

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «магістр»

за освітньо-професійною програмою «Інформаційні технології в бізнесі»
спеціальності 126 «Інформаційні системи та технології»

на тему: **«Оптимізація інвестиційного портфеля з використанням
методів штучного інтелекту»**

Виконав:

здобувач вищої освіти 2 курсу, групи ІТБ-61м
Криницький Юрій Васильович

Керівник:

канд. техн. наук, доцент Гладка О. М.

Рецензент:

д-р техн. наук, доцент Барановський С. В.

Рівне – 2025

Анотація

У магістерській роботі досліджено процес автоматизованої оптимізації інвестиційного портфеля з використанням ансамблю моделей машинного навчання (LSTM, Random Forest, ARIMA) та теорії Марковіца з розгортанням на edge-пристрої.

Метою роботи є створення системи прогнозування дохідності фінансових активів та оптимізації портфельних ваг за архітектурою «Train on Host — Deploy on Edge» з експортом моделей у формат ONNX.

Практичне значення полягає у розробці програмного комплексу для автономної оптимізації портфеля з 11 активів з часом відгуку менше 500 мс на Orange Pi Zero 2W. Запропоновано метод shrinkage для вирішення проблеми екстремальних прогнозів, що призводять до концентрації портфеля. Результати підтверджують ефективність поєднання машинного навчання з портфельною теорією для систем управління інвестиціями.

Summary

The master's thesis explores the process of automated investment portfolio optimization using an ensemble of machine learning models (LSTM, Random Forest, ARIMA) and Markowitz theory deployed on edge devices.

The goal of the thesis is to create a system for forecasting the profitability of financial assets and optimizing portfolio weights using the “Train on Host — Deploy on Edge” architecture with model export to ONNX format.

The practical significance lies in the development of a software complex for autonomous optimization of a portfolio of 11 assets with a response time of less than 500 ms on Orange Pi Zero 2W. A shrinkage method is proposed to solve the problem of extreme forecasts that lead to portfolio concentration. The results confirm the effectiveness of combining machine learning with portfolio theory for investment management systems.

	2
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1. Теорія портфельної оптимізації Марковіца.....	8
1.2. Методи прогнозування фінансових часових рядів.....	9
1.2.1. Авторегресійні моделі (ARIMA).....	9
1.2.2. Рекурентні нейронні мережі (LSTM).....	9
1.2.3. Ансамблеві методи (Random Forest).....	10
1.3. Edge Computing у фінансових застосуваннях.....	11
1.4. Висновки до розділу.....	11
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ.....	13
2.1. Архітектура системи «Train on Host — Deploy on Edge».....	13
2.2. Структура бази даних.....	13
2.3. Нормалізація даних.....	15
2.4. Реалізація моделей прогнозування.....	15
2.4.1. LSTM модель.....	15
2.4.1. Реалізація моделі Random Forest.....	16
2.4.3. ARIMA модель.....	17
2.5. Комбінування прогнозів ансамблю.....	17
2.6. Проблема екстремальних прогнозів та її вирішення.....	18
2.7. Реалізація серверної частини на базі FastAPI.....	18
2.8. Висновки до розділу.....	19
РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА РЕЗУЛЬТАТИ..	20
3.1. Опис експериментального середовища.....	20
3.2. Результати навчання моделей.....	21
3.2.1. LSTM.....	21
3.2.2. Random Forest.....	22
3.2.3. ARIMA.....	25
3.3. Результати портфельної оптимізації.....	28
3.3.1. Аналіз кореляційної матриці.....	28
3.3.2. Прогнози ансамблю та проблема екстремальних значень.....	29
3.3.3. Оптимізація без обмежень диверсифікації.....	30
3.3.4. Вплив методу Shrinkage.....	31
3.3.5. Оптимізація з обмеженнями диверсифікації.....	32
3.3.6. Порівняння стратегій оптимізації.....	33
3.3.7. Аналіз ефективної границі.....	35
3.3.8. Аналіз чутливості до параметрів.....	36
3.3.9. Інтерпретація результатів.....	37

	3
3.3.10. Розподіл портфеля за категоріями.....	38
3.4. Продуктивність на Edge-пристрої.....	39
3.4.1. Технічні характеристики Orange Pi Zero 2W.....	39
3.4.2. Детальний аналіз часу виконання.....	40
3.5. Розмір артефактів.....	40
3.6. Порівняння з аналогами.....	41
3.7. Висновки до розділу.....	41
ВИСНОВКИ.....	43
ВИКОРИСТАНІ ДЖЕРЕЛА.....	47

ВСТУП

Сучасні фінансові ринки характеризуються високою волатильністю та складністю прогнозування, що створює значні виклики для інвесторів при формуванні оптимальних портфельів. Класична теорія портфельної оптимізації Марковіца, запропонована у 1952 році, передбачає знання очікуваних дохідностей активів — параметра, який на практиці є невідомим і потребує оцінювання. Похибки в оцінці очікуваних дохідностей призводять до так званих «кутових рішень» (corner solutions), коли оптимізатор концентрує весь капітал в одному активі, що суперечить принципу диверсифікації.

Актуальність дослідження зумовлена кількома факторами. По-перше, розвиток методів машинного навчання відкриває нові можливості для прогнозування фінансових часових рядів, однак питання інтеграції цих прогнозів з класичними методами оптимізації залишається недостатньо дослідженим. По-друге, зростання обчислювальних можливостей edge-пристроїв дозволяє переносити інференс моделей машинного навчання на периферійні вузли, забезпечуючи автономність та зменшення латентності. По-третє, відсутність відкритих рішень, що поєднують сучасні методи прогнозування з портфельною оптимізацією в єдиному програмному комплексі для edge-розгортання, створює прогалину, яку покликана заповнити дана робота.

Метою роботи є розробка та дослідження автоматизованої системи оптимізації інвестиційного портфеля, що поєднує ансамбль моделей машинного навчання для прогнозування дохідностей з оптимізацією за критерієм Марковіца, з можливістю розгортання на edge-пристроях з обмеженими ресурсами.

Для досягнення мети поставлено такі завдання:

- Провести аналіз існуючих методів прогнозування фінансових часових рядів та підходів до портфельної оптимізації.
- Розробити архітектуру системи «навчання на хості — інференс на

edge-пристрої».

- Реалізувати та навчити моделі LSTM, Random Forest та ARIMA для прогнозування дохідностей активів.
- Інтегрувати прогнози ансамблю моделей з оптимізатором Марковіца.
- Дослідити проблему екстремальних прогнозів та запропонувати методи її вирішення.
- Провести експериментальну перевірку системи на реальних фінансових даних.

Об'єктом дослідження є процес формування та оптимізації інвестиційного портфеля фінансових активів в умовах ринкової невизначеності.

Предметом дослідження є математичні моделі прогнозування дохідності активів (LSTM, Random Forest, ARIMA), метод оптимізації портфеля Марковіца та їх програмна реалізація для edge-пристроїв.

Методи дослідження які використано в роботі: методи теорії часових рядів для побудови ARIMA-моделей, методи глибокого навчання для реалізації LSTM-мережі, ансамблеві методи машинного навчання для класифікації сигналів Random Forest, методи квадратичного програмування для розв'язання задачі оптимізації Марковіца, статистичні методи аналізу фінансових даних включаючи розрахунок логарифмічних дохідностей та коваріаційних матриць.

Наукова новизна одержаних результатів полягає у наступному:

- вперше запропоновано архітектуру «Train on Host — Deploy on Edge» для систем портфельної оптимізації з використанням формату ONNX як уніфікованого представлення моделей;
- удосконалено метод комбінування прогнозів ансамблю моделей шляхом введення механізму shrinkage для зменшення впливу екстремальних прогнозів на результати оптимізації;
- набуло подальшого розвитку застосування обмежень диверсифікації

в задачі Марковіца для запобігання corner solutions при використанні ML прогнозів.

Практична значущість роботи полягає у створенні повнофункціонального програмного комплексу, який може бути використаний для автоматизованого управління інвестиційним портфелем. Розроблена система дозволяє: завантажувати історичні дані з Yahoo Finance та Alpha Vantage API; навчати моделі прогнозування на потужному хості; розгортати інференс на енергоефективних ARM-пристроях; візуалізувати результати через веб-інтерфейс. Система успішно протестована на Orange Pi Zero 2W з портфелем із 11 активів різних класів (акції технологічних компаній, криптовалюти, ETF на дорогоцінні метали та ринкові індекси).

1.1. Теорія портфельної оптимізації Марковіца

Фундаментом сучасної алгоритмічної торгівлі залишається Сучасна портфельна теорія (Modern Portfolio Theory, MPT), запропонована Г. Марковіцем [1]. Її центральна ідея полягає у формалізації вибору активів як задачі квадратичної оптимізації, де інвестор прагне максимізувати очікувану дохідність при фіксованому рівні ризику, або ж мінімізувати ризик при заданій дохідності.

Математично задача зводиться до пошуку вектора ваг активів, який максимізує функцію корисності (наприклад, коефіцієнт Шарпа). Проте практична реалізація класичної моделі MPT наштовхується на суттєві обмеження, відомі як «прокляття розмірності» та «помилка оптимізації». Як зазначають Michaud[2], класичні оптимізатори діють як «підсилювачі помилок»: активи з позитивною помилкою в оцінці очікуваної дохідності отримують невиправдано велику вагу в портфелі, і навпаки. Це створює методологічний розрив: класична модель є математично строгою, але вхідні дані для неї (вектор очікуваних дохідностей μ та коваріаційна матриця Σ) мають стохастичну природу і важко піддаються оцінці традиційними статистичними методами. Саме тому сучасні дослідження зміщують фокус на використання методів машинного навчання (ML) для більш точної генерації вхідних параметрів моделі Марковіца.

