

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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

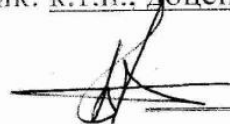
на тему:

Розробка програмного забезпечення автоматизованої системи керування
фондовим ділінгом

Виконав: студент 2 курсу, групи ЗАКІТР-23м
спеціальності 174 – Автоматизація та
комп'ютерно-інтегровані технології, та
робототехніка

 Ростислав ЧЕРВІНСЬКИЙ

Керівник: к.т.н., доцент кафедри АІТ

 Владислав КАБАЧІЙ

«06» 12 2024 р.

Рецензент: к.т.н., доцент кафедри КН

 Ігор ТРЦЕВЮК

«04» 12 2024 р.

Допущено до захисту

Зав. кафедри КСУ

 В'ячеслав Ковтун

«4» 12 2024 р.

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

ЗАТВЕРДЖУЮ

Зав. кафедри КСУ



В'ячеслав КОВТУН

“14” жовтня 2024 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Червінському Ростиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система керуванням фондовим ділінгом для трейдингового відділу банку

керівник роботи Кабачій Владислав Володимирович

затверджені наказом ВНТУ від “17” вересня 2024 року №310

2. Термін подання студентом роботи “1” грудня 2024 року

3. Вихідні дані до роботи:

1. Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. Yahoo Finance. URL: <https://finance.yahoo.com/>
2. Keras documentation: LSTM layer. Keras: Deep Learning for humans. URL: https://keras.io/api/layers/recurrent_layers/lstm/
3. pmdarima.arima.AutoARIMA – pmdarima 2.0.4 documentation. alkaline-ml - Distilling machine learning into its basics. URL: <https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html>
4. 1. Supervised learning. scikit-learn. URL: https://scikit-learn.org/stable/supervised_learning.html

4. Зміст текстової частини: Вступ; Аналіз сучасної теорії та практики автоматизованого керування фондовим ділінгом для трейдингового відділу банку; Аналіз задачі автоматизації керування фондовим ділінгом та методів її вирішення; Розробка програмного забезпечення автоматизованої системи керування фондовим ділінгом; Висновок; Список літератури.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Блок-схема програмного забезпечення, презентація на 13 слайдів, серед основних слайдів: “Титульний слайд”, “Мета і завдання роботи”, “Системи-аналоги”, “Дослідження прогнозування часових рядів”, “Структура програмного забезпечення”, “Тестування розробленого ПО”, “Висновки”.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв

2. Дата видачі завдання "14" жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
	Аналіз об'єкта автоматизації	19.09.2024	26.09.2024	
	Огляд систем-аналогів для керування	26.09.2024	01.10.2024	
	Огляд методів прогнозування часових рядів	01.10.2024	03.10.2024	
	Розробка блок-схеми програмного забезпечення	03.10.2024	10.10.2024	
	Дослідження поведінки методів	01.11.2024	14.11.2024	
	Розробка програмного забезпечення	14.11.2024	20.11.2024	
	Тестування програмного забезпечення	20.11.2024	26.11.2024	
	Розробка презентації	26.11.2024	26.11.2024	
	Оформлення пояснювальної записки	26.11.2024	28.11.2024	
	Захист МКР	11.12.2024		

Студент

Ростислав ЧЕРВІНСЬКИЙ

(підпис)

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

Владислав КАБАЧІЙ

(підпис)

(Ім'я ПРІЗВИЩЕ)

АНОТАЦІЯ

УДК 004.896:336.76

Червінський Р.О. Автоматизована система керуванням фондовим ділінгом для трейдингового відділу. Магістерська кваліфікаційна робота зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2024. 120 с.

На укр. мові. Бібліогр.: 39 назв; Рис.: 61; табл. 12.

Дана магістерська робота написана у відповідності із методичними вказівками до виконання магістерських кваліфікаційних робіт[38]. У даній магістерській кваліфікаційній роботі розроблено автоматизовану систему керування фондовим ділінгом. У оглядово-аналітичній частині було детально побудовано об'єкт дослідження, розглянуті конкуренти, локалізовано напрямок дослідження та оглянуті методи виконання поставленої задачі. У теоретично-методичній частині були досліджені особливості розглянутих у оглядово-аналітичній частині методів, найкращий був обраний для застосування у автоматизованій системі. У практичній частині було приведені результати розробки програмного забезпечення та його тестування. Ілюстративна частина складається із 13-ти плакатів із результатами роботи.

Ключові слова: автоматизована система, часові ряди, прогнозування, випадковий ліс, ARIMA, програмне забезпечення.

ABSTRACT

UDC 004.896:336.76

Chervinskyi R.O. Automated stock dealing management system for the trading department. Master's thesis on specialty 174 "Automation, computer-integrated technologies and robotics", educational program - Intelligent computer systems. Vinnytsia: VNTU, 2024. 120 p.

In Ukrainian speech Bibliography: 39 titles; Fig.: 61; table 12.

This master's thesis was written in accordance with the methodical instructions for completing master's qualifications works [38]. In this master's thesis, an automated stock dealing management system was developed. In the review and analytical part, the research object was constructed in detail, competitors were considered, the direction of the research was localized, and the methods of performing the task were reviewed. In the theoretical and methodological part, the peculiarities of the methods considered in the review and analytical part were investigated, the best one was selected for use in the automated system. In the practical part, the results of software development and testing were given. The illustrative part consists of 13 posters with work results.

Keywords: automated system, time series, forecasting, random forest, ARIMA, software.

ЗМІСТ

РОЗДІЛ 1 АНАЛІЗ СУЧАСТОЇ ТЕОРІЇ ТА ПРАКТИКИ АВТОМАТИЗОВАНОГО КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ ДЛЯ ТРЕЙДИНГОВОГО ВІДДІЛУ БАНКА	6
1.1 Аналіз об'єкта автоматизації	6
1.1.1 Аналіз проблеми фондового ділігу	6
1.1.2 Побудова об'єкта автоматизації	7
1.1.3 Структурний опис об'єкта	7
1.2 Огляд систем-аналогів для керування фондовим ділінгом	9
1.2.1 MetaTrader	9
1.2.2 Bloomberg Terminal	10
1.2.3 FIX Protocol	13
1.2.4 Порівняння систем-аналогів	14
1.3 Методи прогнозування часових рядів	15
1.3.1 K-Means Clustering	16
1.3.2 ARIMA	19
1.3.3 Древа рішень та Випадковий ліс	21
1.3.4 Штучні мережі	24
1.4 Висновки по розділу	28
РОЗДІЛ 2 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЗАДАЧІ АВТОМАТИЗОВАНОГО КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ	30
2.1 Інструменти для роботи, підготовка даних і формулювання цілей дослідження методів прогнозування	30
2.2 Дослідження методів прогнозування часових рядів	33
2.3 Порівняння методів та вибір найкращого для застосування у автоматизованій системі керування	42
2.4 Висновки по розділу	43
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ	44
3.1 Структура програмного забезпечення	44

3.2 Опис інтерфейсу.....	46
3.3 Тестування програмного забезпечення.....	47
ВИСНОВКИ.....	82
Список літератури	83
Додаток А (Обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ НА НАЯВНІСТЬ ЗАПОЗИЧЕНЬ	89
Додаток Б (Обов'язковий) ТЕХНІЧНЕ ЗАВДАННЯ	90
Додаток В (Вибірковий) ЛІСТИНГ ПРОГРАМИ ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	94
Додаток Г (Вибірковий) ЛІСТИНГ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ДІЛІНГОМ.....	97
Додаток Д (Обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА.....	109

ВСТУП

Актуальність теми. В умовах сучасних фінансових ринків, які характеризуються високою волатильністю, великими обсягами інформації та швидкою зміною ринкових умов, ефективне управління операціями з цінними паперами є критично важливим завданням для трейдингових відділів банків. Традиційні методи управління ділінговими операціями стають менш ефективними через необхідність швидкого реагування на ринкові зміни, обробки великих масивів даних та оптимізації ризиків.

Автоматизовані системи керування ділінгом сприяють оптимізації операцій, зменшенню операційних витрат, підвищенню продуктивності трейдерів та забезпеченню конкурентоспроможності банку в умовах цифрової трансформації фінансової галузі. Тому тема розробки та впровадження такої системи є надзвичайно актуальною для банків, що прагнуть залишатися лідерами на фондовому ринку.

Об'єктом дослідження є процес ділінгу фінансових інструментів.

Предметом дослідження є методи аналізу та прогнозування часових рядів.

Метою дослідження є розробка програмного забезпечення, яке буде реалізовувати автоматизовану систему керування ділінгом.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі**:

- дослідження об'єкту автоматизації.
- дослідження конкурентів, для визначення напрямку дослідження роботи у визначенні цінності фінансових інструментів.
- пошук та опис актуальних методів для визначення цінності фінансових інструментів.
- дослідження принципів роботи описаних актуальних методів.
- розробка та тестування програмного забезпечення.

Наукова новизна одержаних результатів:

- розробка системи керування фондовим ділінгом із функцією формування портфелю активів.

Практичною цінністю є розробка програми для прогнозування цінності фінансових інструментів.

Апробація результатів магістерської кваліфікаційної роботи. Матеріали заслуховувалися на конференції ВНТУ: “ЛІІІ Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)”[37].

Публікації результатів магістерської кваліфікаційної роботи. Матеріали опубліковувались тезами до конференції ВНТУ: “ЛІІІ Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)”.[37]

РОЗДІЛ 1 АНАЛІЗ СУЧАСТОЇ ТЕОРІЇ ТА ПРАКТИКИ АВТОМАТИЗОВАНОГО КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ ДЛЯ ТРЕЙДИНГОВОГО ВІДДІЛУ БАНКА

1.1 Аналіз об'єкта автоматизації

1.1.1 Аналіз проблеми фондового ділігу

Ділінг[39] — це процес купівлі та продажу фінансових інструментів, таких як валюта, акції, облігації або деривативи, від імені клієнтів або самого банку. Основною метою ділінгових операцій є отримання прибутку від змін цін цих активів на фінансових ринках.

Напрямок ділінгу, який називають фондовим ділінгом, зосереджений на операціях з акціями та іншими цінними паперами на фондових ринках.

Проведенням фондового ділінгу займається окремий відділ банку, який називають трейдинговим відділом.

У базовому вигляді, для успішного проведення фондового ділінгу потрібно продати фінансовий інструмент (акцію чи інший цінний папір[40]) за більшу ціну, ніж та за яку він був куплений. Для успішного проведення такої операції купівлі та продажу потрібно коректно спрогнозувати ціну на фінансовий інструмент у майбутньому.

Цінність фінансових інструментів залежить від багатьох різноманітних факторів, вплив яких частіше всього не очевидний, і навіть якісний аналіз яких не дає гарантії успішного прогнозу, тому розповсюдженою практикою є збір портфелю активів із різних фінансових інструментів. У тому випадку, якщо прогноз на один із фінансових інструментів не здійсниться, інші компенсують витрати на слабкий інструмент.

Підсумовуючи, процес фондового ділінгу можна розділити на такі етапи:

- Збір інформації про фінансові інструменти.
- Аналіз зібраної інформації.
- Прогноз цінності фінансових інструментів.

- Закупівля вигідних фінансових інструментів, формування з них портфелю активів.
- Продаж тих фінансових інструментів із портфеля, цінність яких зменшується згідно прогнозу.

Перші три розглянуті етапи вимагають великого обсягу одноманітної праці. Автоматизація проведення фондового ділінгу, наприклад здійснення розрахунків для прогнозування цінності фінансових інструментів за допомогою програмного забезпечення, призведе до позитивних результатів. Таким чином буде значно пришвидшена робота, оскільки вона буде виконуватися програмним забезпеченням, зменшена похибка у розрахунках від людського фактору, вивільняться людські ресурси, які застосовувалися для розрахунків.

