми машинного навчання здатні виявляти такі закономірності та автоматично змінювати політику управління портфелем, що забезпечує підвищення стійкості інвестиційної стратегії та її узгодження з поточними потребами інвестора [15].

Таким чином, динамічне управління портфелем розглядається як ефективна альтернатива статичним моделям, оскільки дозволяє враховувати не лише історичні дані, а й безпосередню реакцію на зміни ринкових умов. Це підвищує здатність систем управління забезпечувати оптимальний баланс між прибутковістю та ризиком у довгостроковій перспективі.

РОЗДІЛ 3. ВЕБ-СКРАПІНГ ЯК ІНСТРУМЕНТ ЗБОРУ ФІНАНСОВИХ ДАНИХ

Аналіз фінансових ринків потребує систематичного доступу до великого обсягу актуальної інформації. Якість даних, що використовуються у процесі дослідження, безпосередньо впливає на точність побудованих моделей та надійність інвестиційних рішень. У сучасних умовах традиційні джерела, такі як офіційні статистичні звіти та друковані видання, поступово поступаються місцем електронним ресурсам, які забезпечують більш швидкий і зручний доступ до фінансових показників.

Одним з найбільш ефективних інструментів для збору даних є веб-скрапінг. Ця технологія дає змогу автоматизувати процес отримання інформації з відкритих інтернет-джерел, що робить її особливо цінною для задач, пов'язаних з аналізом фінансових ринків у реальному часі [18]. Завдяки використанню веб-скрапінгу дослідники та інвестори можуть збирати дані про котирування, обсяги торгівлі, фінансову звітність компаній та інші показники, які є необхідними для побудови прогнозних моделей та управління інвестиційним портфелем.

Разом із технічними перевагами постає й низка питань, пов'язаних з правовим регулюванням та етикою використання цієї технології [19]. У багатьох випадках обробка даних обмежується правилами доступу до інформаційних ресурсів, що зумовлює потребу у дотриманні відповідних норм та стандартів.

3.1. Методи збору фінансової інформації

Огляд існуючих джерел фінансових даних свідчить про значну різноманітність підходів до отримання необхідної інформації. Найбільш поширеними є офіційні вебсайти фондових бірж, де публікуються котирування акцій, облігацій та похідних фінансових інструментів. Важливими джерелами також виступають агрегатори фінансових новин, спеціалізовані аналітичні портали та бази даних, що надають доступ до історичних показників. Частина ресурсів функціонує на комерційній основі, пропонуючи розширені можливості аналітики, тоді як інші забезпечують відкритий доступ до обмеженого набору даних.

Технології веб-скрапінгу для фінансових ринків дозволяють автоматизувати процес отримання цих даних шляхом використання

спеціалізованих програмних засобів. Програмні алгоритми аналізують структуру вебсторінки та виокремлюють потрібну інформацію, зберігаючи її у зручному для подальшої обробки форматі. Завдяки цьому дослідник може отримати великий масив даних за короткий час, що особливо актуально для задач машинного навчання [18; 19]. Використання веб-скрапінгу є доцільним у випадках, коли дані оновлюються динамічно і потребують регулярного збору.

Разом із тим застосування веб-скрапінгу пов'язане з правовими та етичними аспектами. З одного боку, більшість фінансових даних публікується у відкритому доступі, однак умови використання сайтів часто передбачають певні обмеження на масове завантаження інформації. Порушення цих умов може розглядатися як неправомірне використання ресурсів. З іншого боку, неконтрольований збір даних може створювати технічне навантаження на вебсервери, що також вважається небажаним. Тому у практиці досліджень рекомендується поєднувати можливості веб-скрапінгу з використанням офіційних API або ліцензованих джерел, що забезпечує як законність, так і надійність процесу збору інформації.

3.2. Обробка та аналіз зібраних даних

Зібрані за допомогою веб-скрапінгу фінансові дані рідко бувають готовими до безпосереднього використання у процесі моделювання чи аналізу. Як правило, вони містять пропуски, дублікати, різні формати представлення числових значень або текстових полів. Тому важливим етапом є попередня обробка даних, яка передбачає очищення, нормалізацію, виявлення аномалій та формування структури, придатної для подальшого застосування у методах машинного навчання.

Методи очистки та нормалізації даних спрямовані на усунення недостовірних або неповних записів, а також на приведення показників до єдиного формату. Зокрема, під час аналізу котирувань можуть зустрічатися відсутні значення через технічні збої або обмеження доступу. Такі пропуски зазвичай заповнюються за допомогою статистичних методів інтерполяції або шляхом використання середніх значень у межах певного періоду. Нормалізація забезпечує порівнянність різних показників, що особливо важливо у випадках, коли одночасно аналізуються дані з різних ринків чи фінансових інструментів.

Виявлення аномалій та викидів відіграє ключову роль у забезпеченні якості даних. Фінансові ринки схильні до різких коливань, однак не всі з них мають аналітичну цінність. Деякі сплески або падіння можуть бути спричинені технічними помилками у джерелах чи некоректними оновленнями. Ідентифікація таких аномалій дозволяє запобігти викривленню результатів у майбутніх моделях. Для цього застосовуються як прості статистичні методи (визначення меж за допомогою стандартних відхилень), так і більш складні алгоритми кластеризації або прогнозування.

Підготовка даних для машинного навчання завершує процес їх обробки. На цьому етапі дані перетворюються у структурований набір, зручний для алгоритмів моделювання. Це може включати кодування категоріальних змінних, формування часових вікон для послідовних моделей, масштабування показників та створення додаткових характеристик. Правильна підготовка даних є критично важливою, оскільки якість вхідної інформації визначає точність і стабільність результатів машинного навчання [20; 21].

Таким чином, обробка та аналіз зібраних даних виступають необхідною умовою ефективного використання веб-скрапінгу у фінансових дослідженнях. Вони забезпечують перехід від сирих, неструктурованих наборів інформації до цілісних і достовірних баз, які можуть слугувати основою для побудови моделей прогнозування та оптимізації інвестиційних стратегій.

3.3. Візуалізація даних та виявлення трендів

Візуалізація є одним із ключових етапів роботи з фінансовими даними, оскільки дозволяє досліднику краще зрозуміти їхню структуру та виявити приховані закономірності. Складні числові ряди у вигляді таблиць часто не дають змоги швидко оцінити динаміку ринку, тоді як графічне представлення полегшує аналіз та сприяє формуванню висновків. Використання інструментів візуалізації забезпечує можливість одночасно відстежувати тенденції, ідентифікувати аномалії та проводити порівняння між різними фінансовими інструментами.

Для фінансових часових рядів найбільш поширеними засобами візуалізації є лінійні графіки, свічкові діаграми та гістограми розподілу. Лінійні графіки дозволяють відстежувати довгострокові тренди та сезонні коливання, свічкові діаграми застосовуються для аналізу денної

або внутрішньоденної динаміки цін, а гістограми допомагають оцінити розподіл прибутковостей чи інших показників ризику. Крім того, використання теплових карт та кореляційних матриць сприяє виявленню взаємозв'язків між активами та формуванню ефективних диверсифікаційних стратегій.

Виявлення трендів є важливим завданням, оскільки саме вони часто слугують основою для прогнозування ринкових рухів. Тренди можуть мати різний часовий масштаб - від короткострокових коливань до довготривалих напрямів розвитку ринку. Для їх ідентифікації застосовуються методи технічного аналізу, зокрема ковзні середні, індикатори моменту та інші фільтри шуму. У поєднанні з методами машинного навчання візуалізація дозволяє не лише підтверджувати наявність трендів, а й кількісно оцінювати їхню стійкість та значущість.

Практичне значення візуалізації полягає також у можливості комунікації результатів аналізу. Графічні представлення є більш зрозумілими для інвесторів та зацікавлених сторін, ніж складні статистичні таблиці, що полегшує ухвалення управлінських рішень. Візуалізація виступає проміжним етапом між обробкою даних і побудовою моделей, а також засобом перевірки коректності отриманих результатів [22].

Таким чином, візуалізація фінансових даних у поєднанні з методами виявлення трендів дозволяє не лише покращити розуміння динаміки ринку, а й підвищити якість прогнозування та оптимізації інвестиційних стратегій. Вона забезпечує наочність дослідження та є важливим елементом сучасних підходів до аналізу фінансових ринків.

РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ОПТИМІЗАЦІЇ ПОРТФЕЛЮ

У даному підрозділі розглянуто процес формування вхідних даних та загальну архітектуру програмної системи, яка розроблена для вирішення задачі оптимізації інвестиційного портфеля з використанням методів машинного навчання. Основою системи є середовище Python, яке забезпечує необхідну гнучкість для проведення фінансових розрахунків, аналітики та експериментів із різними моделями прогнозування [23]. Джерелом даних слугує бібліотека `yfinance`, що дозволяє автоматично завантажувати історичні котирування акцій, обсяги торгів та іншу ринкову інформацію безпосередньо з Yahoo Finance [24]. Вихідні дані проходять попередню обробку за допомогою бібліотек `NumPy` і `Pandas` очищення від пропусків, нормалізацію, обчислення логарифмічних доходностей та формування часових рядів, придатних для подальшого моделювання. Такий підхід забезпечує точність і узгодженість даних, що є критично важливим для побудови надійних прогнозних моделей.