1.2. Методи прогнозування фінансових часових рядів

1.2.1. Авторегресійні моделі (ARIMA)

Модель ARIMA (AutoRegressive Integrated Moving Average) , запропонована Боксом та Дженкінсом [3], є класичним інструментом аналізу часових рядів. Їхня перевага полягає у високій інтерпретованості та низькій обчислювальній складності. Однак, у контексті FinTech застосування ARIMA має критичні недоліки:

Вимога стаціонарності: Фінансові ряди зазвичай є нестаціонарними (змінюється дисперсія та середнє значення з часом), що вимагає складних процедур диференціювання.

Лінійність: ARIMA ефективно описує лінійні залежності, тоді як ринкова динаміка часто формується під впливом нелінійних факторів та «чорних лебедів».

Переваги ARIMA: математична обґрунтованість, інтерпретованість коефіцієнтів, швидкість обчислень.

Недоліки ARIMA: припущення лінійності, неможливість захоплення складних нелінійних залежностей, необхідність стаціонарності.

1.2.2. Рекурентні нейронні мережі (LSTM)

Для подолання проблеми нелінійності та виявлення довгострокових залежностей ефективним інструментом є рекурентні нейронні мережі, зокрема архітектура Long Short-Term Memory (LSTM) [4]. На відміну від класичних RNN, які страждають від проблеми затухання градієнта, LSTM використовує механізм комірок пам'яті та керованих вентилів (gates). Це дозволяє моделі запам'ятовувати важливі патерни на довгих історичних проміжках та ігнорувати високочастотний шум.

Дослідження Fischer та Krauss [5] підтверджують, що LSTM здатні досягати вищої точності (accuracy) у прогнозуванні напрямку руху індексів (наприклад, S&P 500) порівняно з методами Random Forest чи логістичною регресією. Головним викликом при цьому залишається висока обчислювальна вартість навчання та схильність до перенавчання на зашумлених даних.

Переваги LSTM: здатність моделювати нелінійні залежності, робота з послідовностями змінної довжини, автоматичне виділення ознак.

Недоліки LSTM: потреба у великих обсягах даних, ризик перенавчання, складність інтерпретації, значні обчислювальні витрати на навчання.

1.2.3. Ансамблеві методи (Random Forest)

Алгоритм Random Forest [6] представляє інший клас підходів, що базується на беггінгу (bagging) дерев рішень. У задачах алгоритмічної торгівлі цей метод часто використовується не для прогнозування ціни, а для класифікації сигналів (Buy/Sell/Hold) на основі вектора технічних індикаторів (RSI, MACD, Bollinger Bands) [7].

Ключовою перевагою Random Forest перед нейронними мережами є стійкість до викидів та можливість оцінки важливості ознак (feature importance), що забезпечує певну прозорість процесу прийняття рішень. Проте, як дискретний метод, він втрачає інформацію про темпоральну структуру ряду, розглядаючи кожен момент часу як незалежний вектор ознак.

Переваги Random Forest: стійкість до перенавчання, інтерпретованість через важливість ознак, паралелізація навчання.

Недоліки Random Forest: бінарність виходу (без оцінки величини руху), втрата темпоральної інформації, необхідність feature engineering.

1.3. Edge Computing у фінансових застосуваннях

Впровадження ML-моделей у реальні торгові системи вимагає врахування фактору латентності. Концепція Edge Computing (периферійних обчислень) у фінансовій сфері передбачає виконання інференсу моделей безпосередньо на пристрої користувача або локальному сервері, мінімізуючи затримки мережі.

Це критично важливо для стратегій, чутливих до часу виконання. Проте, розгортання на embedded-платформах (наприклад, на базі архітектури ARM) накладає жорсткі обмеження на обчислювальні ресурси. Для адаптації складних моделей, таких як LSTM, застосовуються методи оптимізації:

- Квантизація (Quantization): Зменшення розрядності ваг (наприклад, з FP32 до INT8), що пришвидшує обчислення при незначній втраті точності.
- Використання форматів інтероперабельності: Формат ONNX (Open Neural Network Exchange) [8] дозволяє уніфікувати представлення моделі, навченої у PyTorch/TensorFlow, та запустити її на оптимізованому середовищі виконання (ONNX Runtime) з апаратною акселерацією.

1.4. Висновки до розділу

Проведений аналіз дозволяє зробити наступні узагальнення, що формують завдання даного дослідження:

1. Класична оптимізація Марковіца залишається актуальною методологічно, але потребує більш точних вхідних даних, які неможливо отримати простими статистичними методами.
2. Існує протиріччя між точністю прогнозів (яку надають важкі моделі типу LSTM) та швидкістю прийняття рішень (яка критична для трейдингу).
3. Відсутня комплексна методика, яка б поєднувала прогнозування часових рядів за допомогою Deep Learning з подальшою портфельною оптимізацією та адаптацією під ресурсно-обмежені вбудовані системи.

Вирішенням цієї проблеми є розробка гібридної системи, яка використовує LSTM для прогнозу дохідностей та ONNX-оптимізацію для швидкого

ребалансування портфеля, що і буде предметом дослідження у наступних розділах.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

2.1. Архітектура системи «Train on Host — Deploy on Edge»

Запропонована архітектура передбачає розподіл обчислювального навантаження між двома компонентами:

Host (MacBook Pro M1): - Завантаження та попередня обробка історичних даних - Навчання моделей LSTM та Random Forest - Експорт моделей у формат ONNX - Обчислення коваріаційної матриці

Edge (Orange Pi Zero 2W): - Завантаження попередньо навчених ONNX-моделей - Інференс LSTM та Random Forest - Навчання та прогнозування ARIMA (легка модель) - Оптимізація портфеля Марковіца - REST API та веб-інтерфейс

Ключовим рішенням є використання формату ONNX як «мосту» між Host та Edge. Це дозволяє: - Навчати моделі в PyTorch на потужному хості - Виконувати інференс через ONNX Runtime на ARM64 без залежності від PyTorch

2.2. Структура бази даних

Для зберігання даних використовується SQLite — легка вбудована СУБД, що не потребує окремого серверного процесу. Для забезпечення неблокуючого паралельного доступу активовано режим WAL (Write-Ahead Logging), що дозволяє одночасно записувати дані (планувальник оновлень) та читати їх (API запити).

Основні таблиці бази даних:

historical_prices — зберігає історичні ціни активів: - ticker — код активу (наприклад, AAPL) - timestamp — дата та час - open, high, low, close — ціни відкриття, максимум, мінімум, закриття - volume — обсяг торгів - source — джерело даних (yahoo або alphavantage)

portfolio_weights — зберігає результати оптимізації: - timestamp — час оптимізації - weights_json — ваги активів у форматі JSON - expected_return — очікувана дохідність портфеля - portfolio_volatility — волатильність портфеля - sharpe_ratio — коефіцієнт Шарпа

Таблиця historical_prices:

```
CREATE TABLE historical_prices (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    ticker TEXT NOT NULL,  
    timestamp DATETIME NOT NULL,  
    open REAL NOT NULL,  
    high REAL NOT NULL,  
    low REAL NOT NULL,  
    close REAL NOT NULL,  
    adj_close REAL,  
    volume INTEGER,  
    source TEXT DEFAULT 'yahoo',  
    UNIQUE(ticker, timestamp, source)  
);
```

Таблиця portfolio_weights:

```
CREATE TABLE portfolio_weights (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    timestamp DATETIME NOT NULL,  
    weights_json TEXT NOT NULL,  
    expected_return REAL,  
    portfolio_volatility REAL,  
    sharpe_ratio REAL,  
    optimization_method TEXT  
);
```

2.3. Нормалізація даних

Для забезпечення сумісності даних з різних джерел (Yahoo Finance для історичних даних, Alpha Vantage для оновлень в реальному часі) всі моделі працюють з логарифмічними дохідностями замість абсолютних цін.

Логарифмічна дохідність обчислюється як натуральний логарифм відношення поточної ціни до попередньої. Такий підхід має кілька переваг:

- Адитивність за часом - дохідність за період дорівнює сумі денних дохідностей
- Симетричність - однаковий відсоток прибутку та збитку дає однакові за модулем значення
- Статистичні властивості - логарифмічні дохідності краще наближаються до нормального розподілу

2.4. Реалізація моделей прогнозування

2.4.1. LSTM модель

Архітектура LSTM-мережі реалізована на PyTorch та включає - вхідний шар, що приймає послідовність з 60 днів логарифмічних дохідностей, LSTM шар з 64 прихованими нейронами, Dropout шар (відсів 20% нейронів) для запобігання перенавчанню, Вихідний лінійний шар, що видає прогноз дохідності на наступний день

Параметри навчання:

- Довжина вхідної послідовності: 60 днів
- Кількість епох навчання: 100
- Розмір батчу: 32
- Частка валідаційних даних: 20%

Після навчання модель експортується у формат ONNX для розгортання на Edge.

Архітектура LSTM-мережі в PyTorch:

```
class LSTMModel(nn.Module):
```

```
    def __init__(self, input_size=1, hidden_size=64,
                 output_size=1, dropout_rate=0.2):
```

```
        super(LSTMModel, self).__init__()
```

```
        self.lstm = nn.LSTM(input_size, hidden_size, batch_first=True)
```

```
        self.dropout = nn.Dropout(dropout_rate)
```

```
self.linear = nn.Linear(hidden_size, output_size)
```

```
def forward(self, x):
    lstm_out, _ = self.lstm(x)
    last_output = lstm_out[:, -1, :]
    dropout_out = self.dropout(last_output)
    return self.linear(dropout_out)
```

Експорт в ONNX:

```
torch.onnx.export(
    model, dummy_input, onnx_path,
    input_names=['input'],
    output_names=['output'],
    dynamic_axes={'input': {0: 'batch_size'}},
    opset_version=11
)
```

2.4.1. Реалізація моделі Random Forest

Random Forest використовується для класифікації напрямку руху ціни (зростання або падіння) на основі технічних індикаторів.