1.1.2 Побудова об'єкта автоматизації

Об'єктом автоматизації у даній роботі є процес фондового ділінгу.

Для аналізу не використовувався ніякий конкретний об'єкт із власною технічною документацією. Об'єкт аналізу є типовим процесом, тому для його структурного та функціонального опису використовувалася науково-технічна література по фондовому ділінгу[39] та системи аналоги[1][3][7], як приклад.

Сам фондовий ділінг є процесом купівлі та продажі цінних паперів, такі як акції, облігації, інвестиційні сертифікати, тощо. Цінність таких паперів у моменті залежить від багатьох факторів, які часто змінюються із часом, через що змінюється і цінність паперів. Успішний фондовий ділінг передбачає купівлю цінних паперів перед ростом їх ціни, та продаж перед падінням їх ціни. Для поліпшення ефективності ділінгу у даній роботі розробляється автоматизована система керування, яка зможе прогнозувати цінність фінансових інструментів, та скласти портфель активів із оптимальним розподілом фінансових інструментів.

1.1.3 Структурний опис об'єкта

На рисунку (1) приведено структурну схему, об'єкта автоматизації.

Спочатку працівник банку (1) розпочинає роботу на своєму робочому місці (2). Працівник дистанційно зв'язується із клієнтами банку (3) для обговорення наданих послуг. Працівник використовує програмне забезпечення, робота якого ґрунтується на даних із підготовленої іншим відділом банку бази (4). Враховуючи поради програмного забезпечення, щодо купівлі продажу цінних паперів, працівник зв'язується із депозитарієм (7) та підготовлює до продажу цінні папери із портфелю активів банку (5) та/або клієнта (6). Далі працівник виконує продаж або купівлю цінних паперів на біржі (8), у випадку купівлі передає цінні папери на зберігання у депозитарій (7).

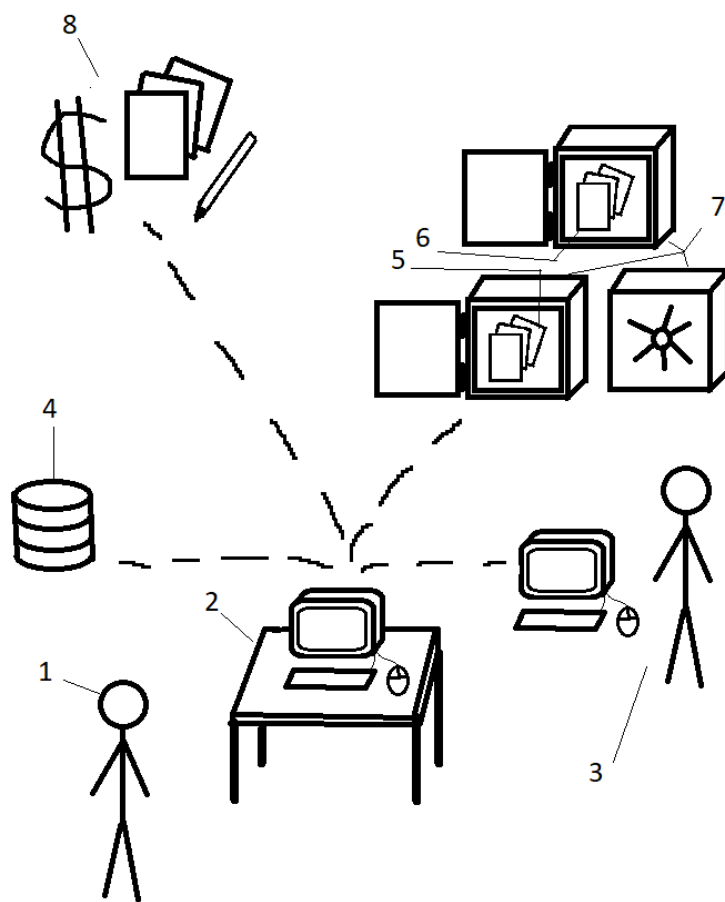


Рисунок 1. Структурна схема об'єкта автоматизації: 1 банківський працівник трейдингового відділу; 2 робоче місце працівника банку; 3 клієнт банку; 4 банківська база даних про характеристики цінних паперів; 5 портфель активів із цінними паперами банку; 6 портфель активів із цінними паперами клієнта; 7 депозитарій; 8 біржа.

Для даної роботи самим важливим елементом структурної схеми є робоче місце працівника банку, на якому буде встановлена розроблена у цій роботі система керування ділінгом.

1.2 Огляд систем-аналогів для керування фондовим ділінгом

1.2.1 MetaTrader

MetaTrader[1] це популярна торгова платформа, яка розроблена компанією MetaQuotes Ltd[2]. Ця компанія є провідним розробником програмного забезпечення для бірж, брокерів і банків.

MetaTrader має як версію для стаціонарних персональних комп'ютерів, так і версію для мобільних пристроїв, для смартфонів, планшетів, тощо. Таким чином користувачі MetaTrader можуть мати цілодобовий доступ до фінансових операцій незалежно від місцеположення користувача, поки у нього є доступ до інтернету.

MetaTrader дозволяє трейдерам здійснювати угоди купівлі та продажу активів (валюти, акції, ф'ючерси, метали та інші фінансові інструменти) у режимі реального часу.

MetaTrader має понад 30 вбудованих технічних індикаторів, таких як ковзні середні (Moving Averages), індекс відносної сили (RSI), MACD, стохастичний осцилятор та інші. Ці інструменти допомагають трейдерам аналізувати ринкові тенденції, визначати тренди, рівні підтримки та опору.

Платформа надає можливість малювати трендові лінії, канали, рівні Фібоначчі, хвилі Елліотта та інші графічні моделі для покращення аналізу ринкових рухів.

Трейдери можуть налаштовувати шаблони для аналізу та отримувати торгові сигнали на основі попередньо визначених умов.

Платформа дозволяє створювати автоматизовані торгові стратегії за допомогою експертних радників. Ці програми можуть виконувати торгові операції автоматично на основі заданих правил і умов.

Для написання експертних радників, індикаторів та скриптів MetaTrader використовує власну мову програмування MQL4 (для MT4) і MQL5 (для MT5). Це дає можливість створювати складні алгоритми для автоматизації торгівлі та аналітики.

Користувачі можуть програмувати торгових роботів, які автоматизують процес торгівлі, і проводити тестування стратегій на історичних даних для визначення їхньої ефективності.

MetaTrader (особливо MT5) включає доступ до економічного календаря з важливими новинами та подіями, які можуть вплинути на ринки (звіти по ВВП, безробіття, інфляція тощо).

Платформа надає поточні ринкові новини та аналітику, що допомагає трейдерам приймати рішення на основі фундаментальних факторів.

У підсумку, MetaTrader є потужною платформою для онлайн-торгівлі, яка поєднує інструменти для технічного та фундаментального аналізу, можливості автоматизації торгівлі через експертних радників і програмування власних стратегій. Проте безпосередньо MetaTrader не здійснює прогнозування цінності валют та цін на акції та інші цінні папери, лише надає користувачу інструменти для програмування власних прогнозуючих та технічні індекатори для аналізу. Платформа підходить як для початківців, так і для досвідчених трейдерів, дозволяючи гнучко управляти своїми активами та ефективно здійснювати операції на фінансових ринках.

1.2.2 Bloomberg Terminal

Bloomberg Terminal[3] — це провідна платформа для фінансових професіоналів, яка надає доступ до потужних інструментів для аналізу ринкових даних, проведення торгових операцій, отримання новин, і підтримує широкий спектр фінансових інструментів. Bloomberg Terminal використовується банками, інвесторами, трейдерами, аналітиками та корпораціями для прийняття рішень на фінансових ринках.

Bloomberg Terminal надає актуальні котирування для різних фінансових інструментів, таких як акції, облігації, валюти, деривативи та сировинні товари. Дані постійно оновлюються в реальному часі, що дає змогу користувачам оперативно реагувати на ринкові зміни. Платформа підтримує доступ до всіх основних ринків, включаючи фондові біржі, ринки облігацій, форекс, ринки товарів та енергоносіїв.

Платформа надає новини в реальному часі, висвітлюючи події в економіці, фінансах, політиці та бізнесі. Bloomberg є одним із провідних постачальників фінансових новин, що дозволяє користувачам бути в курсі останніх змін.

Bloomberg Terminal пропонує інструменти для аналізу ринкових тенденцій, оцінки економічних показників і відстеження корпоративних фінансів.

Bloomberg Terminal надає можливість безпосередньо здійснювати торгові операції з різними активами. Користувачі можуть купувати та продавати акції, облігації, валюти, деривативи та інші інструменти через платформу.

Платформа інтегрована з основними електронними торговими системами, що дозволяє користувачам виконувати операції на глобальних ринках безпосередньо з терміналу.

Bloomberg Terminal включає широкий набір фінансових інструментів для оцінки вартості активів, розрахунку дохідності облігацій, вартості деривативів і аналізу інвестиційних портфелів. Користувачі можуть аналізувати фінансову звітність компаній, оцінювати показники прибутковості, ефективність управління, і порівнювати фінансові дані компаній з їх конкурентами.

Bloomberg Terminal надає інструменти для створення інтерактивних графіків, що дозволяють аналізувати рух цін, обсяги торгівлі, технічні індикатори та тренди. Графіки можна налаштовувати за допомогою різних індикаторів і фільтрів.

Платформа дозволяє створювати діаграми з можливістю перегляду історичних даних, а також проєкцій майбутніх значень на основі фінансових моделей і аналітики.

Користувачі отримують доступ до докладної інформації про компанії: фінансову звітність, показники рентабельності, боргові зобов'язання, оцінки аналітиків та інші дані.

Платформа включає дані про макроекономічні показники, такі як ВВП, інфляція, безробіття, облікова ставка центральних банків, що допомагає аналізувати глобальні економічні тенденції.

Платформа надає інструменти для аналізу ризиків, хеджування та оптимізації портфелів відповідно до різних сценаріїв ринку.

Bloomberg Terminal включає інструменти для оцінки кредитного ризику емітентів облігацій, включаючи рейтинги агентств, спред облігацій, і ймовірність дефолту.

Платформа надає доступ до детальних даних про державні та корпоративні облігації, включаючи прибутковість, купонні платежі та графіки погашення.

Bloomberg Terminal надає інструменти для створення фінансових моделей і прогнозів на основі різних економічних і ринкових даних. Це дозволяє трейдерам і аналітикам будувати складні сценарії розвитку ринків і оцінювати потенційні інвестиції.

Bloomberg Terminal дозволяє створювати та використовувати алгоритми для автоматизованої торгівлі, зокрема на основі технічних індикаторів або ринкових подій. Це дає можливість торгувати швидко і ефективно без ручного втручання.

Bloomberg Terminal є платформою для фінансових професіоналів, яка забезпечує широкий доступ до ринкових даних, новин, торгових інструментів і фінансового аналізу в одному середовищі. На відміну від MetaTrader, інструменти якого більше представлені технічними індекаторами (Moving Averages[4], RSI[5], MACD[6]), Bloomberg Terminal більше зосереджений на фундаментальному аналізі економічного стану на ринках, фінансової звітності компаній та їх конкурентів, руху цін на товари та обсягу торгівлі ними, і все це з урахуванням макроекономічних показників. Завдяки широким можливостям для технічного та, управління портфелями, і автоматизації торгівлі, Bloomberg

Terminal вважається одним із найпотужніших інструментів для роботи на фінансових ринках.

1.2.3 FIX Protocol

FIX Protocol (Financial Information eXchange Protocol)[7] — це стандартний протокол для електронного обміну фінансовою інформацією між учасниками ринків, включаючи біржі, банки, брокерські компанії, інституційних інвесторів та інші фінансові установи. Він широко використовується для передавання даних у реальному часі між системами, що забезпечує швидку та надійну комунікацію в процесі торгівлі.