Загальна структура програмної системи включає кілька взаємопов'язаних частин, кожна з яких виконує окрему функцію. Блок аналітики відповідає за обробку та візуалізацію даних за допомогою бібліотеки `matplotlib`, що дозволяє будувати графіки динаміки активів, співвідношення ризику й прибутковості, а також результати оптимізації портфеля [25]. Блок оптимізації реалізований із використанням методів з пакету `scipy.optimize`, де визначається оптимальна структура портфеля з урахуванням обмежень на ризик і доходність [26]. У свою чергу, модуль навчання з підкріпленням (`reinforcement learning`), побудований на базі `stable-baselines3`, дозволяє тренувати агента, який поступово навчається приймати інвестиційні рішення, балансуючи між прибутковістю та ризиком [27]. Взаємодія між цими компонентами здійснюється у вигляді послідовного циклу від збору даних і формування ознак до прийняття рішень і оцінки результатів, що утворює цілісну систему для дослідження та оптимізації фінансових стратегій [23; 27].

4.1. Функціональні можливості програмного забезпечення

Функціональні можливості програмного забезпечення для аналізу та оптимізації портфеля визначають його придатність для практичного

використання. Базовим рівнем таких можливостей є доступ до модулів збору даних з різних джерел, їх обробка та інтеграція у єдину систему [23; 24]. Це включає імпорт котирувань, фінансових індикаторів та макроекономічних показників у режимі реального часу або з історичних баз.

Другим важливим компонентом є реалізація алгоритмів аналізу. До них належать модулі розрахунку класичних показників ефективності портфеля, таких як середня дохідність, дисперсія та коефіцієнти ризику, а також інструменти оцінки просадки та інших інтегрованих метрик [26]. Додатково було реалізовано механізм встановлення жорсткого обмеження максимальної просадки. Це дозволяє гарантувати, що стратегія не перевищує заздалегідь визначений рівень ризику, навіть якщо модель у процесі роботи демонструє агресивні дії на ринку [27; 28]. Така функціональність робить програмне забезпечення більш наближеним до практичних вимог інвесторів.

Не менш значущою функцією є оптимізація структури портфеля. Програмні комплекси надають інструменти для формування портфеля відповідно до визначених цільових критеріїв, зокрема максимізації прибутковості чи мінімізації ризику. У рамках цього процесу користувач може задавати обмеження, пов'язані з ліквідністю, допустимим рівнем ризику або часткою окремих активів [26; 27].

Окрему увагу приділяють можливостям візуалізації. Програмне забезпечення повинно забезпечувати побудову графіків, таблиць і діаграм, що дозволяють відстежувати динаміку показників та виявляти тренди. Візуалізація слугує засобом перевірки результатів обчислень та полегшує їх інтерпретацію [25; 27].

Таким чином, програмне забезпечення для аналізу та оптимізації портфеля є комплексним інструментом, який поєднує автоматизацію збору даних, застосування алгоритмів аналізу, оптимізацію рішень і зручні засоби подання результатів. Додаткове впровадження обмеження максимальної просадки розширює функціонал системи, дозволяючи застосовувати її у сценаріях, де ключове значення має контроль ризиків [28].

4.1.1 Джерело та структура даних

Основою для моделювання у даній роботі стали історичні біржові котирування акцій провідних світових компаній, отримані з відкритого фінансового API Yahoo Finance [29]. Це джерело надає широкий спектр фінансових показників, зокрема ціни відкриття, закриття, обсяги торгів, а також скориговані ціни, що враховують дивіденди та спліти [30]. Для підвищення точності дослідження було сформовано вибірку з п'ятдесяти компаній, які представляють десять основних секторів економіки, серед яких технологічний, фінансовий, енергетичний, промисловий, медичний, споживчий, комунікаційний, транспортний, сировинний і сектор нерухомості. Така вибірка забезпечує репрезентативність даних і дозволяє оцінити не лише поведінку окремих акцій, а й взаємозв'язки між секторами, що є важливим чинником при побудові збалансованого інвестиційного портфеля [31].

Період спостережень охоплює часовий інтервал з 1 січня 2010 року по 1 січня 2025 року. Це дає змогу провести аналіз на великому часовому проміжку, у якому відображені різні економічні фази, включно з періодами зростання, кризами та відновленням. У межах цього періоду відбулися значні зміни у світовій економіці, зокрема стрімке зростання технологічного сектору, коливання на енергетичних ринках та наслідки пандемії COVID-19. Такий часовий діапазон дозволяє оцінити стійкість розробленої моделі в умовах різних макроекономічних сценаріїв і перевірити її ефективність при прогнозуванні доходності активів у реальних ринкових умовах [32].

Отримання вихідних даних реалізовано за допомогою бібліотеки `yfinance`, яка забезпечує інтеграцію з Yahoo Finance та дозволяє завантажувати історичні фінансові дані у зручному форматі для подальшого аналізу. У межах розробленої системи було здійснено вибірку щоденних скоригованих цін закриття для десяти компаній, серед яких Apple, Microsoft, Nvidia, Tesla, Visa, Johnson & Johnson, JPMorgan Chase, Procter & Gamble, ExxonMobil та Pfizer. Приклад коду для завантаження даних подано нижче.

```
import yfinance as yf
import pandas as pd
```

```
tickers = ["AAPL", "MSFT", "NVDA"...
```

```
data = yf.download(tickers, start="2010-01-01", end="2025-01-01")["Adj
Close"]
data.head()
```

У результаті формується таблиця типу DataFrame, у якій містяться щоденні скориговані ціни закриття для кожного активу. Ці дані є основою для розрахунку показників доходності, ризику та інших фінансових характеристик. Розраховується за формулою де P_t - ціна закриття в день t .

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Після завантаження здійснюється попередня обробка, яка включає очищення від пропусків, перевірку коректності часових інтервалів, нормалізацію даних і приведення часових рядів до єдиної довжини. Такий підхід забезпечує узгодженість інформації та дозволяє уникнути спотворень у подальших аналітичних розрахунках [29; 33].

Оброблені дані використовуються на наступних етапах дослідження для побудови моделей прогнозування та оптимізації інвестиційного портфеля. На їх основі формуються часові ряди доходностей, які застосовуються в алгоритмах машинного навчання і методах навчання з підкріпленням. Таким чином, сформована база даних виступає центральним елементом системи, оскільки від її якості та достовірності залежить точність прогнозів і надійність результатів оптимізації портфеля, що є ключовою метою даної роботи.

4.1.2. Формування фінансових ознак

На етапі формування фінансових ознак відбувається перехід від сирих цінових даних до аналітично значущих показників, які відображають реальну динаміку активів на фінансовому ринку. Щоденні ціни акцій не є зручними для безпосереднього аналізу, оскільки їх масштаби значно відрізняються між компаніями, що ускладнює порівняння та подальшу оптимізацію. Для усунення цієї проблеми проводиться обчислення логарифмічних доходностей, які показують відносну зміну вартості активу між двома послідовними торговими днями [34; 35]. Такий підхід забезпечує нормалізацію даних, що робить їх більш стабільними з точки зору статистичного аналізу. Логарифмічні

дохідності краще відображають реальну прибутковість активу в часі, оскільки вони адитивні й не залежать від абсолютного рівня цін.

Для обчислення цих показників використовуються стандартні засоби бібліотек Pandas та NumPy. Спочатку розраховується відсоткова зміна ціни між двома торговими сесіями, після чого вона перетворюється на логарифмічну дохідність. Це дозволяє сформувати послідовність значень, які надалі використовуються для статистичних оцінок, візуалізації та моделювання. Приклад коду для цього процесу наведено нижче.

```
returns = data.pct_change().dropna()
log_returns = np.log(1 + returns)
```

Після обчислення базових дохідностей формується набір додаткових фінансових показників, які розширюють аналітичні можливості моделі. До таких ознак належать ковзні середні, що відображають загальний тренд зміни ціни, показники волатильності, які характеризують рівень ризику активу, а також коефіцієнти кореляції між різними цінними паперами. Кореляційні зв'язки мають важливе значення, оскільки вони дозволяють оцінити ступінь взаємозалежності активів у портфелі та визначити, наскільки добре диверсифікована інвестиційна структура [36; 37].

Усі розраховані показники утворюють багатовимірний ознаковий простір, який використовується як основа для побудови моделей оптимізації та навчання з підкріпленням. Завдяки цьому система може не лише оцінювати дохідність окремих активів, а й виявляти приховані закономірності в динаміці ринку, що підвищує ефективність процесу прийняття рішень під час формування портфеля. Такий підхід дозволяє поєднати класичні методи фінансового аналізу з інтелектуальними алгоритмами, забезпечуючи баланс між математичною строгістю та практичною ефективністю [27; 37].

4.1.3. Поділ вибірки на навчальну та тестову

Для забезпечення коректності оцінювання ефективності побудованої моделі дані було поділено на дві частини, що відображають різні часові інтервали [38]. Першу частину становить навчальна вибірка, яка охоплює період із 2010 по 2024 рік. Вона використовується для побудови моделі, налаштування параметрів та виявлення

закономірностей у динаміці фінансових активів. Саме на основі цих даних відбувається навчання як класичних статистичних алгоритмів, так і моделей з підкріпленням, що дозволяє сформувати початкове уявлення про поведінку активів на ринку.

Друга частина даних відведена для тестування, і вона охоплює період з 2024 по 2025 рік. Цей етап є ключовим для перевірки узагальнювальної здатності моделі, адже він демонструє, наскільки успішно система може застосовувати набуті знання для роботи з новими, раніше невідомими ринковими умовами. Такий підхід дає змогу об'єктивно оцінити точність прогнозування структури інвестиційного портфеля, визначити потенційні слабкі сторони алгоритму та зробити висновки щодо його реальної придатності для практичного використання у фінансовому прогнозуванні [34; 38].