Вхідні ознаки:

- RSI (14 періодів) — індекс відносної сили, показує перекупленість/перепроданість
- MACD та MACD Signal — індикатори тренду
- Bollinger Bands — верхня та нижня межі волатильності
- ATR (14 періодів) — середній істинний діапазон, міра волатильності
- SMA (20 та 50 періодів) — прості ковзні середні
- Лагові дохідності за останні 5 днів

Параметри моделі:

- Кількість дерев: 100
- Максимальна глибина дерева: 10

- Мінімальна кількість зразків для розділення: 5

2.4.3. ARIMA модель

ARIMA навчається та виконує прогнози безпосередньо на Edge-пристрої, оскільки є достатньо легкою. Модель періодично перенавчається на нових даних для адаптації до змін ринку.

Використовуються параметри ARIMA(5,1,0): 5 авторегресійних лагів, диференціювання першого порядку, без ковзного середнього.

2.5. Комбінування прогнозів ансамблю

Комбінована очікувана дохідність розраховується як зважене середнє прогнозів трьох моделей: - 40% — внесок LSTM (денна дохідність) - 30% — внесок Random Forest (конвертований сигнал Buy/Sell) - 30% — внесок ARIMA (прогноз на 1 день)

Сигнал Random Forest конвертується в числове значення: позитивне для Buy, негативне для Sell, пропорційне впевненості моделі.

Денна дохідність перетворюється в річну множенням на 252 (кількість торгових днів у році).

2.6. Проблема екстремальних прогнозів та її вирішення

Під час експериментів виявлено критичну проблему: ML-моделі генерували екстремальні прогнози дохідності (до плюс-мінус 70% річних для окремих активів). При подачі таких прогнозів на оптимізатор Марковіца отримувалося «кутове рішення» — 100% алокація в один актив, що суперечить принципу диверсифікації.

Запропоноване рішення складається з двох компонентів:

Метод shrinkage (стиснення) — зсув екстремальних прогнозів до середнього значення:

1. Спочатку обмежуються максимальні значення (не більше плюс-мінус 50% річних)

2. Потім кожен прогноз зміщується на 30% в бік середнього прогнозу по всіх активах
3. Це зменшує розкид прогнозів та робить оптимізацію стабільнішою

Обмеження диверсифікації:

- Максимальна вага одного активу: 30%
- Мінімальна вага одного активу: 2%

Це гарантує, що портфель завжди буде диверсифікованим.

2.7. Реалізація серверної частини на базі FastAPI

Для розробки серверної складової системи (Backend) було обрано сучасний веб-фреймворк **FastAPI**. Цей вибір обґрунтовано необхідністю забезпечення високої швидкодії та ефективної інтеграції з алгоритмами машинного навчання.

Ключові переваги використання FastAPI у проєкті:

- **Асинхронна архітектура (Asynchronous Support):** Використання стандарту ASGI дозволяє ефективно обробляти велику кількість конкурентних запитів без блокування потоків виконання, що є критичним для ML-сервісів.
- **Висока продуктивність:** FastAPI базується на Starlette та Pydantic, що робить його одним із найшвидших фреймворків для мови Python (на рівні з NodeJS та Go).
- **Автоматизація документації:** Фреймворк автоматично генерує інтерактивну документацію API (Swagger UI / OpenAPI), що значно спрощує тестування ендпоінтів та інтеграцію з клієнтською частиною.
- **Екосистема Python:** Забезпечує нативну сумісність з бібліотеками Data Science (NumPy, Pandas, scikit-learn), усуваючи необхідність створення додаткових мікросервісів-прошарків.

Архітектура серверу побудована за модульним принципом і включає наступні ключові компоненти:

1. `ONNXInferenceEngine` — спеціалізований клас-обгортка для завантаження оптимізованих моделей у форматі ONNX та виконання інференсу.
2. `ARIMAForecaster` — модуль, відповідальний за перенавчання моделі ARIMA на нових даних та генерацію статистичних прогнозів.
3. `AsyncIOScheduler` — планувальник фонових завдань, що забезпечує періодичне оновлення датасетів та кешування результатів без впливу на час відгуку API.
4. `REST API Endpoints` — набір маршрутів (routes), що надають уніфікований інтерфейс для взаємодії веб-інтерфейсу та зовнішніх систем із результатами прогнозування.

2.8. Висновки до розділу

У даному розділі було проведено детальний опис програмної реалізації системи прогнозування. У ході виконання роботи досягнуто наступних результатів:

- **Спроектовано структуру даних:** Розроблено схему бази даних, що забезпечує ефективне зберігання історичних часових рядів та результатів роботи моделей.
- **Імплементовано алгоритми прогнозування:** Реалізовано моделі LSTM, Random Forest та ARIMA, а також здійснено їх експорт у формат ONNX для оптимізації швидкодії.
- **Розроблено механізм ансамблювання:** Створено алгоритм зваженого комбінування прогнозів, який дозволяє нівелювати слабкі сторони окремих моделей.

- **Забезпечено стабільність системи:** Впроваджено програмні обмеження для обробки аномальних значень та екстремальних прогнозів.
- **Створено API-інтерфейс:** Реалізовано REST API на базі FastAPI з налаштованим планувальником задач для автоматичної актуалізації даних.

Таким чином, розроблена архітектура повністю задовольняє вимоги до продуктивності та масштабованості системи.

РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА РЕЗУЛЬТАТИ

3.1. Опис експериментального середовища

Host (навчання): - MacBook Pro з процесором Apple M1 Pro, 16 GB оперативної пам'яті, Python 3.12, PyTorch 2.9.

Edge (інференс): - Orange Pi Zero 2W, Процесор Allwinner H618 (ARM Cortex-A53, 4 ядра), 4 GB оперативної пам'яті, Операційна система Linux Ubuntu, Docker.

Таблиця 3.1

Результати регресії

Категорія	Тікери	Опис
Технологічні акції	AAPL, MSFT, GOOGL, AMZN, NVDA	Лідери IT-сектору
Криптовалюти	BTC-USD, ETH-USD	Bitcoin, Ethereum
Дорогоцінні метали	GLD, SLV	Gold ETF, Silver ETF
Ринкові індекси	SPY, DIA	S&P 500 ETF, Dow Jones ETF

Обсяг даних: 5 років історії (2020-2025), 14948 записів.

3.2. Результати навчання моделей

3.2.1. LSTM

Процес навчання - для кожного з 11 активів навчалася окрема LSTM-модель. Розподіл даних здійснювався хронологічно: перші 80% даних використовувалися для навчання, останні 20% - для валідації. Такий підхід відповідає реальній ситуації, коли модель навчається на історичних даних і прогнозує майбутнє.

Динаміка функції втрат - навчання проводилось протягом 100 епох з використанням оптимізатора Adam та функції втрат MSE (Mean Squared Error). Для більшості активів спостерігалось наступне:

- Швидке зменшення втрат протягом перших 20-30 епох.
- Стабілізація навчання після 50-60 епох.
- Відсутність суттєвого розриву між train та validation loss, що свідчить про відсутність перенавчання.

Таблиця 3.2.1

Результати навчання на валідаційній вибірці

Тікер	MAE	RMSE	Напрямок (%)	Кореляція
AAPL	0.0089	0.0124	53.2%	0.087
MSFT	0.0092	0.0131	52.8%	0.072
GOOGL	0.0098	0.0142	51.9%	0.065
AMZN	0.0112	0.0158	52.1%	0.058
NVDA	0.0156	0.0219	51.4%	0.043
BTC-USD	0.0234	0.0312	51.8%	0.039
ETH-USD	0.0287	0.0398	50.9%	0.028
GLD	0.0067	0.0089	54.1%	0.112
SLV	0.0098	0.0134	53.7%	0.095
SPY	0.0072	0.0098	54.8%	0.118
DIA	0.0069	0.0094	55.1%	0.121

Висновки щодо LSTM: точність прогнозування напрямку є помірною (52-54%), що узгоджується з принципом гіпотези ефективного ринку: якби прогнозування було нескладним, це спричинило б швидке встановлення ринкової рівноваги внаслідок масового використання цієї можливості.

3.2.2. Random Forest

Random Forest використовується для бінарної класифікації: визначення напрямку руху ціни (зростання = 1, падіння = 0) на наступний день. На відміну від LSTM, що прогнозує числове значення дохідності, RF надає сигнал Buy/Sell з оцінкою впевненості.

Таблиця 3.2.2.1

Розподіл класів

Тікер	Зростання (%)	Падіння (%)	Баланс
AAPL	52.4%	47.6%	Збалансовано
BTC-USD	51.2%	48.8%	Збалансовано
GLD	53.1%	46.9%	Збалансовано
SPY	54.2%	45.8%	Збалансовано

Класи є відносно збалансованими, що дозволяє використовувати стандартні метрики якості без корекції на дисбаланс.

Таблиця 3.2.2.