FIX Protocol був створений у 1992 році для обміну інформацією між трейдерами на фондових ринках. З часом він розвивався і став основним стандартом для електронної торгівлі в різних класах активів. Протокол продовжує розвиватися, додаючи нові функції та можливості для підтримки нових фінансових продуктів і технологій.

FIX використовується для обміну інформацією про торгові ордери між учасниками ринку (наприклад, брокерами та біржами). Сюди входять запити на купівлю/продаж активів, зміну або скасування ордерів, а також інформація про їх виконання.

Після укладення угоди між двома сторонами FIX Protocol передає дані для підтвердження виконання угоди, такі як кількість активів, ціна, дата виконання тощо.

Протокол використовується для відправки котирувань і запитів на котирування між брокерами, маркет-мейкерами та біржами. Це допомагає учасникам ринку бачити поточні ринкові ціни в реальному часі.

FIX може використовуватися для передавання даних про корпоративні дії (наприклад, дивіденди або спліти акцій), новини, що впливають на ринок, або зміну умов торгівлі.

FIX Protocol став міжнародним стандартом для передавання даних на фінансових ринках і широко використовується на фондових, валютних, товарних

і деривативних ринках у всьому світі. Його підтримують більшість бірж і великих фінансових установ.

Протокол є гнучким та модульним, що дозволяє його адаптувати до різних видів фінансових інструментів та учасників ринку. Він підтримує обмін даними для різних типів активів, таких як акції, облігації, деривативи, ф'ючерси та інші.

FIX дозволяє розширювати формат повідомлень для підтримки нових фінансових продуктів або специфічних потреб інституцій.

FIX працює за принципом обміну повідомленнями між двома сторонами (точками) — відправником і отримувачем. Повідомлення складаються з послідовності полів (назва і значення), де кожне поле має унікальний номер і певний формат.

Протокол включає безліч типів повідомлень, наприклад, повідомлення для розміщення ордерів, зміни або скасування ордерів, підтвердження угоди, запити на котирування тощо.

FIX підтримує безперервні сесії передачі даних між сторонами та забезпечує контроль цілісності даних через систему повідомлень для управління сесією.

FIX Protocol є важливим елементом електронної торгівлі на фінансових ринках, забезпечуючи швидкий, стандартизований і надійний обмін інформацією між різними учасниками ринку. Саме обмін інформацією протокол автоматизує безпосередньо, ніяких інструментів для проведення самої аналітики він не надає, тільки інформацію для аналізу. Протокол широко використовується для передавання даних про ордери, підтвердження угод, ринкові котирування та інші фінансові операції, забезпечуючи ефективність і прозорість у глобальній торгівлі.

1.2.4 Порівняння систем-аналогів

Усі розглянуті автоматизовані системи зосереджуються на автоматизації збору інформації, яка необхідна для проведення аналізу об'єкту торгівлі. Аналітика є значною частиною фондового ділінгу, адже допомагає зробити

коректний прогноз економічної доцільності купівлі/продажу цінних паперів у певний момент часу.

Проте розглянуті аналоги не дають інструментів для проведення самого прогнозу цінності цінних паперів, у кращому випадку вони надають допоміжні інструменти для самостійно прогнозування користувачем. Фактично, усі розглянуті системи автоматизують процес підготовки до прогнозування економічної доцільності фондового ділінгу.

У зв'язку із тим, що розповсюджені системи-аналоги мають виражену спеціалізацію, має сенс спрямувати дослідження цієї роботи на автоматизацію прогнозування цін на цінні папери, що є спільною слабкою стороною усіх розглянутих аналогів.

Отже, задачею даної магістерської кваліфікаційної роботи є розробка автоматизованої системи керуванням фондовим ділінгом, для трейдингового відділу банку. Виконання цієї задачі буде виконано через розробку програмного забезпечення, яке буде приймати інформацію по ряду фінансових інструментів, прогнозувати їх цінність та відбирати кілька найкращих, з яких можна сформувати пакет активів.

1.3 Методи прогнозування часових рядів

Прогнозування часових рядів[8] базуються на аналізі попередніх значень ціни, дозволяючи вловлювати закономірності, тренди та сезонні коливання. Ці моделі дозволяють спрогнозувати майбутні зміни ціни, спираючись на послідовність минулих даних, що є критично важливим для ринків із повторюваними патернами і дає можливість інвесторам і трейдерам приймати більш обґрунтовані рішення на ринку. Через що існує велике різноманіття літератури по прогнозуванню фінансових часових рядів[9][10][11].

Для розробки програмного забезпечення автоматизованої системи керування фондовим ділінгом, у даній роботі будуть використовуватися актуальні методи прогнозування часових рядів.

1.3.1 K-Means Clustering

K-Means Clustering[12] — це алгоритм некерованого навчання для кластеризації даних, який розподіляє точки даних у K груп (кластерів) на основі їх схожості. Основна мета — мінімізувати внутрішньокластерну дисперсію, тобто відстань між точками даних у кластері.

Ключові аспекти K-Means Clustering:

- Кластеризація. Мета K-Means полягає в тому, щоб поділити дані на K груп, де K визначається користувачем. Кожна група (кластер) має свій "центр" — центроїд, який представляє середнє значення точок в цьому кластері.
- Етапи алгоритму:
 - Ініціалізація: Вибираються K початкових центроїдів (випадковим чином або за допомогою спеціальних методів, як, наприклад, K-Means++ для кращого старту).
 - Призначення кластерів: Кожна точка даних призначається до найближчого центроїда на основі обраної метрики відстані (зазвичай це евклідова відстань).
 - Оновлення центроїдів: Після того, як всі точки призначені до кластерів, обчислюється новий центроїд для кожного кластера як середнє всіх точок у цьому кластері.
 - Повторення: Призначення точок і оновлення центроїдів повторюється, доки центроїди не перестануть змінюватися або не буде досягнуто максимальної кількості ітерацій.
- Кількість кластерів (K). Число кластерів K зазвичай вибирається за допомогою методів, таких як Elbow Method (Метод ліктя), який аналізує внутрішньокластерну дисперсію при різних значеннях K .
- Обмеження K-Means — алгоритм чутливий до початкового вибору центроїдів, що може призвести до локальних мінімумів. Не завжди добре працює, якщо кластери мають складні форми (наприклад, нерівномірний розмір або густину).

K-Means Clustering у контексті прогнозування часових рядів може бути використаний як метод для виявлення подібних структур або патернів у даних, що дозволяє покращити розуміння поведінки часового ряду та виявити схожі часові періоди для майбутнього прогнозування. Хоча **K-Means** не є прямим методом прогнозування, він може бути використаний для підготовки та попереднього аналізу даних, що сприяє покращенню прогнозів.

Кластеризація сегментів часових рядів. Часовий ряд можна розбити на сегменти (наприклад, по днях, тижнях, місяцях або інших періодах). Використовуючи **K-Means**, можна згрупувати схожі сегменти в кластери, де кожен кластер може представляти певний патерн поведінки (наприклад, підвищення або зниження попиту, сезонні цикли або аномалії). Це дає змогу виявити характерні патерни, які повторюються в різні періоди часу.

Виявлення режимів роботи або станів системи. Алгоритм K-Means може використовуватися для кластеризації даних часових рядів з метою виявлення різних режимів роботи системи або ринку. Наприклад, на фінансових ринках K-Means можна застосувати для кластеризації ринкових умов (тренд, бічний рух, волатильність тощо), що допомагає зрозуміти різні сценарії й адаптувати прогнозні моделі під ці умови.

Попередня обробка для подальших прогнозів. Після того як сегменти або стани часового ряду згруповані в кластери, можна будувати прогнози для кожного кластера окремо. Це дозволяє використовувати різні моделі для кожного патерну, покращуючи точність прогнозування. Наприклад, різні кластери можуть мати різні статистичні або стохастичні моделі, що краще підходять для кожного типу поведінки часового ряду.

Аналіз сезонності та трендів. K-Means можна використовувати для кластеризації даних на основі сезонних трендів. Це допомагає виявити подібні періоди в часі, наприклад, коли попит на продукт найвищий або коли на ринку спостерігається нестабільність. Такий підхід корисний для прогнозування короткострокових та довгострокових трендів.

Приклад застосування K-Means у прогнозуванні часових рядів:

- **Розбиття на сегменти.** Розбиття часового ряду на фіксовані інтервали (наприклад, щоденні або щомісячні інтервали).
- **Витяг ознак.** Для кожного сегмента можна витягти різні ознаки (наприклад, середнє значення, стандартне відхилення, тренди за певний період).
- **Кластеризація сегментів.** Застосування алгоритму **K-Means** до сегментів для кластеризації схожих патернів у даних.
- **Аналіз та прогнозування.** Для кожного кластера можна використовувати окрему модель для прогнозування подальшого розвитку часового ряду. На основі поточних даних можна визначити, до якого кластера належить новий період, і застосувати відповідну модель прогнозування.

Переваги використання K-Means для прогнозування часових рядів:

- **Виявлення структур та патернів** – K-Means дозволяє виявити приховані патерни та структури в даних, які не завжди очевидні при класичному аналізі часових рядів.
- **Адаптивні моделі** – розбиття часового ряду на кластери дає можливість застосовувати різні прогностичні моделі для кожного кластера, що може підвищити точність прогнозування.
- **Обробка сезонних і циклічних даних** – K-Means добре підходить для аналізу часових рядів з вираженою сезонністю або циклічністю, допомагаючи групувати подібні періоди й спрощуючи аналіз.

Недоліки:

- **Чутливість до вибору параметрів** – вибір кількості кластерів **K** може бути складним, і неправильний вибір може призвести до некоректних результатів кластеризації.
- **Не враховує послідовність часу** – алгоритм K-Means не враховує часову залежність між точками. Тому його потрібно використовувати в поєднанні з іншими методами, що враховують часову структуру даних (наприклад, ARIMA або LSTM).

K-Means Clustering є корисним інструментом для виявлення схожих патернів у часових рядах та сегментації даних. Він дозволяє покращити прогнози, адаптуючи моделі до специфічних груп даних. Хоча K-Means сам по собі не є методом для прогнозування часового ряду, його можна ефективно використовувати для попереднього аналізу та кластеризації даних перед прогнозуванням.

1.3.2 ARIMA

ARIMA (AutoRegressive Integrated Moving Average)[13] — це один із найпоширеніших методів для прогнозування часових рядів, який поєднує в собі авторегресію (AR), інтеграцію (I) та ковзне середнє (MA). Він використовується для моделювання даних, які демонструють певну залежність у часі, і особливо ефективний для прогнозування даних, що є стаціонарними або можуть бути приведені до стаціонарного стану.

Компоненти ARIMA:

- **AR (AutoRegressive)** — Авторегресійна частина[14].
- **I (Integrated)** — Інтегрована частина.
- **MA (Moving Average)** — Ковзне середнє.

Параметри ARIMA. Модель ARIMA описується трьома параметрами: (**p**, **d**, **q**), де:

- **p** — порядок авторегресії (кількість попередніх значень, які використовуються для прогнозування),
- **d** — порядок різницювання (кількість разів, коли застосовується операція різницювання для досягнення стаціонарності),
- **q** — порядок ковзного середнього (кількість попередніх помилок, які враховуються в моделі).

Модель AR показує, що поточне значення часового ряду є лінійною комбінацією попередніх значень із використанням **p** попередніх лагів.

Інтеграція означає, що модель працює з різницями між значеннями часового ряду для перетворення нестабільного (нестационарного) ряду на стаціонарний.

Наприклад, якщо ряд не стаціонарний, можна використовувати різницювання, щоб позбутися тренду. d вказує на кількість разів, коли була проведена операція різницювання. Якщо $d = 1$, то прогнозуються не самі значення, а різниці між поточними й попередніми значеннями.