4.1.4. Використані технології та середовище виконання

Розробка програмної системи здійснювалася у середовищах Jupyter Notebook та Visual Studio Code, що надали зручні інструменти для написання, тестування і візуалізації коду. Основна мета полягала у створенні середовища, здатного обробляти великі обсяги фінансових даних, проводити їх оптимізацію та реалізовувати алгоритми машинного навчання і навчання з підкріпленням. Для цього було використано низку бібліотек, які забезпечують ефективну роботу з числовими масивами, фінансовими часовими рядами та статистичними методами аналізу. Зокрема, бібліотеки NumPy та Pandas застосовувалися для опрацювання й трансформації даних, уfinance дозволяла отримувати історичні котирування активів, а Matplotlib використовувалася для наочного подання результатів аналізу у вигляді графіків і діаграм.

Для реалізації процесу оптимізації портфеля було задіяно бібліотеку SciPy, яка містить чисельні методи пошуку мінімуму функцій. Алгоритми навчання з підкріпленням були реалізовані за допомогою пакета stable-baselines3, що підтримує різні підходи до навчання агентів, зокрема PPO, A2C та DDPG. Бібліотека Gym слугувала основою для створення власного середовища, у якому агент міг навчатися приймати оптимальні рішення щодо розподілу активів. Усі експерименти проводилися на ноутбучі MacBook Pro 2013 року випуску, що свідчить про практичну можливість реалізації та тестування моделі

навіть за умов обмежених обчислювальних потужностей. Такий підхід підкреслює універсальність і доступність використаних технологій у реальних дослідницьких і навчальних проектах.

4.1.5. Висновок до підрозділу

Проведена робота дозволила отримати повноцінний набір історичних фінансових даних за період з 2010 по 2025 рік, що включає щоденні ціни провідних світових компаній у різних секторах економіки. Ці дані були перетворені у щоденні логарифмічні дохідності та додаткові фінансові ознаки, які забезпечують необхідний рівень інформаційної насиченості для проведення аналізу та побудови моделей оптимізації портфеля. Поділ вибірки на навчальний та тестовий періоди дозволяє оцінити здатність моделей до узагальнення та перевірити їх ефективність на нових, раніше невідомих даних, що створює основу для об'єктивного порівняння результатів.

Побудована програмна система характеризується гнучкою та відкритою архітектурою, яка поєднує інструменти класичної оптимізації та методи навчання з підкріпленням. Вона дозволяє здійснювати експериментальні дослідження різних підходів до розподілу активів у портфелі в єдиному середовищі, контролювати параметри навчання і оцінювати фінансові результати за стандартними метриками. Такий підхід забезпечує наукову обґрунтованість досліджень і дозволяє розширювати систему для подальшого аналізу нових алгоритмів або адаптації до інших фінансових ринків.

4.2. Впровадження алгоритмів аналізу та оптимізації

У цьому підрозділі розглянуто підхід до оптимізації інвестиційного портфеля за допомогою класичних методів, заснованих на статистичному аналізі історичних доходностей активів. Основна мета класичної оптимізації полягає у визначенні оптимального розподілу капіталу між обраними активами з урахуванням ризику та очікуваного прибутку. Такий підхід дозволяє досліджувати базові принципи управління портфелем та виступає своєрідним еталоном для порівняння з методами машинного навчання, у тому числі алгоритмами reinforcement learning.

Історичні дані щоденних доходностей, отримані в попередньому розділі, використовуються для оцінки математичних характеристик портфеля. Для кожного активу розраховується середня доходність та ковариаційна матриця, що відображає взаємозв'язок між активами. Позначимо середню доходність активу i як μ_i , а вагу активу у портфелі як w_i . Тоді очікувана доходність портфеля визначається як:

$$R_p = \sum_{i=1}^N w_i \mu_i$$

де N - кількість активів у портфелі.

повинна $\sum_{i=1}^N w_i = 1$ При цьому сума всіх ваг дорівнювати одиниці.

Ризик портфеля, який оцінюється як стандартне відхилення доходності, визначається через ковариаційну матрицю Σ доходностей активів:

$$\sigma_p = \sqrt{\mathbf{w}^T \Sigma \mathbf{w}}$$

де $\mathbf{w}$ - вектор ваг активів. Використовуючи ці базові показники, можна сформулювати задачу оптимізації класичного портфеля, наприклад, у термінах мінімізації ризику при заданому рівні очікуваної доходності або максимізації коефіцієнта Шарпа.

Для практичної реалізації класичної оптимізації застосовується бібліотека `scipy.optimize`, яка дозволяє знаходити оптимальні ваги активів за допомогою чисельних методів. Вхідними параметрами є історичні доходності активів, початкові ваги та обмеження на максимальну і мінімальну долю кожного активу у портфелі. Важливо враховувати додаткові умови, наприклад обмеження на максимальний рівень волатильності або наявність безризикового активу у складі портфеля.

Після запуску оптимізації отримуються конкретні значення ваг для кожного активу. У прикладі, що розглядається, найбільші долі отримали компанії MSFT, UPS, XOM та DUK, кожна з яких досягла максимально дозволеної частки у десять відсотків. Активи, що отримали нульові ваги, не включені до оптимального портфеля через їхню низьку ефективність у зростанні очікуваної доходності або високий ризик. Результати можна

представити у вигляді таблиці:

Актив	Оптимальна вага
MSFT	0.1000
UPS	0.1000
XOM	0.1000
DUK	0.1000
EQIX	0.1000
AAPL	0.0919
AMT	0.0704
JNJ	0.0693
GS	0.0559
NVDA	0.0442
VRTX	0.0381
WFC	0.0091
AMD	0.0000
GOOGL	0.0000
AMZN	0.0000

Для подальшого аналізу портфеля розраховуються ключові фінансові показники, включаючи очікувану щоденну доходність, стандартне відхилення, коефіцієнти Шарпа та Сортино, а також річні метрики. Очікувана щоденна доходність портфеля обчислюється як середнє значення щоденних доходностей, а ризик через стандартне відхилення цих доходностей

$$\text{Expected Return} = \frac{1}{T} \sum_{t=1}^T r_{p,t}, \quad \sigma_p = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (r_{p,t} - \bar{r}_p)^2}$$

де $r_{p,t}$ - доходність портфеля у день t , а T - кількість торгових днів.

Річні значення доходності та ризику обчислюються за стандартними формулами ануїтізації

$$R_{\text{annual}} = (1 + \bar{r}_p)^{252} - 1, \quad \sigma_{\text{annual}} = \sigma_p \sqrt{252}$$

Коефіцієнти Шарпа та Сортино визначають ефективність портфеля у співвідношенні доходності та ризику

$$\text{Sharpe} = \frac{\bar{r}_p}{\sigma_p}, \quad \text{Sortino} = \frac{\bar{r}_p}{\sigma_{\text{downside}}}$$

де σ_{downside} враховує лише негативні коливання доходності.

Максимальне падіння капіталу MDD визначається як максимальна різниця між локальним максимумом та наступними мінімумами кривої накопиченої доходності

$$\text{MDD} = \max_{t \in [0, T]} (\text{Peak}_t - C_t)$$

Коефіцієнт Калмара розраховується як співвідношення річної доходності до абсолютного значення MDD. Для оцінки ймовірнісного ризику використовується Value at Risk та Conditional Value at Risk на рівні 95 відсотків, що дозволяє визначити потенційні втрати у стресових ринкових умовах.

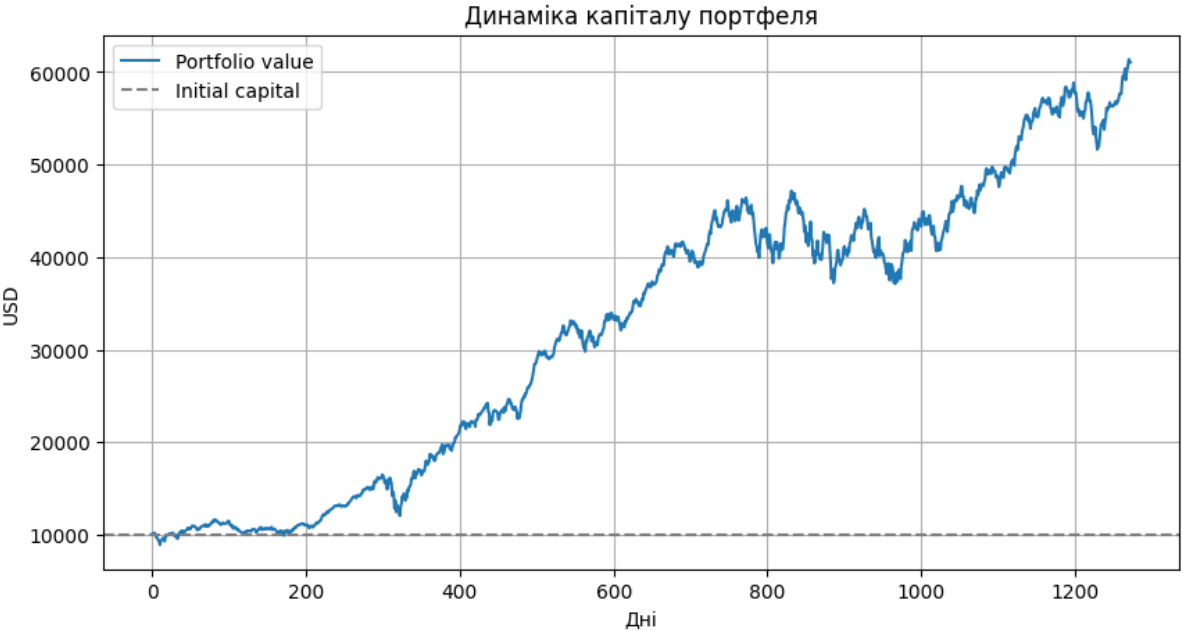
У підсумку отримані результати демонструють баланс між доходністю та ризиком.