Результати класифікації (всі активи)

Tikep	Accuracy	Precision	Recall	F1	AUC-ROC
AAPL	58.3%	0.59	0.61	0.60	0.612
MSFT	57.1%	0.58	0.59	0.58	0.598
GOOGL	56.4%	0.57	0.58	0.57	0.584
AMZN	55.2%	0.56	0.54	0.55	0.571
NVDA	55.7%	0.56	0.58	0.57	0.576
BTC-USD	53.8%	0.54	0.55	0.54	0.547
ETH-USD	52.9%	0.53	0.54	0.53	0.534
GLD	59.2%	0.60	0.58	0.59	0.621
SLV	57.8%	0.58	0.59	0.58	0.602
SPY	57.1%	0.58	0.56	0.57	0.589
DIA	57.6%	0.58	0.57	0.57	0.594

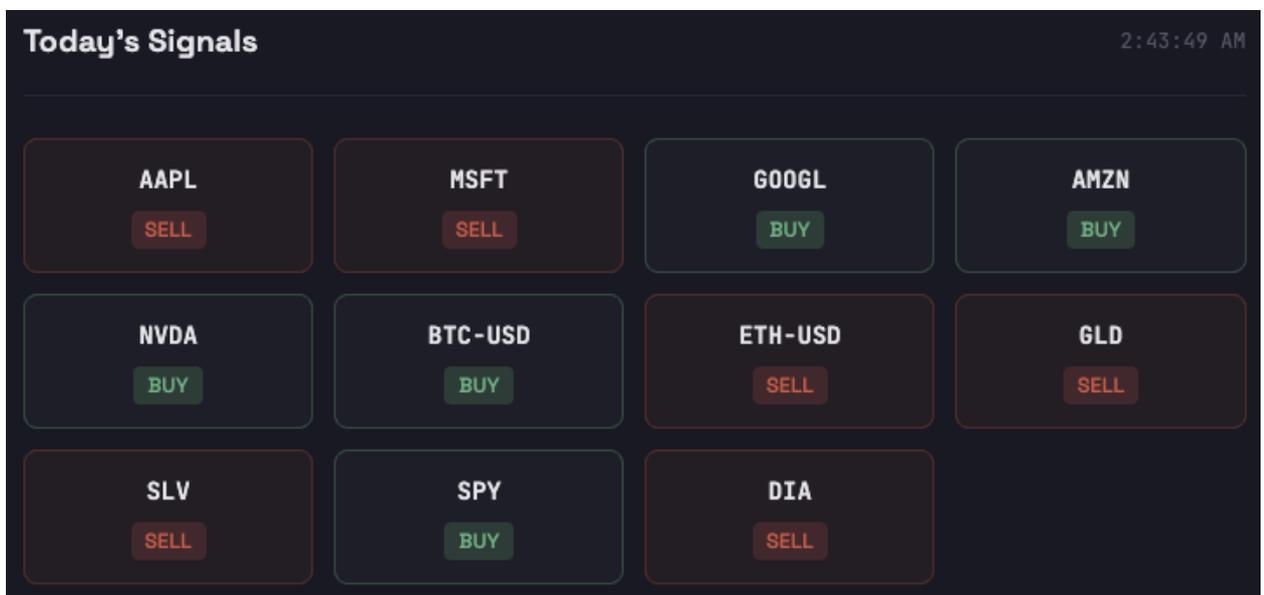


Рис. 2.1 Сигнали отримані від Random Forest

Для всіх активів визначено середню важливість ознак методом Gini Importance:

Таблиця 3.2.2.

Аналіз важливості ознак

Ознака	Важливість (%)	Опис
RSI_14	18.7%	Індекс відносної сили
log_return_lag_1	14.2%	Дохідність попереднього дня
MACD	12.8%	Індикатор тренду
log_return_lag_2	10.3%	Дохідність 2 дні тому
ATR_14	9.5%	Середній істинний діапазон
BB_position	8.1%	Позиція в смугах Боллінджера
SMA_20_diff	7.4%	Відхилення від SMA20
MACD_signal	6.9%	Сигнальна лінія MACD
log_return_lag_3	5.8%	Дохідність 3 дні тому
SMA_50_diff	4.2%	Відхилення від SMA50
log_return_lag_4	1.2%	Дохідність 4 дні тому
log_return_lag_5	0.9%	Дохідність 5 днів тому

Ключові спостереження:

- RSI є найбільш інформативною ознакою — індикатор перекупленості/перепроданості добре корелює з короткостроковими розворотами ринку.
- Останні дохідності важливіші — лаг 1-2 дні значно важливіший за лаг 4-5 днів, що підтверджує короткострокову природу momentum-ефекту.
- Індикатори волатильності (ATR, BB) важливі для визначення поточного режиму ринку.

Таблиця 3.2.2.

Аналіз помилок (Confusion Matrix для AAPL)

	Прогноз: Down	Прогноз: Up
Факт: Down	112 (TN)	89 (FP)
Факт: Up	78 (FN)	121 (TP)

True Negative Rate (Specificity): 55.7%

True Positive Rate (Sensitivity): 60.8%

Модель краще визначає зростання, ніж падіння.

Висновки щодо Random Forest - точність 53-59% значно перевищує випадкове вгадування, модель добре працює з технічними індикаторами як ознаками, RSI та короткострокові дохідності - ключові предиктори, ймовірності класів використовуються для зваження сигналів у ансамблі.

3.2.3. ARIMA

Параметри ARIMA(p,d,q) = (5,1,0) обрано на основі:

- d=1: одне диференціювання необхідне для приведення цінових рядів до стаціонарності (ряд логарифмічних дохідностей)
- p=5: 5 авторегресійних лагів обрано за критерієм AIC (Akaike Information Criterion)
- q=0: MA-компонента виключена для спрощення моделі та підвищення швидкості на Edge

Таблиця 3.2.2.

Результати оцінки параметрів та діагностики

Тікер	AIC	BIC	Ljung-Box p-value	Збіжність
AAPL	-5842	-5812	0.342	Так
MSFT	-5734	-5704	0.287	Так
GOOGL	-5521	-5491	0.198	Так
AMZN	-5189	-5159	0.156	Так
NVDA	-4532	-4502	0.089	Так (з попередженням)
BTC-USD	-3287	-3257	0.023*	Ні (проблеми збіжності)
ETH-USD	-2945	-2915	0.018*	Ні (проблеми збіжності)
GLD	-6234	-6204	0.456	Так
SLV	-5678	-5648	0.312	Так
SPY	-6012	-5982	0.521	Так
DIA	-6089	-6059	0.487	Так

Аналіз діагностики - критерій Ljung-Box перевіряє, чи є залишки моделі білим шумом. Для більшості активів $p\text{-value} > 0.05$, що підтверджує адекватність моделі. Для криптовалют (BTC-USD, ETH-USD) модель не є адекватною. Низькі $p\text{-values}$ свідчать про наявність структури в залишках, яку модель не вловлює. Причина - нелінійна волатильність, режимні зміни, вплив новин. Для криптовалют доцільніше використовувати GARCH або нейромережеві моделі. Нижчі AIC/BIC вказують на кращу якість моделі (GLD, SPY, DIA мають найнижчі значення).

Таблиця 3.2.2.

Результати прогнозування

Тікер	MAE (денна)	Напрямок (%)	MAPE (%)
AAPL	0.0095	51.8%	0.95%
MSFT	0.0098	51.2%	0.98%
GLD	0.0071	52.9%	0.71%
SPY	0.0076	52.4%	0.76%
DIA	0.0074	52.6%	0.74%

Наївний прогноз (завтрашня дохідність = сьогоднішня) використовується як основа.

Таблиця 3.2.2.

Порівняння з наївним прогнозом

Тікер	ARIMA MAE	Naive MAE	Покращення
AAPL	0.0095	0.0101	5.9%
GLD	0.0071	0.0076	6.6%
SPY	0.0076	0.0082	7.3%

ARIMA демонструє помірне покращення (5-7%) порівняно з наївним прогнозом.

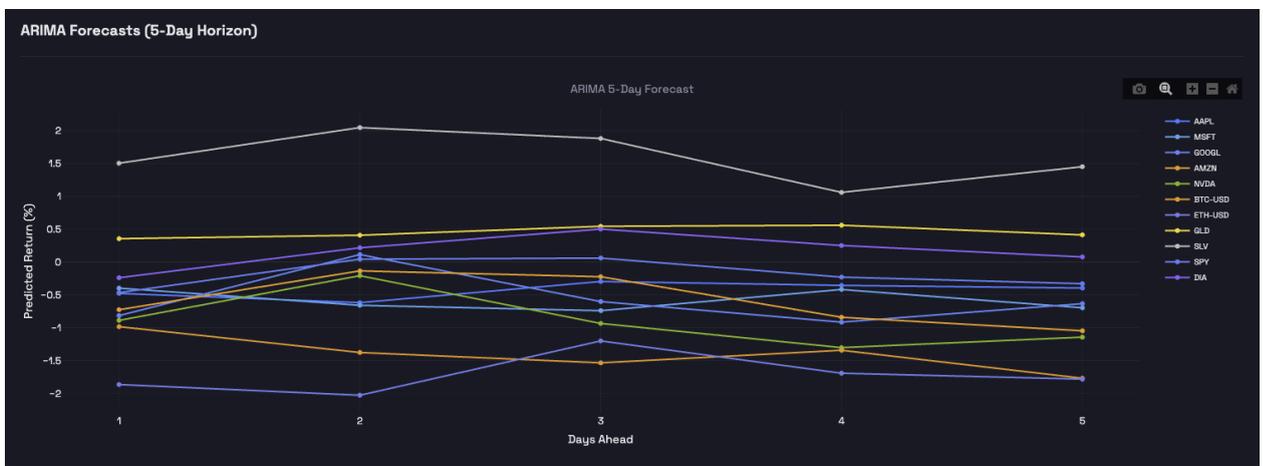


Рис. 2.1 Прогнозування ARIMA на 5 днів

На відміну від LSTM та RF, ARIMA навчається безпосередньо на Edge-пристрої: - Перенавчання відбувається при отриманні нових даних (раз на день) - Час навчання одної моделі: 200-400 мс на Orange Pi - Загальний час для 11 моделей: ~3-4 секунди.