Модель МА використовує залежність поточного значення від випадкових похибок (шуму) попередніх періодів. Вона враховує, як минулі значення випадкових похибок впливають на поточне значення.

Розглянемо принцип роботи ARIMA:

Виявлення стаціонарності: Перший крок — це перевірка, чи є ряд стаціонарним. Стаціонарний часовий ряд має постійні середнє значення та дисперсію, а також немає виражених трендів. Якщо ряд не стаціонарний, застосовується різницювання, щоб перетворити його на стаціонарний.

Визначення параметрів (p , d , q): Використовуються методи, як-от графіки автокореляційної функції (ACF) та часткової автокореляційної функції (PACF), для визначення оптимальних значень параметрів.

Побудова моделі: Після визначення параметрів будується модель ARIMA (p , d , q), яка враховує як авторегресію, так і ковзне середнє, щоб зробити точніший прогноз.

Оцінка моделі: Використовуються різні метрики для оцінки точності прогнозування, наприклад, середньоквадратична похибка (MSE)[15] або середня абсолютна похибка (MAE)[16].

Прогнозування: Після побудови та оцінки моделі вона використовується для прогнозування майбутніх значень часового ряду.

Приклад ARIMA:

Уявімо, що є часовий ряд цін на акції, і ми хочемо прогнозувати ціну на основі минулих значень. Припустимо, що після аналізу ряду було виявлено, що модель ARIMA(1, 1, 1) є найкращою для цього набору даних. Це означає, що:

- **$p = 1$:** використовується одне попереднє значення для авторегресії,
- **$d = 1$:** ряд різницьовано один раз для досягнення стаціонарності,
- **$q = 1$:** використовується одне попереднє значення помилки для моделювання ковзного середнього.

Переваги ARIMA:

- **Гнучкість:** ARIMA добре працює з різними типами даних, які можна привести до стаціонарного вигляду.
- **Можливість комбінування компонентів:** Модель поєднує три компоненти (AR, I, MA), що дозволяє враховувати різні типи залежностей у даних.
- **Простота інтерпретації:** Параметри ARIMA мають чітку математичну інтерпретацію, що полегшує розуміння моделі.

Недоліки ARIMA:

- **Нелінійні залежності:** ARIMA передбачає лінійні відносини між значеннями ряду, що може бути обмеженням у випадку складних, нелінійних залежностей.
- **Не враховує зовнішні фактори:** Стандартна ARIMA не враховує зовнішні (екзогенні) змінні, які можуть впливати на ряд. Для цього використовують моделі типу **ARIMAX** (ARIMA з екзогенними змінними).

ARIMA — це потужний і широко використовуваний метод для прогнозування часових рядів, який підходить для стаціонарних даних або даних, які можна привести до стаціонарного стану. Метод поєднує в собі авторегресію, інтеграцію та ковзне середнє, що дозволяє враховувати різні аспекти часових залежностей у даних. Попри свої обмеження, ARIMA залишається одним із найпопулярніших інструментів для аналізу та прогнозування часових рядів.

1.3.3 Древа рішень та Випадковий ліс

Древа рішень (Decision Trees)[17] та **Випадковий ліс (Random Forest)**[18] — це методи машинного навчання, які широко застосовуються для вирішення регресійних та класифікаційних завдань.

У контексті прогнозування часових рядів ці методи використовуються для побудови моделей, що можуть передбачати майбутні значення ряду на основі попередніх спостережень. Хоча ці моделі не є стандартними для роботи з часовими рядами, вони можуть бути адаптовані для цієї мети.

Проблема прогнозування часових рядів: Для використання дерев рішень або випадкового лісу з часовими рядами, важливо перетворити послідовні дані на формат, який модель зможе обробляти. Це зазвичай включає перетворення часового ряду на багатовимірні дані шляхом створення лагових змінних.

Створення лагових змінних це основний підхід для адаптації дерев рішень і випадкового лісу до часових рядів. Лагові змінні дозволяють моделі "бачити" попередні значення ряду й робити прогнози на їх основі.

Адаптація до складних патернів. Випадковий ліс є особливо корисним у випадках, коли часові ряди мають складні, нелінійні залежності або шуми, які можуть бути важко вловлювати за допомогою традиційних методів.

Переваги використання:

- Моделі добре працюють із нелінійними залежностями, що робить їх гнучкими для різних типів даних.
- Можливість адаптуватися до складних патернів у даних.
- Відсутність вимоги до стаціонарності даних.
- Випадковий ліс забезпечує надійні прогнози за рахунок ансамблевого підходу.

Недоліки:

- Вимагають перетворення даних у формат із лагами, що може призводити до втрати інформації про час як безперервну змінну.
- Не враховують прямі тимчасові залежності, як це роблять спеціалізовані моделі, наприклад, ARIMA чи LSTM.

Дерево рішень — це модель, яка розбиває дані на підмножини на основі значень вхідних змінних, створюючи послідовні розгалуження у вигляді дерева. Кожне розгалуження ґрунтується на розділенні вхідних даних за певним

критерієм, наприклад, на основі мінімізації середньоквадратичної похибки (для регресії) або ентропії (для класифікації).

У прогнозуванні часових рядів дерево рішень можна використовувати як регресійну модель, де кожне значення ряду (напр., значення за останні декілька днів або місяців) може виступати входними змінними для передбачення наступного значення.

Лагові змінні. Для роботи з часовими рядами у дерева рішень використовуються лагові змінні — це значення ряду на попередніх часових кроках, які виступають як входні дані для моделі.

Наприклад, якщо потрібно передбачити значення за $t+1$, дерево рішень може використовувати значення за t , $t-1$, $t-2$ як змінні.

Розбиття даних. На кожному вузлі дерева відбувається розбиття даних на основі найбільш значущих лагових змінних. Це розбиття дозволяє моделі навчитися закономірностям, що існують у часовому ряді.

Переваги:

- **Інтерпретованість** — дерева рішень легко інтерпретувати, що дозволяє зрозуміти, як модель робить прогнози.
- **Відсутність вимог до стаціонарності** — модель дерева рішень не вимагає, щоб ряд був стаціонарним, як у випадку з моделями типу ARIMA.

Недоліки:

- **Перенавчання (overfitting)** — дерева рішень можуть легко перенавчатися, якщо не застосовувати механізм обрізки (pruning).
- **Лінійні часові залежності** — дерева рішень можуть неефективно вловлювати довготривалі лінійні залежності в часових рядах.

Випадковий ліс — це ансамблева модель, що складається з багатьох незалежних дерев рішень, побудованих на різних випадкових підмножинах даних та ознак. Кожне дерево робить своє передбачення, а кінцевий результат виходить шляхом усереднення результатів всіх дерев (для регресії) або голосування (для класифікації).

Використання лагів: Як і в дереві рішень, для часових рядів випадковий ліс використовує лагові змінні. Можна, наприклад, вибрати певну кількість попередніх часових кроків ($t-1$, $t-2$, ... $t-n$) як вхідні змінні для кожного дерева.

Підвибірки даних та ознак: Для кожного дерева у випадковому лісі вибираються випадкові підвибірки даних та змінних, що дозволяє моделі краще враховувати різні аспекти часового ряду і зменшує ризик перенавчання.

Переваги:

- **Зменшення перенавчання** – використання багатьох дерев знижує ризик перенавчання в порівнянні з окремим деревом рішень.
- **Гнучкість** – випадковий ліс може працювати з великими набором лагових змінних та добре справлятися з нелінійними взаємозв'язками в даних.
- **Висока точність** – модель випадкового лісу зазвичай забезпечує вищу точність прогнозування завдяки ансамблевій природі.

Недоліки:

- **Складність інтерпретації** – на відміну від окремого дерева рішень, результат випадкового лісу складніше інтерпретувати, оскільки він є результатом ансамблю багатьох дерев.
- **Високі обчислювальні витрати** – через велику кількість дерев, випадковий ліс є більш ресурсоємною моделлю в порівнянні з окремим деревом рішень.

Хоча дерева рішень та випадковий ліс не є класичними підходами для прогнозування часових рядів, вони можуть бути адаптовані для цієї задачі за допомогою створення лагових змінних. Випадковий ліс, завдяки своїй ансамблевій природі, часто забезпечує високу точність і стійкість у випадках, коли дані мають складні патерни або шуми.

1.3.4 Штучні мережі

Прогнозування часових рядів за допомогою нейронних мереж — це метод, що використовує штучні нейронні мережі для аналізу й прогнозування майбутніх значень на основі історичних даних.

Далі приведені основні типи нейронних мереж для прогнозування часових рядів.

Feedforward Neural Networks (FFNN)[19]. Це найпростіший тип нейронних мереж, де інформація рухається в одному напрямку — від входів до виходів. Для прогнозування часових рядів, FFNN використовують вікно даних із минулих значень (фіксовану кількість попередніх спостережень) для прогнозування наступного значення. Проте цей метод має обмеження, оскільки він не враховує часову залежність між спостереженнями.

Рекурентні нейронні мережі (RNN)[20]. RNN мають пам'ять, яка дозволяє їм враховувати часову залежність між даними. Вони добре підходять для обробки послідовних даних, оскільки зберігають інформацію про попередні етапи у своїх прихованих станах. Однак звичайні RNN мають проблему зникнення градієнта, що ускладнює прогнозування для довгих часових рядів.

Long Short-Term Memory (LSTM)[21]. LSTM — це спеціальний тип рекурентних нейронних мереж, створений для вирішення проблеми зникнення градієнта. LSTM можуть зберігати інформацію на довші проміжки часу і краще підходять для складних часових рядів, де є довгострокові залежності між спостереженнями. LSTM широко використовуються для прогнозування фінансових ринків, енергетики, метеорології тощо.

Прогнозування часових рядів за допомогою нейронних мереж можна розділити на 4 етапи:

- Підготовка даних;
- Навчання моделі;
- Прогнозування;
- Оцінка моделі.

Підготовка даних:

Дані часових рядів повинні бути належним чином підготовлені. Спочатку вони можуть бути нормалізовані або масштабовані для кращої обробки нейронною мережею.

Часовий ряд часто розбивається на окремі вікна, де кожне вікно містить декілька попередніх значень часового ряду, які використовуються для прогнозування наступного значення.

Навчання моделі:

Під час навчання нейронна мережа мінімізує похибку між прогнозованими і реальними значеннями. Зазвичай використовуються функції втрат, такі як середньоквадратична похибка (MSE)[15].

У процесі навчання використовуються методи оптимізації, такі як стохастичний градієнтний спуск (SGD)[22] або його вдосконалені варіанти, щоб коригувати ваги мережі для точного прогнозування.

Прогнозування:

Після навчання мережі на історичних даних вона може використовуватися для прогнозування майбутніх значень часового ряду. Модель намагається виявити приховані закономірності й тренди, щоб створити точні прогнози.

Оцінка моделі:

Після навчання моделі її точність оцінюється за допомогою тестових даних, що не використовувалися в процесі навчання. Метрики, як-от середньоквадратична похибка або абсолютна помилка, дозволяють оцінити, наскільки добре модель виконує прогнозування.

Переваги прогнозування часових рядів за допомогою нейронних мереж:

- Виявлення складних залежностей: Нейронні мережі можуть виявляти нелінійні залежності та складні патерни в даних, що традиційні статистичні методи, такі як ARIMA, можуть пропустити.
- Обробка великої кількості даних: Нейронні мережі здатні обробляти великі обсяги даних та враховувати тривалі часові залежності.
- Адаптивність: Нейронні мережі можуть адаптуватися до змін у часових рядах, що важливо для нестабільних даних, таких як фінансові ринки.

Недоліки:

- Необхідність великої кількості даних: Нейронні мережі потребують великого набору даних для ефективного навчання.

- **Ресурсоемність:** Навчання складних моделей, таких як LSTM або GRU, вимагає значних обчислювальних ресурсів.
- **Налаштування гіперпараметрів:** Налаштування мережі, вибір оптимальної структури та кількості нейронів може бути складним процесом, що вимагає експериментів і досвіду.