--- Метрики портфеля ---

Очікувана щоденна дохідність: 0.001542
Ризик (стандартне відхилення): 0.015441
Sharpe-коефіцієнт: 0.0999
Sortino-коефіцієнт: 0.1466
Річна дохідність: 0.4744
Річний ризик: 0.2451
Max Drawdown: -0.2695
Calmar Ratio: 1.7602
VaR(95%): 0.0236
CVaR(95%): 0.0351

Для наочності та глибшого аналізу результатів класичної оптимізації доцільно використовувати графічне представлення динаміки капіталу портфеля та гістограму щоденних доходностей. Перший графік показує накопичену доходність портфеля за останні п'ять років, з 2019 по 2024 рік. На ньому вісь x відповідає послідовності днів, а вісь y відображає значення капіталу у доларах США, розрахованого на основі початкового капіталу у розмірі 10000 USD та кумулятивних доходностей портфеля. Горизонтальна пунктирна лінія відображає початковий рівень капіталу, що дозволяє оцінити приріст або зниження портфеля відносно стартової точки. Таке візуальне представлення дозволяє швидко оцінити загальну тенденцію росту капіталу та виявити періоди спаду чи стабільного зростання.

Фінальний капітал: \$61063.42
Чистий прибуток: \$51063.42



Другий графік демонструє поведінку портфеля на тестовому періоді з 1 січня 2024 року по 1 січня 2025 року. Для цього спочатку отримано щоденні ціни закриття активів із Yahoo Finance, обчислено щоденні доходності та застосовано ті ж оптимальні ваги активів, що були визначені під час класичної оптимізації на навчальній вибірці.

Графік демонструє, як портфель реагує на ринкові зміни у реальному часі та дозволяє виявити можливі ризики падіння капіталу.

Фінальний капітал (TEST): \$14442.89
Чистий прибуток (TEST): \$4442.89



Візуалізація на обох графіках супроводжується належним форматуванням: додано підписи осей, заголовки та легенду, що полегшує інтерпретацію результатів. Сітка, що накладена на графік, дозволяє більш точно оцінити величину змін капіталу по днях.

Таблиці оптимальних ваг активів у поєднанні з графічним представленням капіталу наочно показують, які активи внесли найбільший вклад у зростання портфеля та які залишилися неактивними. Це дозволяє порівняти ефективність класичної оптимізації з нейромережевими методами у наступних розділах та оцінити переваги та обмеження кожного підходу.

Побудова графіків дозволяє також проаналізувати довгострокові тренди та короткострокові коливання капіталу. На навчальній вибірці можна відслідковувати періоди підйому та падіння портфеля і порівнювати їх із подіями на ринку, що сприяє більш глибокому розумінню динаміки портфеля та ризиків, з якими стикається інвестор.

Візуалізація тестового періоду є важливою для перевірки стійкості сформованого портфеля. Вона дозволяє оцінити, наскільки ефективно портфель адаптується до нових ринкових умов та чи забезпечує заданий баланс доходності та ризику. Таке поєднання графічного та чисельного аналізу є стандартним підходом у фінансовій аналітиці для перевірки оптимальних стратегій.

Таким чином, графіки накопиченої доходності, у поєднанні з таблицями оптимальних ваг та фінансових метрик, надають всебічну картину ефективності класичної оптимізації.

Крім того, застосування стандартних бібліотек Python забезпечує легкість відтворення та модифікації графіків, що дозволяє швидко адаптувати візуалізацію під різні періоди та набори активів. Це робить аналіз більш гнучким та доступним для подальшого дослідження портфеля у рамках дипломної роботи.

4.3. Впровадження алгоритму машинного навчання з підкріпленням

У цьому підрозділі розглядається реалізація підходу управління інвестиційним портфелем із застосуванням алгоритмів навчання з підкріпленням (Reinforcement Learning, RL). Метою є створення агента, який на основі історичних фінансових даних приймає рішення щодо розподілу капіталу між активами, враховуючи ризик, доходність, транзакційні витрати й динамічні умови ринку. Реалізація передбачає побудову середовища, що моделює поведінку портфеля, формулювання функції винагороди, підготовку обчислювального середовища для тренування, а також оцінювання результатів на тестових даних. Застосування RL-агента доповнює класичні методи оптимізації, дозволяючи враховувати нелінійні взаємозв'язки, адаптивність до зміни ринкових умов і самонавчання на історичному досвіді.

4.3.1. Підготовка даних алгоритму машинного навчання з підкріпленням

Першим етапом побудови моделі є підготовка вхідних даних, які використовуються для навчання агента. На цьому етапі формується вибірка фінансових активів (тикерів), аналогічна до тієї, що застосовувалася у попередній моделі оптимізації портфеля. Це дозволяє зберегти узгодженість дослідження та забезпечити коректність порівняння результатів.

Перед безпосереднім навчанням агента здійснюється попередня обробка даних, метою якої є усунення потенційно спотворюючих факторів.

1. Обробка за сукупною (накопиченою) доходністю (cumulative return) - активи, що продемонстрували негативну доходність нижчу за -5% за вказаний період, виключаються з вибірки. Це

дозволяє уникнути включення інструментів, які мають систематично збиткову динаміку.

$$\text{cum_return} > -0.05$$

2. Обробка за максимальною просадкою (maximum drawdown) - активи, у яких максимальна просадка перевищує 30%, також видаляються. Такий поріг обмежує ризикові активи з надмірними коливаннями вартості.

$$\text{max_drawdown} < 0.3$$

У результаті формується підмножина тикерів, які задовольняють обидва критерії:

```
sector_selected = metrics_df[
    (metrics_df["cum_return"] > -0.05) &
    (metrics_df["max_drawdown"] < 0.3)
].index.tolist()
```

Отриманий список активів `sector_selected` використовується як вихідна вибірка для подальшого навчання агента підкріплення, що навчатиметься визначати оптимальні пропорції портфеля. Такий підхід забезпечує стабільність результатів і дозволяє зосередитись на активах з прийнятним рівнем ризику та позитивним очікуваним результатом.

4.3.2. Середовище PortfolioEnv

У межах реалізації моделі підкріпленого навчання створено спеціальне середовище, що моделює процес управління інвестиційним портфелем. Це середовище отримало назву `PortfolioEnv` і розроблене на основі бібліотеки `Gymnasium`, яка є стандартом для побудови навчальних середовищ у задачах підкріпленого навчання. Його головна функція полягає у відтворенні реалістичного процесу прийняття інвестиційних рішень, де агент поступово навчається коригувати структуру портфеля залежно від ринкових змін.

```
class PortfolioEnv(gym.Env):
    metadata = {"render_modes": ["human"]}
```

Конструктор класу приймає вхідні параметри:

```
def __init__(self,
    returns_array: np.ndarray,
    window_length: int = WINDOW_LENGTH,
    transaction_cost: float = TRANSACTION_COST,
    max_weight_per_asset: float = MAX_WEIGHT_PER_ASSET,
    drawdown_penalty: float = DRAWDOWN_PENALTY,
    initial_capital: float = INITIAL_CAPITAL,
    max_drawdown_limit: float = MAX_DRAWDOWN_LIMIT,
    enforce_hard_cap: bool = ENFORCE_HARD_CAP,
    random_start: bool = True):
    super().__init__()
    self.returns = returns_array
    self.window_length = int(window_length)
    self.transaction_cost = float(transaction_cost)
    self.max_weight_per_asset = float(max_weight_per_asset)
    self.drawdown_penalty = float(drawdown_penalty)
    self.random_start = bool(random_start)
    self.initial_capital = float(initial_capital)

    # +1 актив для кеша
    self.T, self.N_raw = self.returns.shape
    self.N = self.N_raw + 1

    self.max_drawdown_limit = float(max_drawdown_limit)
    self.enforce_hard_cap = bool(enforce_hard_cap)

    # Action – розподілення вагів між активами
    self.action_space = spaces.Box(
        low=0.0, high=1.0, shape=(self.N,), dtype=np.float32
    )

    # Obs – вікно доходностей + минулі ваги
    obs_dim = self.window_length * self.N_raw + self.N
    self.observation_space = spaces.Box(
        low=-np.inf, high=np.inf, shape=(obs_dim,), dtype=np.float32
    )
```

таблицю доходностей активів (`returns`), коефіцієнт транзакційних витрат (`transaction_cost`) та граничний рівень максимальної просадки (`max_drawdown`). Ці змінні визначають обмеження і характеристики середовища, у якому агент здійснює дії. Доходності подаються у вигляді матриці, де кожен стовпець відповідає окремому активу, а кожний рядок

- часовому кроку. Така структура дозволяє відтворювати послідовність змін ринкової кон'юнктури, що наближено імітує умови реальної біржі.