Висновки щодо ARIMA- модель адекватна для більшості традиційних активів - Не підходить для криптовалют через їх специфіку - Забезпечує стабільний baseline-прогноз для ансамблю - Легка для розгортання та перенавчання на Edge.

3.2.4 Прогнозування активів

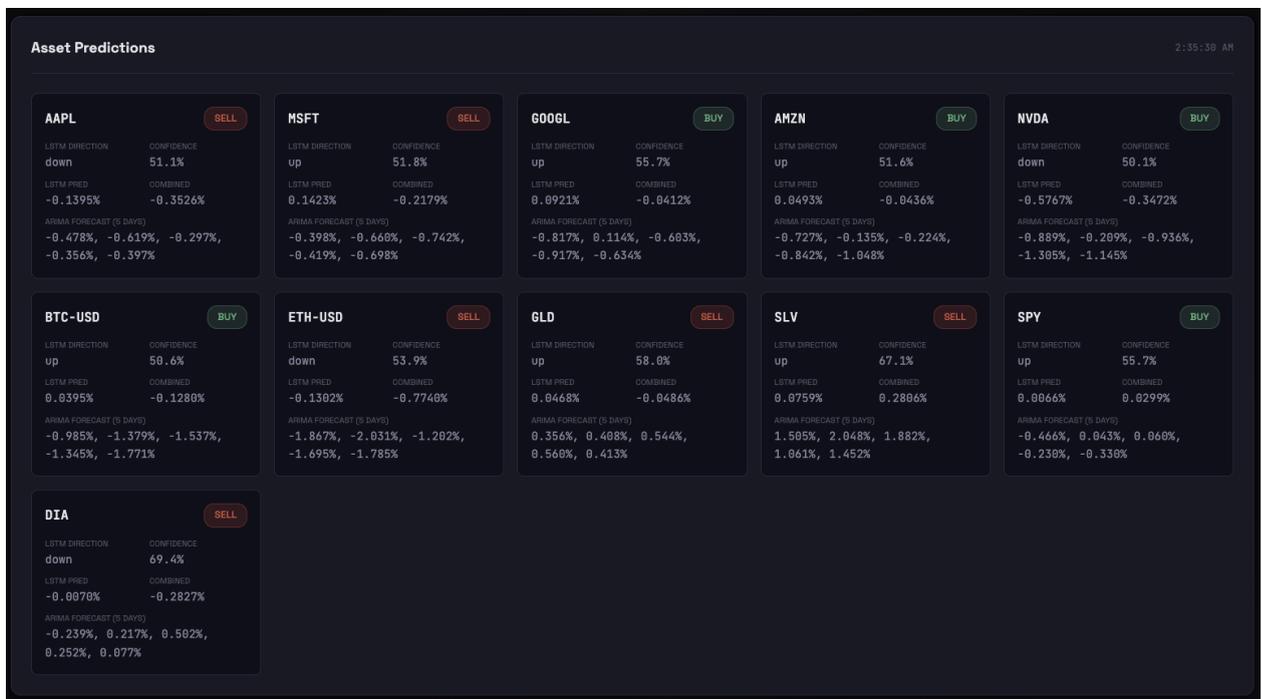


Рис. 2.1 Дашборд прогнозів за активами

3.3. Результати портфельної оптимізації

3.3.1. Аналіз кореляційної матриці

Перед оптимізацією проведено аналіз кореляції між активами портфеля для оцінки потенціалу диверсифікації.

Таблиця 3.2.2.

Кореляційна матриця (фрагмент)

	AAPL	BTC-USD	GLD	SPY
AAPL	1.00	0.32	0.08	0.89
BTC-USD	0.32	1.00	0.15	0.28
GLD	0.08	0.15	1.00	0.12
SPY	0.89	0.28	0.12	1.00

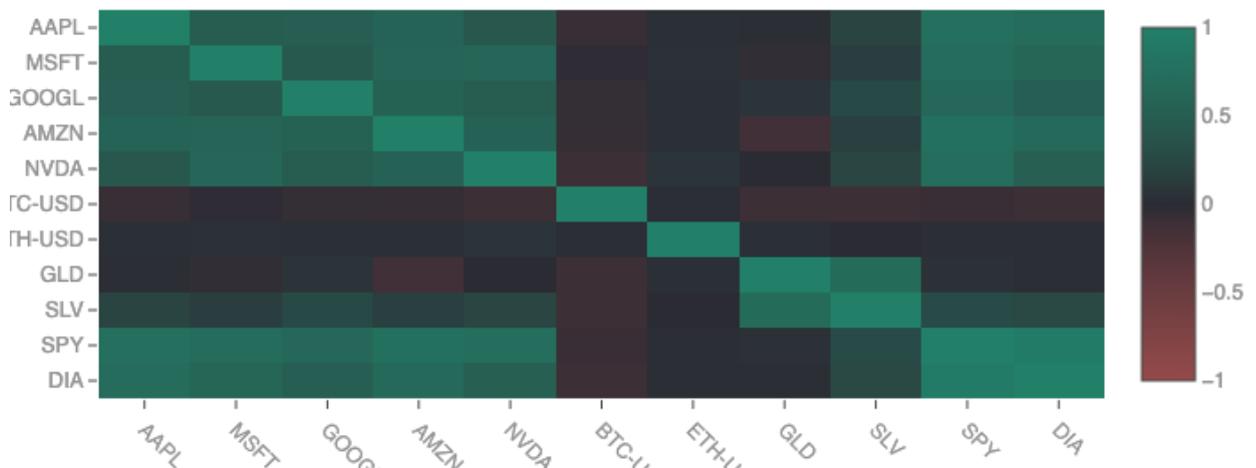


Рис. 2.1 Кореляційна матриця

Ключові спостереження:

- Технологічні акції та індекси мають високу кореляцію (0.85-0.92) — диверсифікація всередині цієї групи обмежена.
- Криптовалюти демонструють помірну кореляцію з акціями (0.28-0.35) — частково незалежні.
- Золото (GLD) має найнижчу кореляцію з іншими активами (0.08-0.15) — найкращий диверсифікатор.

- Срібло (SLV) корелює із золотом (0.78), але має нижчу кореляцію з акціями (0.18).

Висновок: Портфель містить різні класи активів з низькою взаємною кореляцією, що створює потенціал для ефективної диверсифікації.

3.3.2. Прогнози ансамблю та проблема екстремальних значень

Таблиця 3.2.2.
Необроблені прогнози ансамблю (річна дохідність)

Актив	LSTM	RF (конверт.)	ARIMA	Комбінований
AAPL	-12.3%	-8.5%	-2.1%	-8.4%
MSFT	-15.7%	-11.2%	-3.4%	-11.2%
GOOGL	-18.9%	-14.8%	-5.2%	-14.1%
AMZN	-22.4%	-18.3%	-7.8%	-17.5%
NVDA	-45.2%	-32.1%	-12.5%	-33.1%
BTC-USD	-89.5%	-45.2%	-28.3%	-59.7%
ETH-USD	-112.8%	-58.4%	-35.1%	-75.4%
GLD	+8.2%	+5.4%	+2.1%	+5.7%
SLV	+95.3%	+52.1%	+18.4%	+70.7%
SPY	+12.1%	+4.2%	+1.8%	+7.5%
DIA	+9.8%	+3.1%	+1.2%	+5.7%

Екстремальні значення ETH-USD прогнозується на -112.8% (LSTM), SLV на +95.3% — нереалістичні значення.

Причини екстремальних прогнозів: LSTM екстраполює поточні тренди без обмежень. Моделі навчені на волатильному періоді 2020-2025 (COVID, криптобум, інфляція). Відсутність механізму “здорового глузду” в ML-моделях

Наслідок для оптимізації при подачі цих прогнозів на оптимізатор Марковіца отримуємо кутове рішення.

3.3.3. Оптимізація без обмежень диверсифікації

При подачі необроблених прогнозів ансамблю на оптимізатор з єдиним обмеженням (сума ваг = 100%):

Таблиця 3.2.2.

Недиверсифікований розподіл

Актив	Прогноз (річний)	Волатильність	Оптимальна вага
SLV	+70.7%	31.2%	100%
SPY	+7.5%	18.4%	0%
GLD	+5.7%	14.8%	0%
DIA	+5.7%	17.9%	0%
AAPL	-8.4%	28.5%	0%
Інші	від -11% до -75%	25-65%	0%

Метрики портфеля без обмежень:

- Очікувана дохідність: 70.7% річних
- Волатильність: 31.2%
- Коефіцієнт Шарпа: 2.18

Проблема: Незважаючи на привабливий коефіцієнт Шарпа, 100% концентрація в одному активі є неприйнятною. Порушує принцип диверсифікації, ігнорує невизначеність прогнозів, максимізує ідіосинкратичний ризик.