Нейронні мережі є потужним інструментом для прогнозування часових рядів, особливо коли дані мають складні нелінійні залежності. Моделі, такі як LSTM добре підходять для обробки послідовних даних, що дає можливість створювати більш точні прогнози порівняно з традиційними статистичними методами. Тому розглянемо LSTM більш детально.

Long Short-Term Memory (LSTM) — це спеціальний тип рекурентної нейронної мережі (RNN), розроблений для обробки послідовних даних і подолання проблеми "зникнення градієнта", яка заважає звичайним RNN ефективно запам'ятовувати довгострокові залежності в часових рядах. LSTM добре підходять для завдань, де важливо враховувати часову послідовність, наприклад, у прогнозуванні часових рядів, розпізнаванні мовлення та обробці природної мови.

Основні компоненти LSTM:

- **Осередок (cell state):** Це ключовий компонент LSTM, який може зберігати інформацію на довгий проміжок часу. Осередок дозволяє зберігати й передавати інформацію через багато часових кроків, з мінімальними змінами.
- **Забутні (forget) ворота:** Визначає, яку частину інформації з попередніх етапів слід забути. Забутні ворота використовують функцію сигмоїди для обчислення значення між 0 і 1 для кожного елемента у стані осередку. "0" означає, що інформацію слід повністю забути, а "1" — зберегти.
- **Ворота запису (input gate):** Ворота запису визначають, яку нову інформацію слід додати до осередку. Тут використовуються дві операції: сигмоїда вирішує, які значення оновлювати, а тангенс визначає можливі нові значення для додавання до стану.

- **Ворота виходу (output gate):** Ворота виходу контролюють, яка частина інформації з осередку передається як вихід. Використовується сигмоїд для визначення, що передавати, та гіперболічний тангенс для отримання кінцевого вихідного значення.

Принцип роботи:

LSTM зберігає інформацію через декілька часових кроків, що дозволяє йому обробляти тривалі часові ряди. На кожному кроці часу три згадані ворота (forget, input і output) керують потоком інформації. Вони вирішують, яку інформацію оновлювати, зберігати чи передавати далі:

- **Забування непотрібної інформації:** Визначає, яку частину попередньої інформації потрібно видалити.
- **Додавання нової інформації:** Ворота запису оновлюють стан осередку, додаючи нову інформацію, важливу для поточного моменту часу.
- **Формування виходу:** Ворота виходу обирають, що слід передати як вихідний сигнал до наступного шару або часової точки.

Чому LSTM краще за звичайні RNN:

- **Робота з довгостроковими залежностями:** LSTM має здатність зберігати та використовувати інформацію протягом тривалих періодів часу, на відміну від стандартних RNN, які зазвичай забувають ранні дані.
- **Подолання проблеми зникнення градієнта:** Завдяки своїй архітектурі, LSTM може ефективно навчатися на послідовностях, де є довгострокові залежності, тоді як RNN зазвичай "забувають" важливу інформацію через згасання градієнтів під час навчання.

1.4 Висновки по розділу

Даний розділ був присвячений трьом поставленим у вступі завданням, це дослідження об'єкту автоматизації, дослідження конкурентів, для визначення напрямку дослідження роботи у визначенні цінності фінансових інструментів та

пошук та опис актуальних методів для визначення цінності фінансових інструментів.

У рамках дослідження об'єкту автоматизації було проаналізовано проблему фондового ділінгу, зіставлено детальний опис об'єкта автоматизації та розроблено схему, яка описує його структуру.

При дослідженні конкурентів були проаналізовані системи-аналоги. Результатом аналізу став висновок – конкурентні системи зосереджені на інформуванні користувача, нехтуючи безпосереднім прогнозом ціни на фінансові інструменти. Тому дослідження даної роботи буде зосереджено на прогнозуванні цін фінансових інструментів. Для проведення прогнозування були обрані методи прогнозування часових рядів.

Також у даному розділі було описано актуальні методи машинного навчання, які можуть застосовуватися для прогнозування часових рядів.

РОЗДІЛ 2 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЗАДАЧІ АВТОМАТИЗОВАНОГО КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ

2.1 Інструменти для роботи, підготовка даних і формулювання цілей дослідження методів прогнозування

Розробка моделей для прогнозування часових рядів та автоматизованої системи керування ділінгом у даній роботі виконувалася у середовищі програмування Microsoft Visual Studio на мові програмування Python. Середовище для розробки було обрано через зручність та звичність для розробника, а мова програмування за наявності широкого вибору бібліотек, які зручні для проведення машинного навчання та прості в опануванні.

Для виконання прогнозування у даній роботі використовувалися дані, які зберігаються у форматі “.hst”. Такі файли зберігають історичні дані по цінам (відкриваючій, найвищій, найнижчій та закриваючій) на фінансові інструменти та об’єму його торгівлі. Для того щоб Python міг працювати із файлами формату “.hst” потрібно встановити[23] публічний модуль “mt4forexparser”[24], він дозволяє зчитувати дані із файлів розширення “.hst” та формувати із них змінну формату “dataframe”[25], для роботи з яким мені знадобився мануал Олега Бондаренко[26]. Змінна “dataframe”, яка отримується у результаті зчитування “.hst” файла, містить відповідну кількість записів(залежно від частоти фіксації ціни це може бути щогодинний, щоденний, щомісячний графік тощо) кількість рядків, та сім стовпчиків, у яких зберігається дата запису, час запису, значення відкриваючої торги ціни, найвище значення ціни за час спостереження, найнижче значення ціни за час спостереження, значення закриваючої торги ціни, та загальна кількість акцій, які були обміняні за період спостережень.

Далі, на рисунках (2,3) наведено приклад результату зчитування даних, та побудованого на їх основі графіку часового ряду.

	date	time	open	high	low	close	volume
0	2016.10.18	00:00	149.33	150.91	147.73	150.64	13479
1	2016.10.19	00:00	151.20	152.37	150.76	151.20	7257
2	2016.10.20	00:00	151.06	152.83	150.85	151.60	10032
3	2016.10.21	00:00	150.38	151.03	149.50	149.62	6881
4	2016.10.24	00:00	150.44	151.43	150.35	150.58	4921
...	...	...	...	...	...	...	...
2042	2024.11.29	00:00	227.59	230.26	226.97	227.29	7009
2043	2024.12.02	00:00	227.10	228.29	225.10	227.52	4851
2044	2024.12.03	00:00	226.89	228.68	226.54	228.67	3892
2045	2024.12.04	00:00	229.54	233.56	229.18	233.54	5342
2046	2024.12.05	00:00	233.70	236.44	233.45	234.81	5310

Рисунок 2. Приклад структури даних часового ряду.

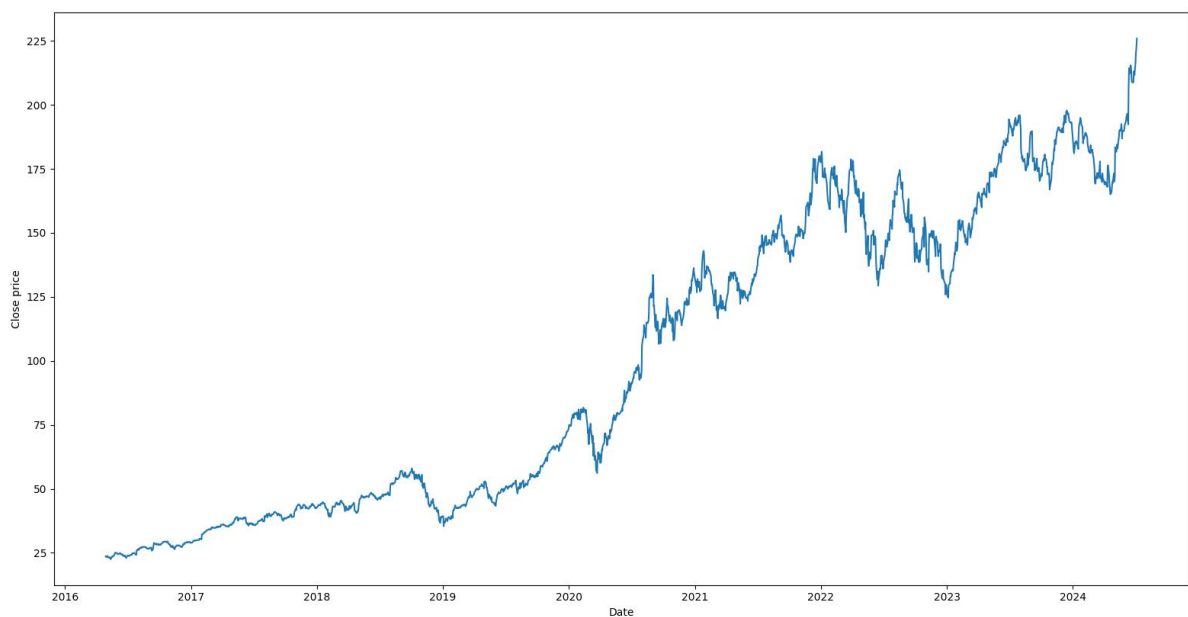


Рисунок 3. Графік історії змін закриваючої ціни з часом.

Фактично – задача прогнозування ціни на фінансові інструменти є задачею регресії, проте вона має кілька важливих особливостей. Перед тим як переходити до навчання моделі прогнозування у регресійній задачі проводиться випадкове перемішування даних та їх розподіл на навчальну та тестову вибірки, головною особливістю яких повинна бути репрезентативність. У випадку прогнозування часових рядів, проводити цей етап у такому вигляді буде помилковим рішенням. Ціна фінансових інструментів росте або падає з часом під дією зовнішніх факторів, тому коректне навчання моделі вимагає використання послідовних навчальних даних, від самих давніх до самих свіжих. Тому розподіл даних на навчальну і тестову вибірки для перевірки роботи моделей прогнозування

часових рядів повинні проводитися по деякій даті, раніше якої всі значення входять до навчальної вибірки а всі подальші до тестової.

У даній роботі за цільову змінну розглядається значення ціни фінансового інструменту на кінець торгів за день. Головною ідеєю прогнозування майбутньої ціни є визначення ціни close фінансового інструменту на деякий момент часу у майбутньому, на основі використання даних за деякий момент часу у минулому. У даній роботі буде реалізовуватися прогнозування ціни на тиждень вперед, тому стовпчик даних цільової змінної буде виглядати як стовпчик даних close зміщений на сім позицій. Приклад наведено далі на рисунку (4).

	date	time	open	high	low	close	volume	Target
0	2016.10.18	00:00	149.33	150.91	147.73	150.64	13479	153.25
1	2016.10.19	00:00	151.20	152.37	150.76	151.20	7257	152.41
2	2016.10.20	00:00	151.06	152.83	150.85	151.60	10032	153.6
3	2016.10.21	00:00	150.38	151.03	149.50	149.62	6881	152.82
4	2016.10.24	00:00	150.44	151.43	150.35	150.58	4921	151.74
5	2016.10.25	00:00	150.24	151.08	149.74	150.77	5732	152.16
6	2016.10.26	00:00	150.62	152.87	150.10	151.67	7834	152.39
7	2016.10.27	00:00	152.77	153.97	151.94	153.25	9366	155.61

Рисунок 4. Структура вхідних та вихідного даних.

Сформулюємо більш ґрунтовно цілі, які потрібно досягнути за допомогою методів прогнозування часових рядів.

Зверніть увагу на дані з рисунку (4), припустимо що поточна дата та час це “2016-10-18” і “00:00”. У кінцевому результаті у нас повинна вийти модель, яка навчається на всіх даних до дати “2024-10-18”, а данні цієї дати використовує для прогнозування ціни close на “2024-10-27”. Таким чином, доки час не відійшов від “00:00”, і поточна ціна не відхилилася від останнього close “150.64”, у нас є можливість прийняти рішення щодо доцільності вкладання у фінансовий інструмент, який ми розглядаємо, із наміром здобути прибуток з його продажу через тиждень.