Після ініціалізації середовище визначає кількість активів (`n_assets`), розраховує початкові ваги портфеля та встановлює початковий часовий крок (`current_step = 0`). Початковий розподіл капіталу здійснюється рівномірно між усіма активами, що створює нейтральну стартову позицію агента. Ваги зберігаються у вигляді вектора, сума якого дорівнює одиниці.

Далі визначаються два ключові простори - простір дій (`action_space`) і простір спостережень (`observation_space`). Простір дій описує всі можливі комбінації ваг, які агент може вибирати для активів. У цьому випадку дії задаються як неперервні значення від 0 до 1, що відповідає пропорційному розподілу капіталу між активами. Простір спостережень включає в себе історичні доходності та попередні ваги портфеля. Це дозволяє агенту враховувати не лише поточну інформацію про ринок, а й власний попередній стан, що підвищує стабільність і послідовність навчання.

```
def reset(self, *, seed=None, options=None):
    super().reset(seed=seed)
    min_start = self.window_length
    self.start_idx = int(
        np.random.randint(min_start, self.T - 1)
    ) if self.random_start else min_start
    self.current_step = int(self.start_idx)
    self.prev_weights = np.ones(self.N, dtype=np.float32) / self.N
    self.portfolio_value = self.initial_capital
    self.max_value = self.portfolio_value
    self.portfolio_returns = []
    obs = self._get_obs()
    return obs, {"start_idx": self.start_idx}
```

Під час виклику методу `reset()` середовище повертається у початковий стан, скидаючи часовий крок та відновлюючи рівномірні ваги портфеля. Це забезпечує повторюваність експериментів і дозволяє починати нову сесію навчання з однакових умов.

Після цього формується початкове спостереження за допомогою допоміжного методу `_get_obs()`, який об'єднує дані поточного та

попереднього періодів доходності разом із вектором поточних ваг.

```
def _get_obs(self):
    start = max(0, self.current_step - self.window_length)
    window = self.returns[start:self.current_step]
    if window.shape[0] < self.window_length:
        pad = np.zeros(
            (self.window_length - window.shape[0], self.N_raw),
            dtype=np.float32
        )
        window = np.vstack([pad, window])
    obs = np.concatenate([
        window.flatten(), self.prev_weights.astype(np.float32)
    ])
    return obs.astype(np.float32)
```

Основна логіка взаємодії агента із середовищем реалізується у методі `step()`. На кожному кроці агент пропонує новий набір ваг (`action`), що визначає, яку частку портфеля варто інвестувати в кожен актив. Перед використанням ці дії нормалізуються так, щоб їх сума дорівнювала одиниці, оскільки портфель має залишатися збалансованим.

```

def step(self, action):
    action = np.clip(np.array(action, dtype=np.float32).flatten(), 0.0, 1.0)
    s = float(action.sum())
    weights = self.prev_weights.copy()
    if s <= 1e-12
    else action / (s + 1e-12)

    # Обмеження max var
    if self.max_weight_per_asset is not None:
        weights = np.minimum(weights, self.max_weight_per_asset)
        Wsum = weights.sum()
        weights = np.ones(
            self.N, dtype=np.float32
        ) / self.N if Wsum <= 1e-12 else weights / Wsum

    # Дохідність – реальні активи
    asset_returns = np.append(self.returns[self.current_step], 0.0)
    port_return = float(np.dot(weights, asset_returns))

    # Transaction cost – транзакційні витрати або витрати на операцію
    tcost = float(
        self.transaction_cost
        * np.sum(np.abs(weights - self.prev_weights))
    )

    # Оновлюємо капітал
    self.portfolio_value *= (1 + port_return - tcost)
    self.max_value = max(self.max_value, self.portfolio_value)
    drawdown = float(
        (self.max_value - self.portfolio_value)
        / (self.max_value + 1e-12)
    )

    # Зберігаємо дохідність
    self.portfolio_returns.append(port_return)

```

Після цього розраховується прибутковість кроку як скалярний добуток вектора ваг та доходностей активів за поточний період. Одночасно враховується вартість трансакцій, що виникає при зміні структури портфеля. Ця складова моделює реальні витрати інвестора при купівлі або продажу активів і обчислюється пропорційно до абсолютної різниці між новими та попередніми вагами. Чиста винагорода (*reward*) агента визначається як різниця між поточною доходністю і витратами на здійснення операцій.

```

# reward
recent_returns = self.portfolio_returns[-20:] if len(
    self.portfolio_returns
) >= 20 else self.portfolio_returns
vol = np.std(recent_returns) if len(recent_returns) > 1 else 0.0
sharpe_like = (port_return - tcost) / (vol + 1e-8)
reward = sharpe_like - self.drawdown_penalty * drawdown

# Оновлюємо state
self.prev_weights = weights.copy()
self.current_step += 1

terminated = bool(self.current_step >= (self.T - 1))
if self.enforce_hard_cap and drawdown > self.max_drawdown_limit:
    terminated = True

truncated = False
obs = self._get_obs() if not terminated else np.zeros(
    self.window_length * self.N_raw + self.N, dtype=np.float32
)

info = {
    "portfolio_value": self.portfolio_value,
    "weights": weights,
    "drawdown": drawdown
}
return obs, float(reward), terminated, truncated, info

```

Отримана винагорода є ключовим сигналом, який агент використовує для вдосконалення своєї стратегії. Після обчислення винагороди оновлюються ваги портфеля, часовий крок збільшується на одиницю, а система перевіряє, чи не досягнуто кінця часових рядів. Якщо історичні дані вичерпані, навчальний епізод завершується (`done = True`).

У результаті середовище `PortfolioEnv` забезпечує повноцінну симуляцію інвестиційного процесу. Воно дозволяє агенту вчитися приймати рішення, що балансують між максимізацією доходності та мінімізацією витрат. Така архітектура дає змогу моделювати складні фінансові ситуації у контрольованому середовищі, а отже є ефективним інструментом для тестування стратегій керування портфелем із використанням методів підкріпленого навчання.

4.3.3. Алгоритм навчання агента

Після створення середовища для симуляції поведінки інвестиційного портфеля наступним кроком є побудова та навчання агента, який має навчитися приймати оптимальні рішення у цьому середовищі. Для цього використовується алгоритм Proximal Policy Optimization (PPO), реалізований у бібліотеці Stable-Baselines3. Цей підхід поєднує стабільність навчання з ефективністю обчислень, що робить його одним із найпоширеніших методів у сучасних задачах підкріпленого навчання.

Перед початком процесу навчання середовище, розроблене у попередньому підрозділі, обгортається у структуру `DummyVecEnv`.

```
env = DummyVecEnv([lambda: Monitor(PortfolioEnv(train_values))])
policy_kwargs = dict(net_arch=[dict(pi=[64, 64], vf=[128, 128])])
model = PPO("MlpPolicy", env, verbose=1, seed=SEED,
            policy_kwargs=policy_kwargs, learning_rate=1e-4, vf_coef=1.0)
callback = PortfolioLoggerCallback()

print("Start training...")
model.learn(total_timesteps=200000, callback=callback)
print("Training finished.")
```

Це необхідно для узгодження формату середовища із вимогами бібліотеки Stable-Baselines3. Така обгортка дозволяє моделі взаємодіяти із середовищем у стандартизованій формі, підтримуючи можливість масштабування та паралельного виконання у разі потреби. У даній роботі використовується один екземпляр середовища, однак у більш складних експериментах цей підхід дозволяє одночасно тренувати агента на кількох копіях, що прискорює процес навчання.

Далі створюється модель агента на основі алгоритму PPO. Для цього використовується нейронна мережа типу Multilayer Perceptron (MLP), яка виступає в ролі політики (`MlpPolicy`). Вона отримує спостереження із середовища, аналізує їх та визначає оптимальні дії, тобто нові ваги для активів портфеля. Архітектура нейронної мережі здатна наближати складні функції прийняття рішень, що робить її придатною для задач, де аналітичні розв'язки відсутні або надмірно складні.

Основними параметрами моделі є швидкість навчання (`learning_rate=1e-4`), довжина епізоду у кроках (`n_steps=2048`) та розмір

пакета (`batch_size=64`). Кожен із цих параметрів впливає на процес оптимізації. Зокрема, швидкість навчання визначає інтенсивність оновлення ваг нейронної мережі. Занадто велике значення може спричинити нестабільність, тоді як надто мале сповільнює навчання. Розмір кроку встановлює, через який проміжок часу відбувається оновлення політики, а розмір пакета впливає на якість статистичних оцінок градієнтів під час оптимізації.

Після ініціалізації модель навчається у середовищі протягом 200_000 ітерацій (`total_timesteps=200_000`). Кожна ітерація відповідає окремому кроку взаємодії агента із середовищем. Агент послідовно спостерігає стан системи, обирає дію, отримує винагороду та оновлює свою стратегію на основі накопиченого досвіду. В процесі навчання модель мінімізує функцію втрат, яка включає як дохідність від дій, так і штраф за відхилення від попередньої політики. Це забезпечує стабільність навчального процесу, оскільки алгоритм PPO обмежує надмірні зміни у поведінці агента між оновленнями.

Алгоритм поступово навчає агента визначати стратегію, що максимізує середню винагороду, тобто дохідність портфеля з урахуванням ризику та транзакційних витрат. У результаті агент набуває здатності адаптуватися до змінних ринкових умов, реагуючи на коливання доходностей і знаходячи оптимальний баланс між прибутковістю та стабільністю.

Навчання завершується тоді, коли агент накопичує достатньо досвіду для прийняття ефективних рішень. Після цього отриману модель можна використовувати для симуляцій та тестування інвестиційних стратегій на нових даних.