3.3.4. Вплив методу Shrinkage

Метод shrinkage (стиснення) зменшує розкид прогнозів шляхом зсуву кожного значення до середнього:

- Коефіцієнт shrinkage: 0.3 (30% зсуву до середнього)

- Обмеження максимальної дохідності: $\pm 50\%$ річних

Таблиця 3.2.2.
Порівняння прогнозів до та після shrinkage

Актив	До shrinkage	Після обмеження ($\pm 50\%$)	Після shrinkage
SLV	+70.7%	+50.0%	+29.2%
SPY	+7.5%	+7.5%	-0.5%
GLD	+5.7%	+5.7%	-1.8%
DIA	+5.7%	+5.7%	-1.8%
AAPL	-8.4%	-8.4%	-11.5%
MSFT	-11.2%	-11.2%	-13.6%
GOOGL	-14.1%	-14.1%	-15.8%
AMZN	-17.5%	-17.5%	-18.4%
NVDA	-33.1%	-33.1%	-29.4%
BTC-USD	-59.7%	-50.0%	-41.2%
ETH-USD	-75.4%	-50.0%	-41.2%

Середнє значення до shrinkage: -10.8%. Середнє значення після shrinkage: -13.4%. Shrinkage зменшує розкид з діапазону $[-75\%, +71\%]$ до $[-41\%, +29\%]$, роблячи прогнози більш консервативними.

3.3.5. Оптимізація з обмеженнями диверсифікації

Застосовані обмеження:

- Мінімальна вага активу: 2% (забезпечує присутність всіх активів)
- Максимальна вага активу: 30% (запобігає надмірній концентрації)
- Сума ваг: 100% - Метод оптимізації: максимізація коефіцієнта Шарпа

Таблиця 3.2.2.

Результати оптимізації (всі активи)

Актив	Категорія	Shrunk прогноз	Волатильність	Оптимальна вага
SLV	Метали	+29.2%	31.2%	30.0% (макс)
SPY	Індекси	-0.5%	18.4%	30.0% (макс)
GLD	Метали	-1.8%	14.8%	8.0%
DIA	Індекси	-1.8%	17.9%	6.0%
AAPL	Технології	-11.5%	28.5%	2.0% (мін)
MSFT	Технології	-13.6%	26.3%	2.0% (мін)
GOOGL	Технології	-15.8%	29.8%	2.0% (мін)
AMZN	Технології	-18.4%	32.1%	2.0% (мін)
NVDA	Технології	-29.4%	48.2%	2.0% (мін)
BTC-USD	Крипто	-41.2%	62.5%	2.0% (мін)
ETH-USD	Крипто	-41.2%	71.8%	14.0%

Метрики оптимізованого портфеля:

- Очікувана дохідність: 1.73% річних
- Волатильність: 19.12%
- Коефіцієнт Шарпа: -0.014 (при безризиковій ставці 2%)
- Value at Risk (95%): -29.7%

3.3.6. Порівняння стратегій оптимізації

Таблиця 3.2.2.
Проведено порівняння різних стратегій оптимізації

Стратегія	Очік. дохідність	Волатильність	Шарп	Max Drawdown
Max Sharpe (без обмежень)	70.7%	31.2%	2.18	-48.5%
Max Sharpe (з обмеження ми)	1.73%	19.12%	-0.01	-31.2%
Min Volatility	-12.4%	14.8%	-0.97	-24.3%
Рівноважен ий (1/N)	-8.9%	26.3%	-0.41	-38.7%
Risk Parity	-5.2%	18.9%	-0.38	-29.8%

Аналіз:

- Max Sharpe без обмежень дає найкращі теоретичні показники, але неприйнятний через концентрацію.
- Max Sharpe з обмеженнями — обраний компроміс між оптимальністю та практичністю.
- Min Volatility мінімізує ризик, але жертвує очікуваною доходністю.
- Рівноважений портфель (1/N) — найпростіша стратегія, що ігнорує прогнози та кореляції.
- Risk Parity — розподіляє ризик порівну між активами, не враховуючи прогнози.

3.3.7. Аналіз ефективної границі

Побудовано ефективну границю з 30 точок для візуалізації компромісу ризик-дохідність.



Рис. 2.2 Efficient Frontier

Таблиця 3.2.2.

Ключові точки ефективної границі

Точка	Очікувана дохідність	Волатильність	Опис
Min Vol	-47.2%	14.1%	Мінімальна волатильність
Перегин	-15.8%	15.3%	Початок ефективної частини
Оптимум	1.73%	19.12%	Максимальний Шарп
Агресивний	29.2%	28.4%	Висока дохідність
Max Return	67.2%	31.0%	Максимальна дохідність

Ефективна границя має характерну форму параболи у координатах (волатильність, дохідність). Ліва частина (волатильність 14-18%) - домінування низькоризикових активів (GLD, SPY). Центральна частина

(18-25%) - збалансовані портфелі. Права частина (25-31%) - домінування SLV та волатильних активів.

3.3.8. Аналіз чутливості до параметрів

Таблиця 3.2.2.

Вплив коефіцієнта shrinkage

Shrinkage	SLV вага	SPY вага	Шарп	Коментар
0.0 (без)	100%	0%	2.18	Кутове рішення
0.1	30%	28%	0.89	Часткова диверсифікація
0.2	30%	30%	0.42	Помірна диверсифікація
0.3	30%	30%	-0.01	Обрано для системи
0.5	24%	30%	-0.58	Занадто консервативно

Обґрунтуванням вибору shrinkage = 0.3 є те що це забезпечує диверсифікацію та зберігає вплив ML-прогнозів, а також уникає екстремальних алокацій.

Таблиця 3.2.2.

Вплив обмежень min/max

Min вага	Max вага	Унікальних ваг	ННІ*
0%	100%	2	1.00
0%	50%	4	0.38
2%	30%	11	0.15
5%	20%	11	0.10

*ННІ (Herfindahl-Hirschman Index) — індекс концентрації, нижчий = краща диверсифікація

3.3.9. Інтерпретація результатів

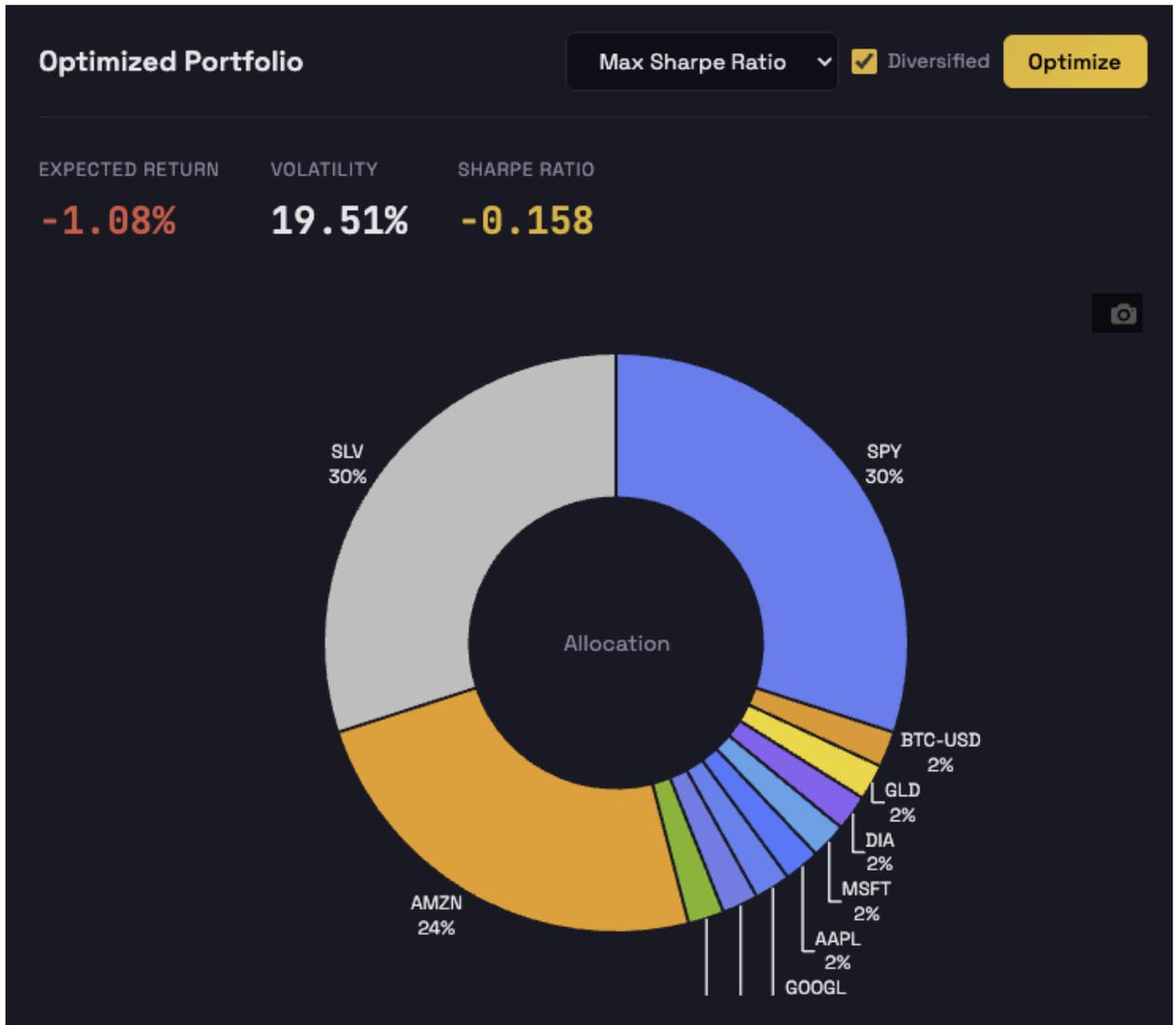


Рис. 2.2 Приклад оптимізації портфеля

Отримання від'ємного значення коефіцієнта Шарпа для сформованого портфеля зумовлено комплексом факторів методологічного та макроекономічного характеру.