Для оцінки побудованих моделей використовувалися наступні ознаки:

- Середня абсолютна похибка (MAE): Вимірює середню абсолютну величину помилки.

- Середньоквадратична похибка (MSE): Покарання за великі помилки сильніше, ніж у MAE.
- Корінь середньоквадратичної похибки (RMSE): Зручний для інтерпретації, оскільки має ті ж одиниці виміру, що й початкові дані.
- Середня абсолютна відносна похибка (MAPE): Показує похибку у відсотках від реального значення.
- Точність прогнозу напрямку руху графіку: Показує з якою вірогідністю ціна росте відповідно із прогнозом росту та падає при прогнозі падіння ціни.
- Ріст ціни у випадках прогнозу її росту: сума всіх значень росту ціни у випадку прогнозу її росту. А також аналогічний показник для випадків прогнозу падіння та загальний.
- Падіння ціни у випадках прогнозу її росту: сума всіх значень зменшення ціни у випадку прогнозу її росту. А також аналогічний показник для випадків прогнозу падіння та загальний.
- Різниця між ростом ціни та її падінням (між загальним ростом і падінням та ростом і падінням при слідуванні прогнозу моделі) за час дослідження: у загальному показує загальну вигоду від вкладання у фінансовий інструмент за досліджений період.

2.2 Дослідження методів прогнозування часових рядів

Почнемо дослідження із метода K-Means Clustering.

Як описано вище у пункті (1.3.1), цей метод не придатний для прогнозування самотійно, проте він може виступати допоміжним. За його допомогою можна виділити групи подібних за даних, що може вказувати на сезонність. У такому разі результати кластеризації доцільно використати як один із видів вхідних даних. Перевіримо роботу метода для різної кількості кластерів. Результати кластеризації наведені далі.

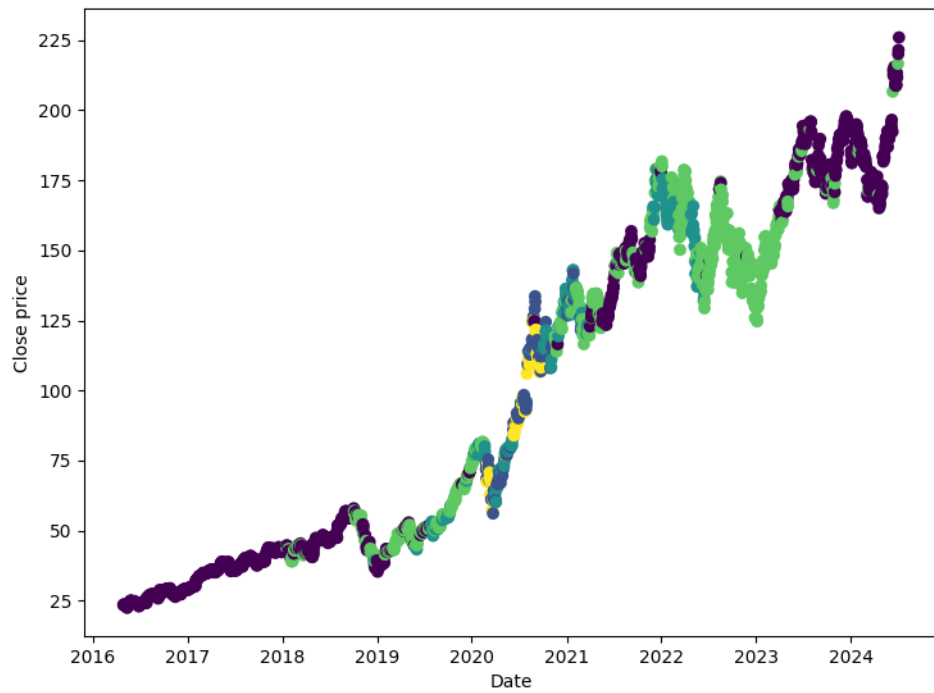


Рисунок 5. Кластеризований графік історії змін закриваючої ціни, кількість кластерів – 5.

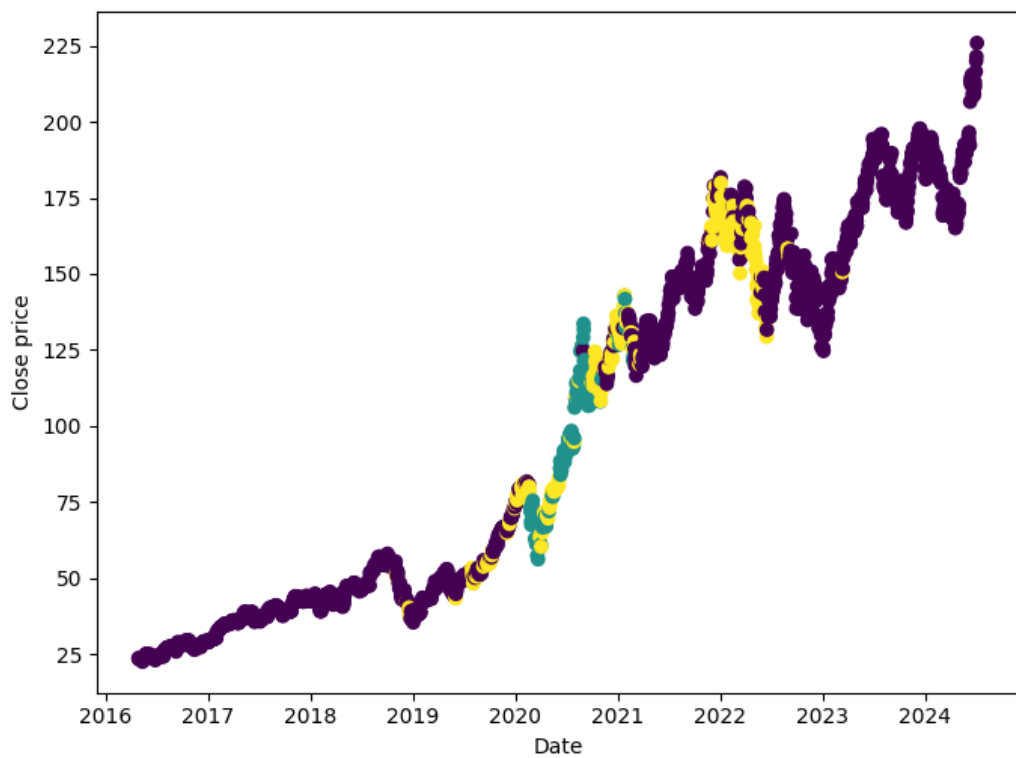


Рисунок 6. Кластеризований графік історії змін закриваючої ціни, кількість кластерів – 3.

Як видно із рисунків (5,6), дані які метод об'єднав у групи не виділяються якоюсь особливою поведінкою. Схоже що для прогнозування даного часового ряду, додаткове використання K-Means Clustering не є доцільним, використання його даних у навчанні методів прогнозування буде лише зайвим шумом.

Оскільки реалізація користі від даного методу можлива не для всіх часових рядів, а обсяг роботи для розробки окремого програмного рішення, для оцінки кожного фінансового інструмента і вибіркового використання K-Means Clustering для різних фінансових інструментів, можна попередньо оцінити як завеликий для можливої користі такого рішення, K-Means Clustering у подальшій роботі не буде використовуватися.

Перейдемо безпосередньо до прогнозування часових рядів. Почнемо із прогнозування на один крок за допомогою моделі “RandomForest”. Для роботи із цією моделлю була використана бібліотека “sklearn”[27]. Побудуємо кілька графіків із розділенням вибірок на навчальну та тестову у різні дати, та проведемо оцінку моделі за її прогнозами перших кроків. Досліджені характеристики моделі приведено далі.

```
MAE: 5.0237500000000015
MSE: 39.40397951440003
RMSE: 6.2772589172663595
MAPE: 7.06%
direction accuracy: 45.00%
```

Рисунок 7. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №1, частина №1.

```
price growth in cases of price growth forecast: 109.590
price growth in cases of a price drop forecast: 100.640
total price increase: 210.230

price drop in cases of price growth forecast: -110.480
price drop in cases of price drop forecast: -52.530
total price drop: -163.010

benefit in cases of price growth forecast: -0.890
benefit in cases of price drop forecast: 48.110
total price benefit: 47.220
```

Рисунок 8. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №1, частина №2.

Проаналізуємо дані на рисунках (7,8). Враховуючи що значення ціни знаходяться діапазоні від 22.55 до 225.93, здобуті характеристики MAE, MSE, RMSE та MAPE дуже хороші. Точність прогнозованого напрямку 45%, що корелює із тим що вигода від прийняття рішення протилежно прогнозу моделі перевищує вигоду від слідування її прогнозу. Варто зауважити, що прийняття рішень протилежно прогнозу моделі підняло вигоду від використання фінансового інструменту ($48.11 > 47.22$), хоча і незначно.

Перевіримо графік часового ряду ціни та її прогнозу на наявність аномалій, він наведений далі.



Рисунок 9. Графік прогнозу ціни “close” моделлю “RandomForest”.

Перше що кидається в очі на рисунку (9) це те, що поведінка моделі схожа на те, що вона не здатна прогнозувати ціну вище тої, яка була у навчальній вибірці. Для настільки хороших характеристик, які були досліджені у моделі, така поведінка здається дуже дивною, та може свідчити про помилку при побудові моделі чи оцінці її характеристик. Проте обміркувавши ситуація я дійшов до висновку, що ця аномалія пов’язана із зосередженістю. Перевіримо характеристики моделі для пари інших фінансових інструментів.

```
MAE: 7.053947826086999
MSE: 65.24239657652205
RMSE: 8.077276556892308
MAPE: 3.65%
direction accuracy: 34.78%
```

Рисунок 10. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №2, частина №1.

```
price growth in cases of price growth forecast: 0.400
price growth in cases of a price drop forecast: 58.250
total price increase: 58.650

price drop in cases of price growth forecast: -12.160
price drop in cases of price drop forecast: -31.310
total price drop: -43.470

benefit in cases of price growth forecast: -11.760
benefit in cases of price drop forecast: 26.940
total price benefit: 15.180
```

Рисунок 11. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №2, частина №2.

Характеристики моделі рисунках (10,11) підтверджують схильність моделі до прогнозування напрямку зміни ціни протилежно її реальному напрямку зміни. Модуль досі збільшує ефективність від використання фінансового інструменту ($26.94 > 15.18$), і для другого інструмента вона працює краще ніж для першого. При такій поведінці модель є працездатною, проте рішення щодо використання фінансових інструментів потрібно приймати протилежно її прогнозам.

Перейдемо до наступного фінансового інструмента.

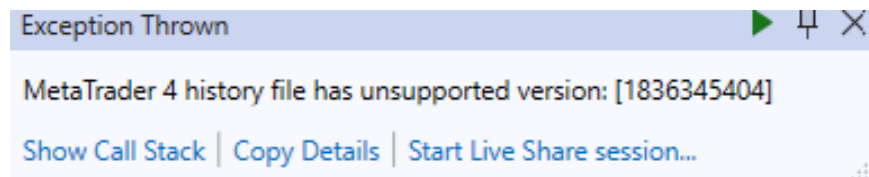


Рисунок 12. Помилка читання файлу “.hst”.