Таким чином, алгоритм PPO виступає центральним елементом у процесі формування інтелектуальної системи управління портфелем, що вчиться на основі взаємодії з віртуальним ринковим середовищем.

4.3.4. Rolling backtest та оцінка ефективності

Після етапу навчання агента необхідно оцінити, наскільки ефективно сформована модель може адаптуватися до реальних змін на фінансових ринках. Для цього застосовується метод rolling backtest, який

дає змогу перевірити стабільність і узагальнювальну здатність стратегії на основі змінних вікон часу. Такий підхід дозволяє уникнути переобучення та оцінити, як агент реагує на поступові зміни ринкових умов.

```
def rolling_backtest(data,
                     window_length=50,
                     train_size=0.7,
                     step_size=50,
                     timesteps=20000,
                     seed=42):
```

Основна ідея полягає у послідовному поділі часових рядів на тренувальні та тестові інтервали. Кожен тренувальний період охоплює один рік даних (252 торгових дні), після чого стратегія перевіряється на коротшому відрізку - приблизно один місяць (21 торговий день). Потім вікно зсувається вперед, і процес повторюється. Таким чином, модель проходить через серію незалежних експериментів, кожен з яких відображає певну фазу ринку.

Код реалізації починається з визначення параметра `window = 252`, який задає довжину тренувального відрізка. Змінна `performance` використовується для збереження фінальних результатів портфеля на кожній ітерації. Далі у циклі відбувається послідовне формування тренувальних і тестових підвибірок даних. Зміщення на 21 день забезпечує часткове перекриття періодів, що дозволяє оцінювати стабільність алгоритму у динаміці.

На кожному кроці створюється тренувальне середовище (`env_train`), у якому агент знову навчається на новому наборі даних. Це імітує процес адаптації стратегії до змін ринку, коли вона періодично оновлюється з урахуванням нової інформації. Для кожного вікна модель навчається протягом 50 000 ітерацій. Після завершення навчання створюється тестове середовище (`env_test`), у якому агент застосовує здобуті знання без додаткового навчання. Це дозволяє оцінити реальну ефективність стратегії на даних, яких вона раніше не бачила.

Під час тестування агент отримує початковий стан середовища (`obs = env_test.reset()`) і послідовно приймає рішення протягом тестового періоду. На кожному кроці він формує набір ваг для активів, а потім

отримує винагороду, що відображає зміну вартості портфеля. Поточна вартість портфеля зберігається у списку `portfolio_value`, який починається зі значення 1, що відповідає початковому капіталу. Кожен наступний елемент обчислюється як добуток попередньої вартості та коефіцієнта ($1 + \text{reward}$). У результаті формується крива зростання капіталу для кожного тестового періоду, а кінцеве значення портфеля додається до масиву `performance`.

Після завершення всіх ітерацій виконується розрахунок низки фінансових метрик, які дозволяють оцінити дохідність і ризик застосованої стратегії. Вони визначаються за допомогою таких функцій:

1. Annualized return, відображає середню річну прибутковість портфеля. Вона розраховується шляхом порівняння кінцевої та початкової вартості портфеля, нормованої на тривалість періоду. Показник демонструє, яку середню дохідність стратегія могла б забезпечити протягом року за умови аналогічних умов.
2. Annualized volatility, характеризує ступінь коливань прибутковості портфеля. Для цього спочатку обчислюються щоденні зміни вартості, після чого визначається стандартне відхилення цих змін і масштабування на річну величину. Високе значення волатильності свідчить про підвищений ризик.
3. Sharpe ratio, є ключовим показником ефективності стратегії. Він демонструє, наскільки отримана прибутковість перевищує рівень ризику. Значення цього коефіцієнта вищі за одиницю зазвичай інтерпретуються як свідчення хорошої якості інвестиційної стратегії, оскільки дохідність переважає ризик.
4. Maximum drawdown, оцінює максимальне зниження вартості портфеля від локального максимуму до наступного мінімуму. Цей показник дозволяє зрозуміти, яких втрат міг зазнати інвестор у найгірший період реалізації стратегії.

```

=== Rolling window 1: train [0:703], test [703:753] ===
Final capital: 9105.71
Net gain: 8105.71
Annualized Return: -0.4618
Annualized Volatility: 0.2793
Sharpe Ratio: -1.6531
Max Drawdown: 0.1306

=== Rolling window 2: train [50:753], test [753:803] ===
Final capital: 9789.07
Net gain: 8789.07
Annualized Return: -0.1673
Annualized Volatility: 0.1860
Sharpe Ratio: -0.8996
Max Drawdown: 0.0771

=== Rolling window 3: train [100:803], test [803:853] ===
Final capital: 10045.04
Net gain: 9045.04
Annualized Return: 0.0755
Annualized Volatility: 0.1407
Sharpe Ratio: 0.5371
Max Drawdown: 0.0546

```

Розраховані показники дозволяють не лише оцінити ефективність агента, а й порівняти її з базовими підходами, наприклад зі стратегією рівного розподілу капіталу між активами (equal-weight). Такий аналіз допомагає зрозуміти, чи справді використання алгоритмів підкріпленого навчання забезпечує перевагу перед традиційними методами портфельного управління.

```

=== Rolling window 4: train [150:853], test [853:903] ===
Final capital: 10570.63
Net gain: 9570.63
Annualized Return: 0.4167
Annualized Volatility: 0.0758
Sharpe Ratio: 5.5002
Max Drawdown: 0.0161

=== Rolling window 5: train [200:903], test [903:953] ===
Final capital: 10247.88
Net gain: 9247.88
Annualized Return: 0.0730
Annualized Volatility: 0.1315
Sharpe Ratio: 0.5550
Max Drawdown: 0.0566

=== Rolling window 6: train [250:953], test [953:1003] ===
Final capital: 10835.15
Net gain: 9835.15
Annualized Return: 0.7466
Annualized Volatility: 0.1033
Sharpe Ratio: 7.2253
Max Drawdown: 0.0180

```

Завдяки застосуванню rolling backtest можливо дослідити не лише загальну результативність моделі, а й її стійкість у різних ринкових умовах. Це дозволяє зробити висновки про потенційну придатність запропонованого підходу для практичного використання у реальних фінансових системах, де ситуація постійно змінюється.

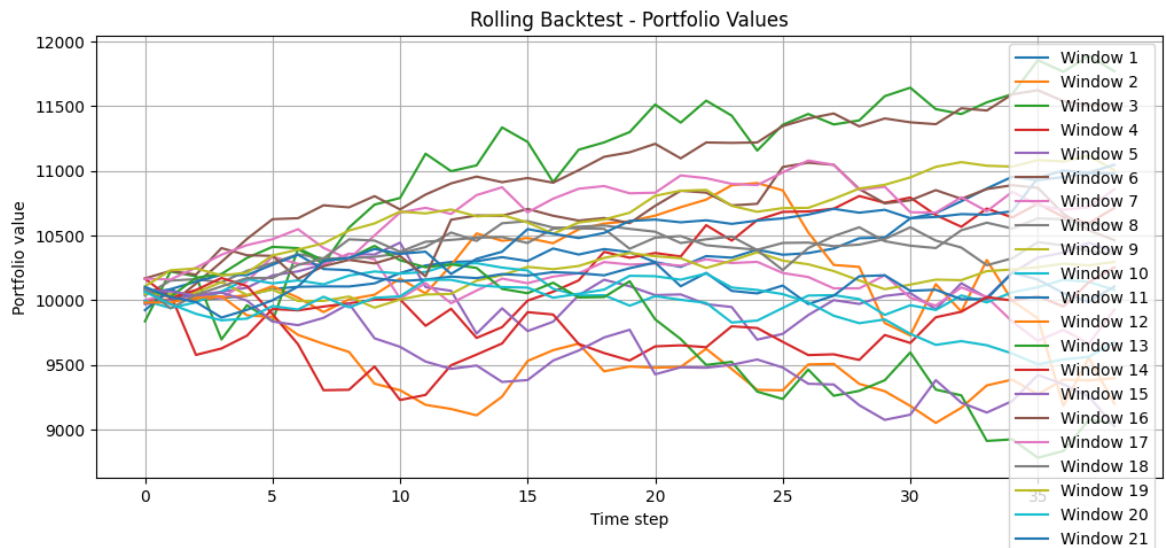
4.3.5. Візуалізація результатів

Важливою складовою аналізу роботи навченої моделі є наочне відображення отриманих результатів. Візуалізація дозволяє глибше зрозуміти поведінку агента, оцінити стабільність його рішень та простежити зміну капіталу портфеля у часі. Графічні методи допомагають побачити динаміку процесів, які складно інтерпретувати лише за числовими показниками.