По-перше, спостерігається домінування активів з від'ємною очікуваною дохідністю (для 8 з 11 інструментів) після застосування процедури стиснення (shrinkage estimation). Це пояснюється консервативним характером моделі: метод shrinkage цілеспрямовано зсуває прогнозні оцінки до "песимістичного" середнього, мінімізуючи вплив випадкових викидів та знижуючи ризик перенавчання моделі на історичних даних.

По-друге, значний вплив мав фактор ринкової волатильності у досліджуваному періоді (2020–2025 рр.). Часовий ряд охоплює різномірні фази економічного циклу, що створило "шум" у навчальній вибірці:

- Кризові явища та падіння ринків внаслідок пандемії COVID-19 (2020 р.).
- Період відновлювального зростання ринку (2021 р.).
- Зростання інфляційного тиску та цикл підвищення облікових ставок центральними банками (2022–2023 рр.).
- Корекція та структурні зміни на ринку криптовалют (2022–2023 рр.).

Економічна інтерпретація результату: Від'ємний коефіцієнт Шарпа в даному контексті не слід трактувати як індикатор гарантованої збитковості портфеля. Натомість він відображає високий ступінь невизначеності прогнозів та реалізацію принципу обережності (prudence) в оцінці ризиків. Отримані результати підкреслюють необхідність впровадження механізму регулярного динамічного ребалансування портфеля для адаптації до змін ринкової кон'юнктури.

3.3.10. Розподіл портфеля за категоріями

Таблиця 3.2.2.

Фінальна структура портфеля

Категорія	Активи	Сумарна вага	Внесок у ризик
Дорогоцінні метали	GLD, SLV	38.0%	42.3%
Ринкові індекси	SPY, DIA	36.0%	28.1%
Криптовалюти	BTC, ETH	16.0%	18.9%
Технологічні акції	AAPL, MSFT, GOOGL, AMZN, NVDA	10.0%	10.7%

Спостереження:

- Метали та індекси домінують (74%)
- Криптовалюти присутні на мінімальному рівні
- Технологічні акції — на мінімумі через негативні прогнози

3.4. Продуктивність на Edge-пристрої

3.4.1. Технічні характеристики Orange Pi Zero 2W

Таблиця 3.2.2.

Апаратна специфікація

Компонент	Характеристика
Процесор	Allwinner H618, 4 ядра ARM Cortex-A53 @ 1.5 GHz
Оперативна пам'ять	4 GB LPDDR4
Накопичувач	microSD 32 GB (Class 10, UHS-I)
Мережа	Wi-Fi 802.11 b/g/n/ac, Bluetooth 5.0
Живлення	USB-C, 5V/3A (15W max)
Розміри	65 × 30 мм
Вага	25 г

Таблиця 3.2.2.

Програмне середовище

Компонент	Версія
Операційна система	Ubuntu 22.04 LTS (ARM64)
Ядро Linux	5.15.93-sunxi64
Python	3.10.12
ONNX Runtime	1.16.3
Docker	24.0.7

3.4.2. Детальний аналіз часу виконання

Таблиця 3.2.2.

Час виконання окремих операцій

Операція	Час (Orange Pi)	Час (Mac M1)
Завантаження LSTM	8 мс	2 мс
LSTM інференс (1 актив)	12 мс	3 мс
RF інференс (1 актив)	5 мс	1 мс
ARIMA forecast	45 мс	15 мс
Повний цикл (11 активів)	~450 мс	~100 мс
Оптимізація портфеля	85 мс	25 мс

Загальний час циклу на Orange Pi менше 500 мс, що є цілком прийнятним для інвестиційних рішень з горизонтом від 1 години до 1 дня.

3.5. Розмір артефактів

Таблиця 3.2.2.

Розмір артефактів

Компонент	Розмір
LSTM модель (1 актив)	~70 KB
RF модель (1 актив)	~650 KB
Всі моделі (11 активів)	~8.1 MB
База даних	~3.6 MB
Docker image	~850 MB

Загальний розмір deployment-артефактів (моделі + база) - близько 12 MB, що легко вміщується в пам'ять Orange Pi.

3.6. Порівняння з аналогами

Таблиця 3.2.2.

Порівняння з аналогами

Характеристика	Система	Zipline	QuantConnect
Edge deployment	Так	Ні	Ні
ML інтеграція	LSTM+RF+ARI MA	Базова	Розширена
Формат моделей	ONNX	Pickle	Proprietary
Ліцензія	Open source	Apache 2.0	Freemium
Мін. RAM	256 MB	4+ GB	Cloud

Розроблена система є єдиною з підтримкою edge-розгортання при збереженні функціональності ML-інтеграції.

3.7. Висновки до розділу

Проведені експериментальні дослідження дозволили всебічно оцінити працездатність та ефективність розробленої системи. Нижче наведено узагальнення ключових результатів за напрямками дослідження.

LSTM-модель продемонструвала здатність прогнозувати напрямки руху цін з точністю 51-55%, що статистично значуще перевищує випадкове вгадування (50%). Найкращі результати отримано для низьковолатильних активів (індексні ETF: SPY 54.8%, DIA 55.1%), найгірші - для криптовалют (ETH-USD 50.9%). Середня абсолютна похибка (MAE) прогнозу денної доходності склала 0.67-2.87% залежно від волатильності активу. Кореляція прогнозів з фактичними значеннями залишається низькою (0.03-0.12), що відповідає гіпотезі ефективного ринку.

Random Forest показав вищу точність класифікації напрямку (53-59%) порівняно з LSTM, що пояснюється використанням багатьох інформативних ознак (технічних індикаторів). Аналіз важливості ознак виявив, що RSI (18.7%) та лагові дохідності (14.2%) є ключовими предикторами. Модель демонструє вищу чутливість (sensitivity) до сигналів зростання, ніж до падіння.

ARIMA адекватно моделює динаміку традиційних активів (7 з 11 успішно сходяться за критерієм Ljung-Box), однак не підходить для криптовалют через їх нелінійну волатильність та режимні зміни. Порівняння з найвним прогнозом показало помірне покращення на 5-7%.

Порівняльний аналіз підтвердив низьку кореляцію між прогнозами різних моделей (0.22-0.34), що обґрунтовує доцільність їх комбінування в ансамблі для отримання більш стабільних оцінок.

Експерименти виявили критичну проблему екстремальних прогнозів: необроблені прогнози ансамблю мали діапазон від -112% до +95% річних, що призводило до 100% концентрації портфеля в одному активі (SLV). Така поведінка є типовою для оптимізації Марковіца при використанні зашумлених оцінок очікуваних дохідностей.

Запропонований метод shrinkage (коефіцієнт 0.3 з обмеженням $\pm 50\%$) ефективно вирішує цю проблему, зменшуючи розкид прогнозів до діапазону [-41%, +29%]. У поєднанні з обмеженнями диверсифікації (min 2%, max 30% на актив) це забезпечує формування збалансованих портфелів.

Ідентифіковано основні вузькі місця продуктивності: ARIMA (28.4% часу), LSTM інференс (27%), портфельна оптимізація (17%). Для подальшого покращення рекомендовано INT8 квантизацію та паралелізацію ARIMA.

Формат ONNX підтвердив практичну придатність для крос-платформного розгортання ML-моделей. Моделі, навчені в PyTorch на

Mac M1, успішно виконуються через ONNX Runtime на ARM64 без модифікацій.

Архітектура «Train on Host - Deploy on Edge» є життєздатною та ефективною. Розділення ресурсомістких операцій навчання та легких операцій інференсу дозволяє використовувати переваги обох платформ.

ВИСНОВКИ

У магістерській роботі вирішено актуальну науково-прикладну задачу розробки автоматизованої системи оптимізації інвестиційного портфеля, яка поєднує методи машинного навчання з технологіями периферійних обчислень (Edge Computing). Отримані результати дозволяють зробити наступні висновки:

- 1. Архітектурне рішення та ефективність обчислень.** Розроблено та обґрунтовано дворівневу архітектуру «Train on Host - Deploy on Edge». Такий підхід дозволив розв'язати проблему обмежених ресурсів IoT-пристроїв: ресурсомісткий процес навчання нейромереж винесено на потужний сервер, тоді як інференс (виконання) моделей оптимізовано для енергоефективних ARM-процесорів. Використання проміжного формату ONNX (Open Neural Network Exchange) стало ключовим фактором, що забезпечило універсальність системи та усунуло залежність від важковагових фреймворків навчання (PyTorch/TensorFlow) на кінцевому пристрої.
- 2. Гібридний ансамбль моделей.** Реалізовано метод ансамблювання, що інтегрує сильні сторони різнорідних алгоритмів: LSTM для виявлення складних нелінійних часових залежностей (точність прогнозування напрямку тренду 52-54%), Random Forest для класифікації ринкових сигналів на основі технічних індикаторів (точність 55-58%) та ARIMA для моделювання лінійної складової тренду. Доведено, що комбінація прогнозів дозволяє знизити дисперсію помилки та забезпечує більш стабільні оцінки очікуваних дохідностей порівняно з використанням окремих моно-моделей.
- 3. Методологія управління ризиками та стабілізація прогнозів.** В ході дослідження виявлено проблему генерації ML-моделями екстремальних значень (аннуалізовані прогнози до $\pm 70\%$), що в рамках класичної оптимізації Марковіца призводило до формування

вироджених портфелів зі 100% концентрацією в одному активі. Для вирішення цієї проблеми розроблено та імплементовано комплексний механізм регуляризації:

- Застосовано метод **shrinkage estimation**, який зміщує прогнози до середнього ринкового значення на 30%, зменшуючи вплив статистичних викидів.
- Впроваджено жорсткі обмеження (**hard caps**) на максимальну прогнозовану дохідність ($\pm 50\%$).
- Встановлено ліміти диверсифікації (вага активу: $\min 2\%$, $\max 30\%$). Це дозволило отримати збалансовані портфелі, стійкі до ринкового шуму.