Як видно із рисунку (12), модуль “mt4forexparser”[24] не дозволяє зчитувати дані із всіх версій розширення “.hst”. Надалі у дослідженні, та у

розробці автоматизованої системи, для зчитування даних по часовим рядам будуть використовуватися файли формату “.csv”.

```
MAE: 1.0262579999999999
MSE: 1.5771178330000002
RMSE: 1.255833521212108
MAPE: 7.42%
direction accuracy: 46.00%
```

Рисунок 13. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №3, частина №1.

```
price growth in cases of price growth forecast: 7.880
price growth in cases of a price drop forecast: 17.840
total price increase: 25.720

price drop in cases of price growth forecast: -3.540
price drop in cases of price drop forecast: -5.930
total price drop: -9.470

benefit in cases of price growth forecast: 4.340
benefit in cases of price drop forecast: 11.910
total price benefit: 16.250
```

Рисунок 14. Характеристики прогнозування моделі “RandomForest”, тестовий фінансовий інструмент №3, частина №2.

Дані на рисунках (13 і 14) показують що модель не може збільшити ефективність від вкладів у третій фінансовий інструмент. Тобто завжди приймати рішення протилежно прогнозу моделі недостатньо для того, щоб завжди здобувати корисну роботу від неї, потрібно окремо визначати, чи ефективно модель працює із кожним фінансовим інструментом окремо, та вірогідно окремо для актуального проміжку часу.

Перейдемо до застосування моделі “ARIMA”. Для роботи із цією моделлю була використана бібліотека “pmdarima”[28]. Повторимо навчання моделі на ряду вибірок і подивимося на її характеристики на рисунках (15,16).

```
MAE: 4.612124112470352
MSE: 33.49911149029436
RMSE: 5.787841695338113
MAPE: 6.56%
direction accuracy: 58.00%
```

Рисунок 15. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №1, частина №1.

```
price growth in cases of price growth forecast: 152.540
price growth in cases of a price drop forecast: 57.690
total price increase: 210.230

price drop in cases of price growth forecast: -90.180
price drop in cases of price drop forecast: -72.830
total price drop: -163.010

benefit in cases of price growth forecast: 62.360
benefit in cases of price drop forecast: -15.140
total price benefit: 47.220
```

Рисунок 16. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №1, частина №2.

Як видно із рисунків (15,16), модель підняла вигоду від вкладу у фінансовий інструмент ($62.36 > 47.22$). Також якщо порівняти рисунки (15,16) та (7,8), можна побачити що характеристики “ARIMA” краще за характеристики моделі “RandomForest” для аналогічного фінансового інструменту. При цьому, для використання моделі “ARIMA” достатньо просто слідувати її прогнозам, перевіримо чи буде це твердження актуальне для решти фінансових інструментів. Характеристики моделі для наступного інструмента представлені на рисунках (17,18).

```
MAE: 6.18676934940763
MSE: 57.970838857915375
RMSE: 7.613858342385638
MAPE: 3.21%
direction accuracy: 56.52%
```

Рисунок 17. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №2, частина №1.

```

price growth in cases of price growth forecast: 58.250
price growth in cases of a price drop forecast: 0.400
total price increase: 58.650

price drop in cases of price growth forecast: -36.780
price drop in cases of price drop forecast: -6.690
total price drop: -43.470

benefit in cases of price growth forecast: 21.470
benefit in cases of price drop forecast: -6.290
total price benefit: 15.180

```

Рисунок 18. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №2, частина №2.

По рисунках (17,18) можна побачити, що модель “ARIMA” дозволяє збільшувати ефективність від вкладів у фінансовий інструмент №2, але якщо порівняти характеристики “ARIMA” із характеристиками на рисунку (11), для цього інструменту вона працює не так добре як “RandomForest” ($21.47 < 26.94$). Проте для моделі “ARIMA” достатньо просто слідування її прогнозам, а не діяти всупереч ним, чи окремо визначати ефективність моделі для інструменту.

```

MAE: 0.7983365666847985
MSE: 0.9334992239769125
RMSE: 0.9661776358294123
MAPE: 5.83%
direction accuracy: 52.00%

```

Рисунок 19. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №3, частина №1.

```

price growth in cases of price growth forecast: 11.680
price growth in cases of a price drop forecast: 14.040
total price increase: 25.720

price drop in cases of price growth forecast: -3.360
price drop in cases of price drop forecast: -6.110
total price drop: -9.470

benefit in cases of price growth forecast: 8.320
benefit in cases of price drop forecast: 7.930
total price benefit: 16.250

```

Рисунок 20. Характеристики прогнозування моделі “ARIMA”, тестовий фінансовий інструмент №3, частина №2.

Оглянувши характеристики моделі на рисунках (19,20), може скластися враження що слідування прогнозу моделі для фінансового інструмента №3 не робить вкладання у нього краще, проте це не зовсім так. Якщо порівняти різницю між ростом і падінням ціни із її загальним ростом, можна побачити що у після відрахування втрат “total price benefit” складає 63% від “total price increase”, тоді як у випадку слідування прогнозам моделі ми маємо “benefit in cases of price growth forecast” складає 71% від “price growth in cases of price growth forecast”, тобто слідування прогнозам моделі робить вкладання у фінансовий інструмент №3 ефективнішим, але у загальному модель знайшла небагато випадків, коли можна зробити вигідні вклади. Зокрема описаної характеристики у цілому виглядають нормальними та не вказують відзнак аномалій. Для того щоб впевнитися у їх відсутності, проведемо дослід графіку прогнозування моделлю, він наведений далі.



Рисунок 21. Графік прогнозовану закриваючої ціни моделлю “ARIMA”.

Дослідженням графіку на рисунку (21) не виявлено нічого незвичайного. Перейдемо до моделі на основі нейронної мережі типу моделі “LSTM”. Для

роботи із цією моделлю були використані бібліотеки “keras”[29] і “tensorflow”[30].

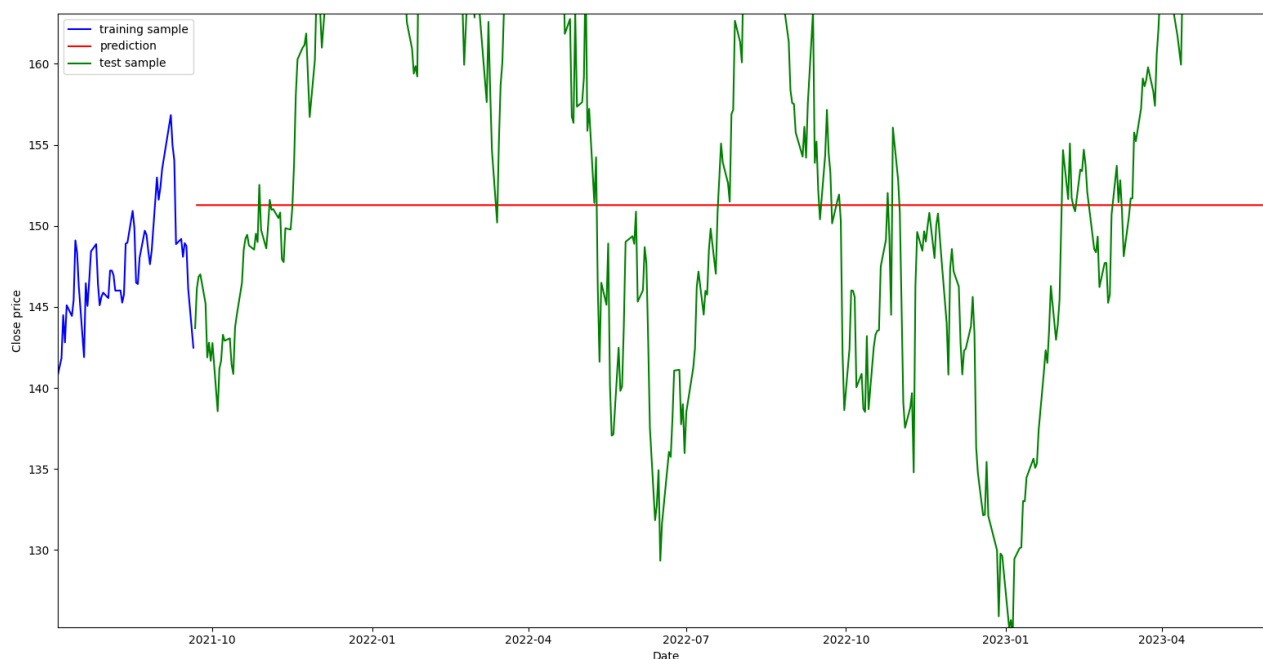


Рисунок 22. Характеристики прогнозування моделі “LSTM” на один крок вперед.

Як видно із рисунку (22) модель “LSTM” не працює. Незалежно як довго я навчав модель, та які конфігурації налаштувань виставляв, нейронна мережа “LSTM” у моєму виконанні не виконує ніякої корисної роботи.

2.3 Порівняння методів та вибір найкращого для застосування у автоматизованій системі керування

У результаті дослідження методів прогнозування часових рядів було отримано дві робочі моделі.

Порівнявши характеристики моделей “RandomForest” на рисунках (8, 11 і 14) із характеристиками моделей “ARIMA” на рисунках (16,18 і 20), можна побачити що для фінансових інструментів №1 і №3 ефективніше себе показує модель “ARIMA”, а для фінансового інструменту №2 більшу ефективність показує “RandomForest”.

Модель “RandomForest” показує найбільшу ефективність лише в одному випадку, також вона нестабільна та виказує незрозумілу поведінку на графіку.

Модель “ARIMA” стабільно видає позитивні результати при слідуванні її прогнозам та не виказує наявності аномалій у роботі.

Ідеальним рішенням, з точки зору максимізації середньої вірогідності слідування прогнозам моделі, було б скомбінувати обидві моделі для подальшої розробки автоматизованої системи керування ділінгом. Для цього також потрібно було б зайнятися розробкою окремого програмного рішення, яке буде проводити внутрішнє дослідження кожного фінансового інструменту на предмет того, яка модель покаже себе ефективніше при роботі із ним. Проте для кількості перевірок, які необхідні для порівняння моделей у такому досліді, потрібно витратити щонайменше 20 хвилин (суб’єктивне спостереження). При наявності у портфелі активів десятків фінансових інструментів, проведення такого дослідження буде займати години, а при наявності сотні інструментів – більше дня, тобто за час підбору оптимальних налаштувань для моделей – вони можуть втратити актуальність.

Тому для розробки автоматизованої системи буде використовуватися модель побудована методом “ARIMA”.