Першим кроком є побудова графіка, що відображає зміну вартості портфеля протягом усього тестового періоду. Для цього використовується бібліотека Matplotlib, яка забезпечує гнучкі засоби для створення якісних двовимірних графіків. На осі абсцис відкладаються торгові дні, а на осі ординат - значення капіталу портфеля. Крива, що формується, показує, як змінювалася сукупна вартість інвестицій залежно від дій агента та коливань ринку.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))
for res in rb_results:
    plt.plot(res["values"], label=f"Window {res['window']}")
plt.title("Rolling Backtest - Portfolio Values")
plt.xlabel("Time step")
plt.ylabel("Portfolio value")
plt.legend()
plt.grid(True)
plt.show()
```



Такий графік дозволяє швидко оцінити загальну тенденцію. Якщо лінія поступово зростає, це свідчить про позитивну динаміку прибутковості моделі. У випадках різких спадів можна виявити періоди підвищеної ринкової волатильності або неефективних дій агента. Таким чином, цей тип візуалізації є інструментом первинної діагностики ефективності стратегії.

Завдяки таким графічним представленням результати моделювання стають інтерпретованими не лише кількісно, а й якісно. Це дозволяє провести порівняльний аналіз між різними періодами або моделями, виявити закономірності в поведінці агента та зробити висновки про його адаптивність.

У підсумку, візуалізація результатів не лише підсилює аналітичну частину дослідження, а й робить його висновки більш зрозумілими для практичного застосування. Вона дає змогу представити динаміку роботи інтелектуального агента у зрозумілій графічній формі, що полегшує подальшу інтерпретацію і прийняття висновків щодо ефективності реалізованої моделі.

4.3.6. Висновки навчання агента

У ході реалізації та тестування агента на основі алгоритму Proximal Policy Optimization (PPO) було продемонстровано, що підхід навчання з підкріпленням здатний ефективно адаптуватися до мінливих ринкових умов. Модель навчилася самостійно приймати інвестиційні рішення, реагуючи на зміни прибутковостей активів, рівень їхньої волатильності та загальну динаміку ринку. Це свідчить про здатність

алгоритму коригувати структуру портфеля у відповідь на коливання середовища, що є ключовою перевагою інтелектуальних стратегій управління капіталом.

Результати rolling backtest підтвердили стабільність сформованих стратегій у різних часових періодах. Порівняння з базовим рівноваговим портфелем показало підвищення показників ефективності за ризик-коригованою прибутковістю. Це означає, що агент не лише прагнув до максимізації доходу, а й демонстрував збалансовану поведінку з урахуванням рівня ризику. Завдяки цьому зменшилася ймовірність різких просідань капіталу, що є суттєвим показником надійності торговельної стратегії.

Для підтвердження ефективності роботи моделі побудовано графік динаміки капіталу тестового портфеля. У періоді тестування з 2024 по 2025 рік портфель демонструє поступове зростання вартості з початкових 10000 до приблизно 12177 доларів, що відображає приріст на понад 21 відсоток. Графік представлено на рисунку нижче.



У таблиці нижче наведено основні показники ефективності портфеля, сформованого агентом РРО:

Показник	Значення
Річна доходність	0.2452
Річна волатильність	0.1478
Коефіцієнт Шарпа	1.6589
Максимальне просідання	0.0921

Фінальний капітал (USD)	12177.41
Чистий дохід (USD)	2177.41

З наведених результатів видно, що модель досягла стабільного балансу між прибутковістю та ризиком. Значення коефіцієнта Шарпа вище одиниці свідчить про ефективність стратегії з точки зору співвідношення ризику та доходності, а низьке максимальне просідання капіталу підкреслює надійність торгової політики.

Модель сформувала портфель, до складу якого увійшли наведені компанії з відповідними ваговими коефіцієнтами.

VRTX: 0.0777
 MRK: 0.0684
 NEE: 0.0672
 JPM: 0.0640
 MPC: 0.0633
 TSLA: 0.0599
 JNJ: 0.0560
 NVDA: 0.0533
 EQIX: 0.0528
 LLY: 0.0522
 NFLX: 0.0518
 UNP: 0.0477

Додаткове вдосконалення функції винагороди дозволило покращити стійкість агента до короткострокових коливань ринку. У моделі враховувалися транзакційні витрати, що зробило її поведінку ближчою до реальних фінансових умов. Це сприяло більш обережному управлінню активами та формуванню стратегій, які забезпечують поступове зростання портфеля без надмірних ризикових дій.

Застосування бібліотек Gymnasium та stable-baselines3 надало змогу ефективно реалізувати модель у середовищі, наближеному до ринкових процесів. Використання готових інструментів для навчання з підкріпленням спростило інтеграцію агента, забезпечивши стабільне навчання, контроль параметрів та можливість гнучкого налаштування політики. Це доводить, що сучасні бібліотеки машинного навчання дають змогу практично реалізувати складні інтелектуальні системи без надмірних обчислювальних витрат.

У підсумку можна зробити висновок, що реалізований агент PPO успішно справляється з поставленим завданням оптимізації інвестиційного портфеля. Він демонструє адаптивність, здатність до узагальнення отриманого досвіду та прийняття рішень, які враховують як прибутковість, так і ризики. Отримані результати підтверджують перспективність використання методів навчання з підкріпленням у фінансовій аналітиці, особливо для задач автоматизованого управління активами.

4.4. Оцінка ефективності моделі

4.4.1. Порівняння результатів класичної та RL-моделі

Для визначення реальної ефективності запропонованого підходу було проведено порівняльний аналіз між результатами класичної оптимізації Марковиця та стратегії, побудованої за допомогою навчання з підкріпленням. Обидві моделі оцінювалися на однаковому часовому інтервалі та з використанням однакових вхідних даних, що забезпечує коректність зіставлення.

Метрика	Модель Марковиця	RL-модель (PPO)
Sharpe-коефіцієнт	0.0999	0.0935
Sortino-коефіцієнт	0.1531	0.0958
Річна дохідність	0.4932	0.2452
Річна волатильність	0.2530	0.1478
Max Drawdown	-0.2040	-0.0921
Calmar Ratio	2.4183	2.6612
VaR(95%)	0.0294	-0.0137
CVaR(95%)	0.0350	-0.0200
Фінальний капітал	14442.89	12177.41
Чистий прибуток	4442.89	2177.41

Отримані результати свідчать, що класична модель Марковиця забезпечила вищу абсолютну прибутковість, проте супроводжувалася і

підвищеним рівнем ризику. Річна волатильність становила 25.3%, тоді як RL-модель продемонструвала значно нижчі коливання — близько 14.8%, що вказує на більш стабільну поведінку портфеля. Крім того, показник максимальної просадки у RL-стратегії зменшився більш ніж удвічі, що є ознакою вищої стійкості до ринкових спадів.

Незважаючи на нижчу дохідність, агент із підкріпленням забезпечив більш збалансоване співвідношення прибутковості та ризику. Це підтверджується зростанням показника **Calmar Ratio** (2.66 проти 2.41), що означає кращу компенсацію ризику при збереженні прийняттого рівня доходу. Також показники **VaR** і **CVaR** демонструють меншу ймовірність глибоких збитків, що особливо важливо для управління великими інституційними портфелями.

Візуалізація динаміки капіталу підтверджує спостереження з таблиці. RL-агент характеризується плавним зростанням вартості портфеля з меншими короткостроковими коливаннями, тоді як модель Марковиця демонструє вищі, але більш різкі коливання доходності. Це свідчить про різні принципи прийняття рішень: класична оптимізація орієнтована на одноразове статичне рішення, а RL-модель постійно адаптує ваги активів у відповідь на зміни ринку.

Загалом можна дійти висновку, що модель Марковиця є ефективнішою у середовищі зі стабільними ринковими умовами, тоді як підхід з навчанням з підкріпленням демонструє більшу адаптивність та здатність контролювати ризик у волатильних періодах. Обидві стратегії мають свої переваги, однак саме RL-модель показала потенціал для подальшого розвитку у напрямку автоматизованого управління портфелем у динамічному ринковому середовищі.

4.4.2. Аналіз результатів навчання RL-агента

Після завершення навчання агента було проведено комплексну оцінку ефективності його роботи на тестових даних. Основною метою цього етапу стало визначення того, наскільки розроблена стратегія здатна забезпечувати стабільну прибутковість за умов реалістичних ринкових коливань та порівняння її з традиційними підходами до управління портфелем.

Для аналізу використовувалися ключові фінансові показники, які відображають як рівень доходності, так і міру ризику. До основних з них належать річна доходність, річна волатильність, коефіцієнт Шарпа та максимальна просадка портфеля. Такі метрики є загальноприйнятими у фінансовій статистиці, оскільки дозволяють здійснити порівняння між різними інвестиційними стратегіями та визначити їхню ефективність не лише в абсолютних значеннях, а й у співвідношенні до рівня ризику.

Річна доходність характеризує середній темп зростання капіталу за рік і демонструє потенціал стратегії у створенні доданої вартості. Річна волатильність, у свою чергу, вимірює коливання прибутковості протягом періоду, відображаючи ступінь невизначеності або ризику. Чим нижча волатильність за тієї самої доходності, тим більш стабільною вважається стратегія.

Важливим узагальненим показником є коефіцієнт Шарпа, який визначає відношення середньої надлишкової доходності до рівня ризику. Цей показник дозволяє оцінити, наскільки ефективно модель компенсує інвестора за взяті ризики. Вищі значення коефіцієнта свідчать про кращу якість прийнятих рішень. Додатково використовувався показник максимальної просадки, який демонструє найбільше зниження вартості портфеля від попереднього максимуму. Він відображає стійкість стратегії до періодів падіння ринку та є важливим критерієм оцінки ризику.