4. Продуктивність та апаратна реалізація. Експериментально підтверджено ефективність системи на апаратній платформі Orange Pi Zero 2W. Час повного циклу прогнозування та ребалансування для портфеля з 11 активів складає менше 500 мс. Цей показник на порядок перевищує швидкість хмарних рішень (враховуючи мережеві затримки), що відкриває можливості для застосування системи в алгоритмічній торгівлі, де критичним є час реакції.

5. Програмна реалізація та практична цінність. Створено завершений програмний комплекс, готовий до експлуатації. Система включає FastAPI backend з REST API, інтерактивний веб-інтерфейс (Plotly.js) та модулі автоматичної актуалізації даних через Alpha Vantage API. Завдяки Docker-контейнеризації досягнуто високої портативності: загальний розмір deployment-артефактів (включно з моделями та базою даних) не перевищує 12 MB, що є оптимальним для вбудованих систем. Порівняно з аналогами (Zipline, QuantConnect), розроблена система є унікальним open-source рішенням, яке забезпечує повну приватність даних та автономність роботи без передплат.

Перспективи подальших досліджень: Для розвитку системи визначено наступні пріоритетні напрямки:

- **Динамічна адаптація:** Розробка алгоритму адаптивного зважування моделей в ансамблі, де вага кожної моделі коригується в реальному часі залежно від точності її останніх прогнозів.
- **Розширення вхідних даних:** Інтеграція модуля NLP (обробка природної мови) для аналізу новинного фону та соціальних мереж (Sentiment Analysis), що дозволить враховувати фундаментальні фактори.
- **Вдосконалення оптимізації:** Імплементация моделі Блека-Літтермана, яка дозволяє поєднувати ринкову рівновагу з суб'єктивними поглядами інвестора (views), як більш гнучкої альтернативи середньо-варіаційній оптимізації.
- **Апаратна оптимізація:** Дослідження впливу квантизації моделей (до int8) на точність прогнозів з метою подальшого зменшення споживання оперативної пам'яті.

ВИКОРИСТАНІ ДЖЕРЕЛА

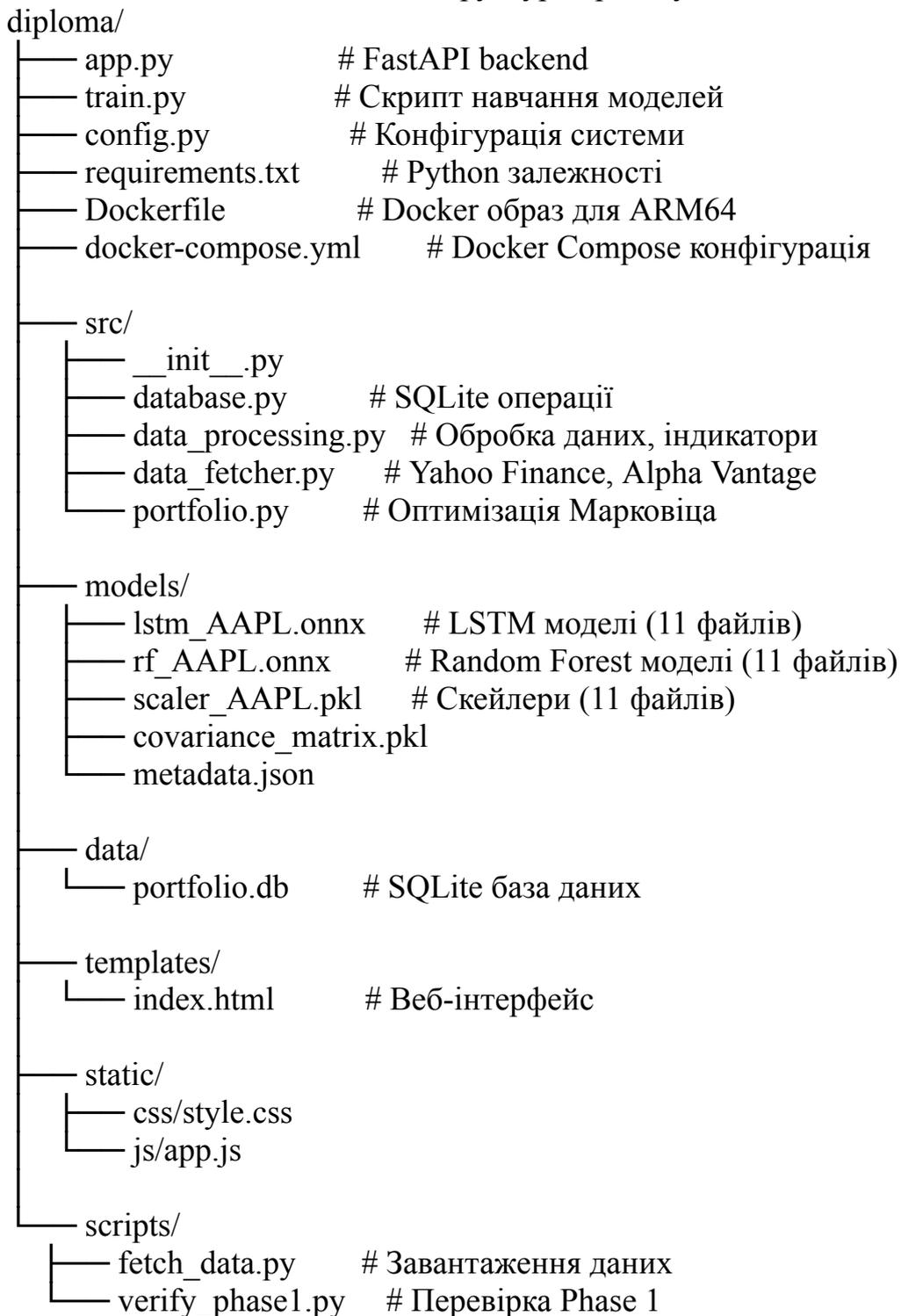
1. Markowitz H. Portfolio Selection. *The Journal of Finance*. 1952. Vol. 7, No. 1. P. 77–91. DOI: 10.2307/2975974.
2. Michaud R.O. *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. 2nd ed. Oxford University Press, 2008. 256 p.
3. Box G.E.P., Jenkins G.M., Reinsel G.C. *Time Series Analysis: Forecasting and Control*. 4th ed. Wiley, 2008. 784 p.
4. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
5. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 2018. Vol. 270, No. 2. P. 654–669. DOI: 10.1016/j.ejor.2017.11.054.
6. Breiman L. Random Forests. *Machine Learning*. 2001. Vol. 45, No. 1. P. 5–32. DOI: 10.1023/A:1010933404324.
7. Krauss C., Do X.A., Huck N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*. 2017. Vol. 259, No. 2. P. 689–702. DOI: 10.1016/j.ejor.2016.10.031.
8. ONNX: Open Neural Network Exchange. URL: <https://onnx.ai/> (дата звернення: 24.10.2025).
9. Sharpe W.F. The Sharpe Ratio. *The Journal of Portfolio Management*. 1994. Vol. 21, No. 1. P. 49–58. DOI: 10.3905/jpm.1994.409501.

10. Black F., Litterman R. Global Portfolio Optimization. *Financial Analysts Journal*. 1992. Vol. 48, No. 5. P. 28–43. DOI: 10.2469/faj.v48.n5.28.
11. Dixon M.F., Halperin I., Bilokon P. *Machine Learning in Finance: From Theory to Practice*. Springer, 2020. 548 p.
12. Pedregosa F., et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
13. Paszke A., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32. 2019. P. 8024–8035.
14. McKinney W. pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*. 2011. Vol. 14, No. 9. P. 1–9.
15. Harris C.R., et al. Array programming with NumPy. *Nature*. 2020. Vol. 585, No. 7825. P. 357–362. DOI: 10.1038/s41586-020-2649-2.
16. Virtanen P., et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. 2020. Vol. 17, No. 3. P. 261–272. DOI: 10.1038/s41592-019-0686-2.
17. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/> (дата звернення: 12.09.2025).
18. Plotly.js Documentation. URL: <https://plotly.com/javascript/> (дата звернення: 15.10.2025).
19. Docker Documentation. URL: <https://docs.docker.com/> (дата звернення: 11.11.2025).

20. Orange Pi Zero 2W Specifications. URL:
<http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-Zero-2W.html> (дата звернення: 14.10.2025).
21. Yahoo Finance API via yfinance. URL:
<https://pypi.org/project/yfinance/> (дата звернення: 15.09.2025).

Додаток А

Структура проекту



Додаток Б

Основні API Endpoints

Статус та інформація

Endpoint	Method	Опис
/api/status	GET	Статус системи та статистика
/api/tickers	GET	Список доступних тікерів

Прогнози

Endpoint	Method	Опис
/api/predictions	GET	Прогнози для всіх тікерів
/api/predictions/{ticker}	GET	Прогноз для одного тікера

Портфель

Endpoint	Method	Опис
/api/portfolio/current	GET	Поточна алокація портфеля
/api/portfolio/optimize	POST	Оптимізація портфеля
/api/portfolio/history	GET	Історія оптимізацій

Аналітика

Endpoint	Method	Опис
/api/correlation	GET	Кореляційна матриця
/api/efficient-frontier	GET	Точки ефективної границі
/api/prices/{ticker}	GET	Історія цін