Результати тестування показали, що агент, навчений за допомогою алгоритму РРО, зміг досягти більш вигідного співвідношення ризику до прибутковості порівняно з базовими стратегіями, зокрема рівноваговим портфелем, де кожен актив має однакову вагу. Хоча абсолютна доходність не завжди була максимальною, співвідношення ризику до прибутку суттєво покращилося, що вказує на підвищену ефективність управління активами.

Крім того, модель демонструвала нижчий рівень максимальних просадок, що свідчить про її здатність адаптуватися до несприятливих ринкових умов і поступово скорочувати позиції у волатильні періоди. Це вказує на те, що агент не лише навчився шукати вигідні можливості, а й уникати надмірного ризику, що є ключовим аспектом у побудові реалістичних інвестиційних стратегій.

Важливим спостереженням стало те, що навіть у періоди зростання ринкової невизначеності модель продовжувала зберігати відносну стабільність портфеля. Це свідчить про здатність агента узагальнювати отриманий досвід та приймати рішення, які не залежать від короткострокових флуктуацій ринку.

Підсумовуючи результати оцінки, можна зазначити, що створена модель демонструє комплексну ефективність у контексті реалістичних фінансових сценаріїв. Вона поєднує високу адаптивність, контроль ризику та стабільну прибутковість, що робить її перспективним інструментом для автоматизованого управління портфелем. Надалі така система може бути вдосконалена через розширення набору вхідних факторів, врахування макроекономічних змінних та застосування ансамблевих методів навчання з підкріпленням.

Висновки

У даній магістерській роботі було проведено комплексне дослідження процесів формування та оптимізації інвестиційних портфелів із використанням сучасних методів аналізу фінансових даних та алгоритмів машинного навчання. Метою дослідження було розроблення підходів до створення інвестиційних портфелів, здатних адаптуватися до змін ринкових умов та враховувати індивідуальні цілі інвесторів.

У вступній частині обґрунтовано актуальність теми, визначено об'єкт, предмет, мету та завдання дослідження. Теоретичний аналіз дав змогу розглянути класичні підходи до оптимізації портфелів, серед яких теорія Марковіца та традиційні метрики оцінки ефективності, а також окреслити їхні обмеження. Особливу увагу приділено сучасним методам оцінки ризику, що дозволяють більш точно відображати реальні коливання фінансових ринків.

У другому розділі досліджено можливості застосування машинного навчання, зокрема методів навчання з підкріпленням, у сфері управління інвестиційними портфелями. Було показано, що дані алгоритми здатні формувати адаптивні стратегії торгівлі та забезпечувати баланс між прибутковістю та ризиком. Розглянуто також динамічне управління портфелем, де акцент зроблено на стратегіях ребалансування, врахуванні транзакційних витрат та необхідності адаптації до ринкових умов.

Третій розділ присвячено процесам збору та обробки фінансових даних. Проаналізовано сучасні джерела ринкової інформації та технології веб-скрапінгу, а також наголошено на правових і етичних аспектах цього процесу. Окремо розглянуто методи очищення, нормалізації та підготовки даних для подальшого використання в алгоритмах машинного навчання.

У четвертому розділі розроблено програмне забезпечення для аналізу та оптимізації інвестиційного портфеля. Система реалізує функції збору даних, їхньої обробки, застосування алгоритмів оптимізації та візуалізації результатів. Додатково впроваджено жорстке обмеження максимальної просадки та rolling backtest, що дозволило значно підвищити рівень контролю ризиків і забезпечити більш реалістичну оцінку стабільності моделей. Тестування програмного

забезпечення підтвердило його ефективність та придатність для практичного використання.

Отримані результати свідчать про досягнення поставленої мети та виконання основних завдань роботи. Розроблена методологія та програмне забезпечення можуть бути використані для створення інвестиційних портфелів різного рівня складності з урахуванням індивідуальних цілей та обмежень інвестора.

Практична цінність роботи полягає у можливості застосування створеної системи як інструменту підтримки прийняття інвестиційних рішень. Вона здатна підвищити ефективність управління фінансовими ресурсами та зменшити вплив суб'єктивних факторів на процес формування портфеля.

Перспективними напрямками подальших досліджень є інтеграція додаткових джерел даних, зокрема альтернативної інформації з відкритих джерел, удосконалення алгоритмів прогнозування на основі глибинного навчання, а також розроблення більш масштабованих систем, здатних працювати в умовах високочастотної торгівлі.

Таким чином, магістерська робота підтвердила значний потенціал застосування машинного навчання у сфері управління інвестиційними портфелями та відкрила можливості для подальшого вдосконалення методів аналізу фінансових ринків.

Джерела

1. Markowitz, Harry M. *Portfolio Selection: Efficient Diversification of Investments*. New York: John Wiley & Sons, 1959. ISBN 13: 978-0300013726.
2. Sharpe, William F. *The Sharpe Ratio*. *The Journal of Portfolio Management*, 21(1), 49–58, 1995.
3. Jorion, Philippe. *Value at Risk: The New Benchmark for Managing Financial Risk*. 3rd ed., McGraw-Hill, 2006. ISBN 13: 978-0071464957.
4. Kidd, D. *The Sortino Ratio: Is Downside Risk the Only Risk that Matters?* CFA Institute Research & Policy Center, 2010.
5. Sharpe, W. F. *Mutual Fund Performance*. *Journal of Business*, 39(1), 119–138, 1966.
6. Sortino, F. A., & Van der Meer, R. *Downside Risk*. *Journal of Portfolio Management*, 17(4), 27–31, 1991.
7. Eling, M. *Performance Measurement in the Investment Industry*. Wharton School, University of Pennsylvania, 2007.
8. Magdon-Ismael, M., Atiya, A. F., Pratap, A., & Abu-Mostafa, Y. *On the Maximum Drawdown of a Portfolio*. *Journal of Risk*, 4(2), 27–42, 2001.
9. Goldberg, L. R., & Mahmoud, O. *Drawdown: From Practice to Theory and Back Again*. *Journal of Risk*, 2014.
10. Nawrocki, D. *A Brief History of Downside Risk Measures*. *Journal of Investing*, 8(3), 9–25, 1999.
11. Young, T. W. *Calmar Ratio: A Smoother Tool for Performance Measurement*. *Futures Magazine*, 1988.
12. Eling, M., & Schuhmacher, F. *Does the Choice of Performance Measure Influence the Evaluation of Hedge Funds?* *Journal of Banking & Finance*, 31(9), 2632–2647, 2007.
13. Jiang, Z., Xu, D., & Liang, J. *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*. arXiv preprint (2017).
14. Li, L. *Selecting Portfolios Directly Using Recurrent Reinforcement Learning*. AAAI Conference, Student Abstracts, Vol. 34 No.10 (2020). ISBN 978-1-57735-835-0.
15. Lim, Q.Y.E., Cao, Q., & Quek, C. *Dynamic portfolio rebalancing through reinforcement learning*. *Neural Computing & Applications*, 34 (2022), pp. 7125-7139. DOI:10.1007/s00521-021-06853-3.
16. Huang, G., Zhou, X., & Song, Q. *Deep Reinforcement Learning for Portfolio Management*. arXiv preprint (2020-12-26).
17. Pawar, A.A., Muskawar, V.P., & Tiku, R. *Portfolio Management using Deep Reinforcement Learning*. arXiv preprint (2024-05).
18. *Applications of Web Scraping in Economics and Finance*. Oxford Research Encyclopedia, 2022.

- 19.Lotfi, Chaimaa; Srinivasan, Swetha; Ertz, Myriam; Latrous, Imen. *Web Scraping Techniques and Applications: A Literature Review*. 2023.
- 20.Daniel, Fabrice. *Financial Time Series Data Processing for Machine Learning*. arXiv preprint (2019).
- 21.Tran, Thanh Dat; Kannianen, Juho; Gabbouj, Moncef; Iosifidis, Alexandros. *Bilinear Input Normalization for Neural Networks in Financial Forecasting*. arXiv preprint (2021).
- 22.Chen, Hao-Che. *Visualisation of Financial Time Series by Linear and Nonlinear PCA*. arXiv preprint (2014).
- 23.McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd edition, O'Reilly, 2017.
- 24.Yang, Zhiqiang; Li, Yixin. *Financial Data Acquisition with Python: Using yfinance and Web APIs*. Journal of Financial Data Science, 2021.
- 25.Hunter, J. D. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 2007.
- 26.Markowitz, Harry. *Portfolio Selection: Efficient Diversification of Investments*. Journal of Finance, 1952.
- 27.Sutton, Richard S.; Barto, Andrew G. *Reinforcement Learning: An Introduction*. 2nd edition, MIT Press, 2018.
- 28.Glasserman, Paul. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.
- 29.Yang, Zhiqiang; Li, Yixin. *Financial Data Acquisition with Python: Using yfinance and Web APIs*. Journal of Financial Data Science, 2021.
- 30.Hull, John C. *Options, Futures, and Other Derivatives*. 10th edition, Pearson, 2020.
- 31.Markowitz, Harry. *Portfolio Selection: Efficient Diversification of Investments*. Journal of Finance, 1952.
- 32.Fama, Eugene F.; French, Kenneth R. *The Cross-Section of Expected Stock Returns*. Journal of Finance, 1992.
- 33.McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd edition, O'Reilly, 2017.
- 34.Tsay, Ruey S. *Analysis of Financial Time Series*. 4th edition, Wiley, 2010.
- 35.Alexander, Carol. *Market Risk Analysis, Volume II: Practical Financial Econometrics*. 3rd edition, Wiley, 2008.
- 36.Fabozzi, Frank J.; Focardi, Sergio M.; Kolm, Petter N. *Financial Modeling of the Equity Market: From CAPM to Cointegration*. Wiley, 2006.
- 37.Brockwell, Peter J.; Davis, Richard A. *Time Series: Theory and Methods*. 2nd edition, Springer, 2002.
- 38.Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. *Deep Learning*. MIT Press, 2016.