2.4 Висновки по розділу

Даний розділ було присвячено виконанню завдання дослідження принципів роботи описаних у попередньому розділі актуальних методів для визначення цінності фінансових інструментів. Дослідження було спрямовано на порівняння моделей, побудованих методами “RandomForest” і “ARIMA”, при роботі із різними фінансовими інструментами. Після порівняння цих моделей, для використання у розробці програмного забезпечення, яке реалізує автоматизовану систему керування фондовим ділінгом, була обрана модель “ARIMA”.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ФОНДОВИМ ДІЛІНГОМ

3.1 Структура програмного забезпечення

Поведінку програмного забезпечення, яке розроблено у даній роботі, можна представити у вигляді UML діаграми діяльності, яка представлена на рисунках (23,24):

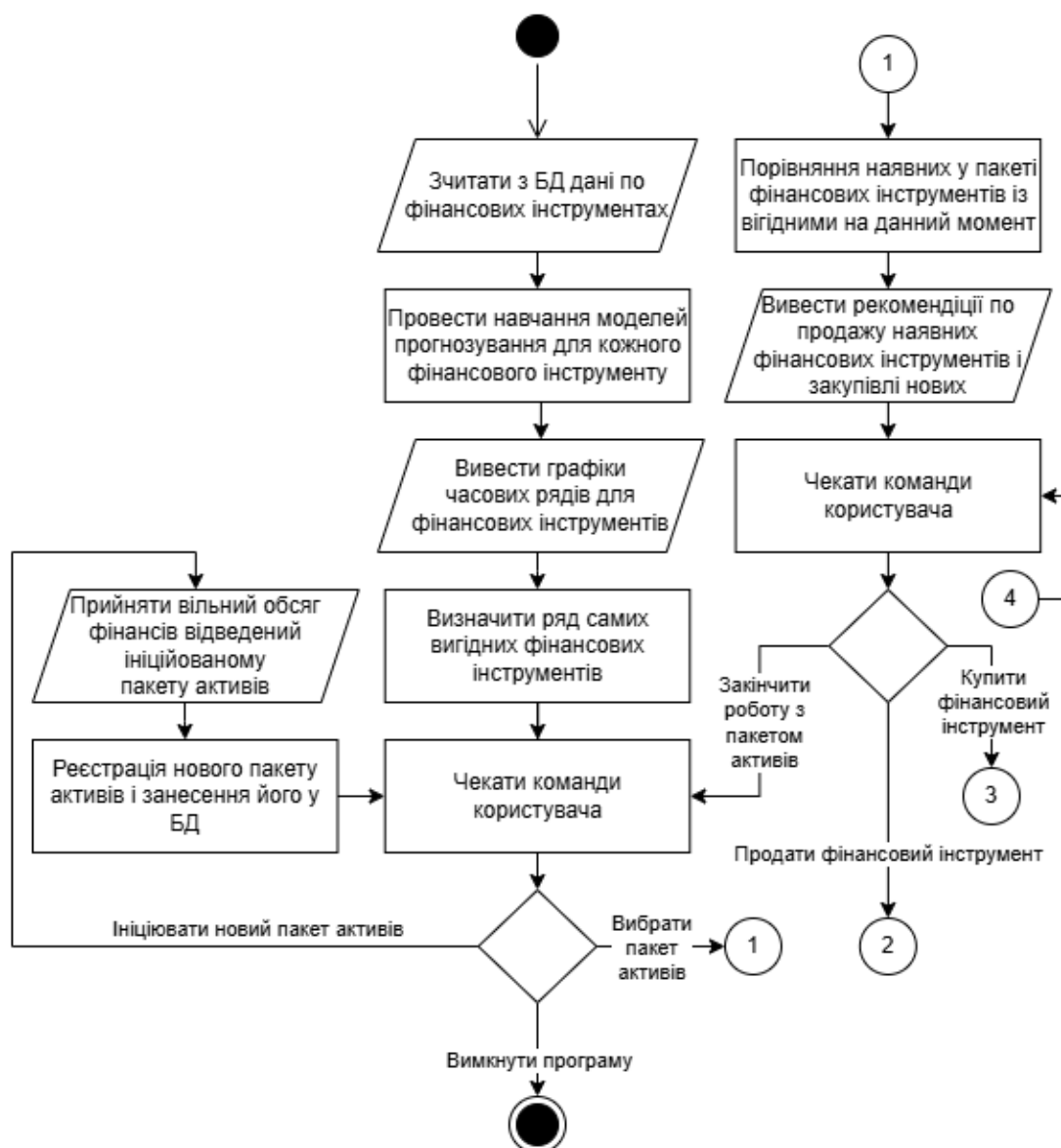


Рисунок 23. UML діаграми діяльності програмного забезпечення, частина 1.

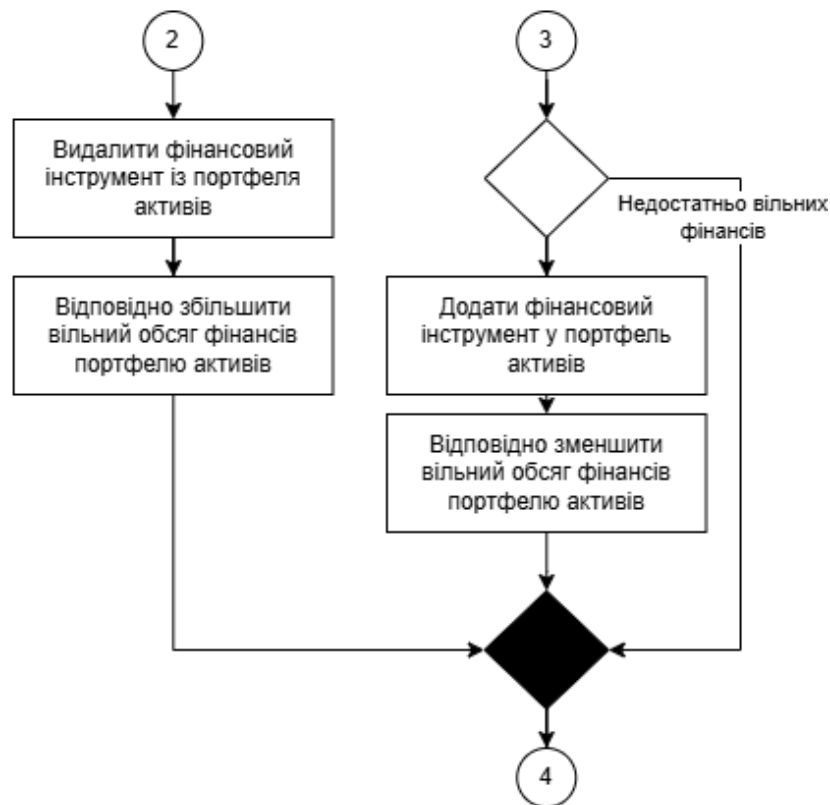


Рисунок 24. UML діаграми діяльності програмного забезпечення, частина 2.

Дане програмне забезпечення призначене для прогнозування ціни на акції та інші фінансові інструменти, а також для визначення ряду самих вигідних інструментів із прогнозованих, із яких потрібно скласти портфель активів.

Програмне забезпечення також буде підтримувати роботу із багатьма портфелями активів, відповідно для роботи відділу банку як від свого імені, так і від імені клієнтів.

При початку роботи, програма зачекає вибір користувачем існуючого в БД портфелю активів, чи ініціалізацію нового. Далі вона зчитає дані по наявним у портфелі фінансовим інструментам із БД, після чого приступить до побудови моделей для прогнозування, за результатами роботи яких виведе рекомендації щодо розподілу фінансів між самими вигідними, для формування портфелю активів, фінансовими інструментами.

Після розрахунків, програма дасть користувачу можливість продивитися графіки часових рядів фінансових інструментів для власної аналітики. Також, за

наявності даних по новим фінансовим інструментам, користувач може додати їх до портфеля активів, або вилучити із портфеля наявні.

Після проведення користувачем усіх бажаних заходів по купівлі/продажу в усіх портфелях активів, які його цікавлять, програма закінчить роботу.

Програмне забезпечення складається із наступних компонентів:

- Виконавчий (“EXE”) файл системи керування.
- Файл бази даних із портфелями активів, з якими працює система керування.
- Історичні файли (“csv”) по кожному із використовуваних фінансових інструментів.

Для коректної праці програмного забезпечення необхідні наступні умови:

- Усі файли ПО повинні знаходитися у одній папці.
- Потрібно встановити наступні не стандартні бібліотеку “rmdarima”.

3.2 Опис інтерфейсу

Інтерфейс програмного забезпечення було написано за допомогою бібліотеки “tkinter”[31], яка пропонує скласти віконний інтерфейс із запропонованих елементів – віджетів[32]. Під час написання інтерфейсу програмного забезпечення я користувався як офіційним сайтом бібліотеки, так й іншими мануалами і відео-уроками[33], зокрема по властивостям віджетів[34] та їх розміщенню на вікні[35].

Інтерфейс програмного забезпечення складається із чотирьох вікон. Перше вікно надає користувачу вибір між існуючими пакетами активів, а також можливість створити новий. Після вибору портфеля вікно закриється, і відкриється вікно для взаємодії із обраним портфелем.

У вікні взаємодії із портфелем активів є три умовні зони:

- Зона огляду фінансових інструментів, у який можна вибрати фінансові інструменти, та вивести окреме вікно із часовими рядами цих інструментів.

- Зона налаштування портфеля, у якій можна додати до портфеля новий фінансовий інструмент, вибравши “.csv” файл з даними його історії через окреме вікно огляду директорії файлів. Також у цій зоні можна видалити із портфеля присутній у портфелі інструмент. Також у цій зоні виводяться рекомендації по розподілу фінансів між обраними для портфеля активами.
- Зона трейдингу, де можна оглянути баланс фінансів і активів, а також перевести фінанси у активи і навпаки.

3.3 Тестування програмного забезпечення

3.3.1 Тестування інтерфейсу та основних функцій системи

Проведемо тестування програмного забезпечення. Для початку запустимо його. Вікно для вибору портфелю активів, із яким користувач хоче працювати показано на рисунку (25)

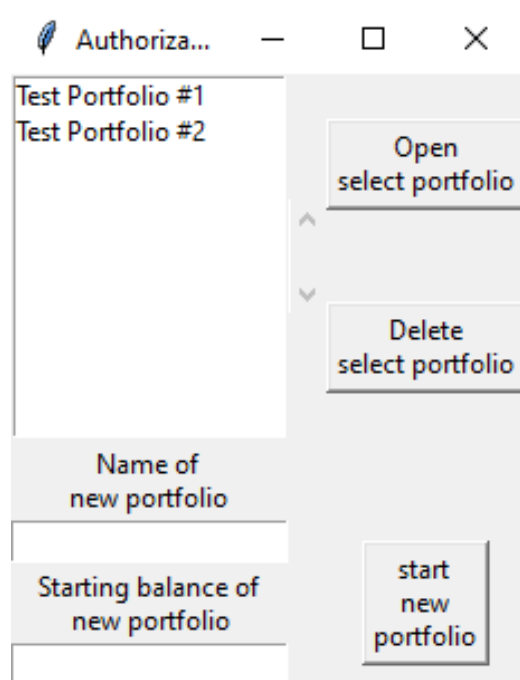


Рисунок 25. Початкове вікно програмного забезпечення.

Програма успішно запустилася. У списку вже присутні деякі тестові портфелі, які були створені під час розробки, видалимо один із них, вибравши його у списку та натиснувши кнопку “Delete select portfolio”.

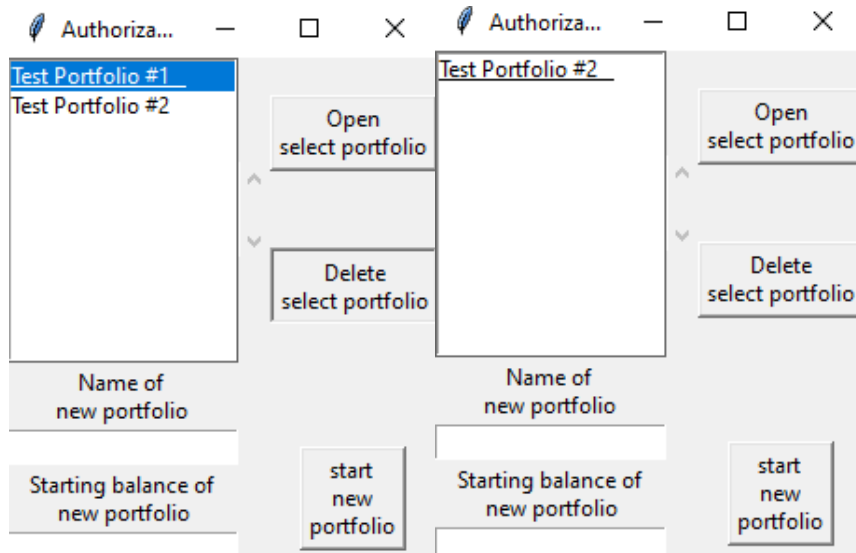


Рисунок 26. Тестування функції видалення портфеля активів.

Як видно на рисунку (26), функція видалення портфеля працює.

Протестуємо функцію створення нового портфеля. Введемо найменування портфеля та його стартовий баланс та натиснемо кнопку “start new portfolio”, також натиснемо цю кнопку передчасно та після створення портфеля, щоб перевірити як програма себе поведе у таких випадках.

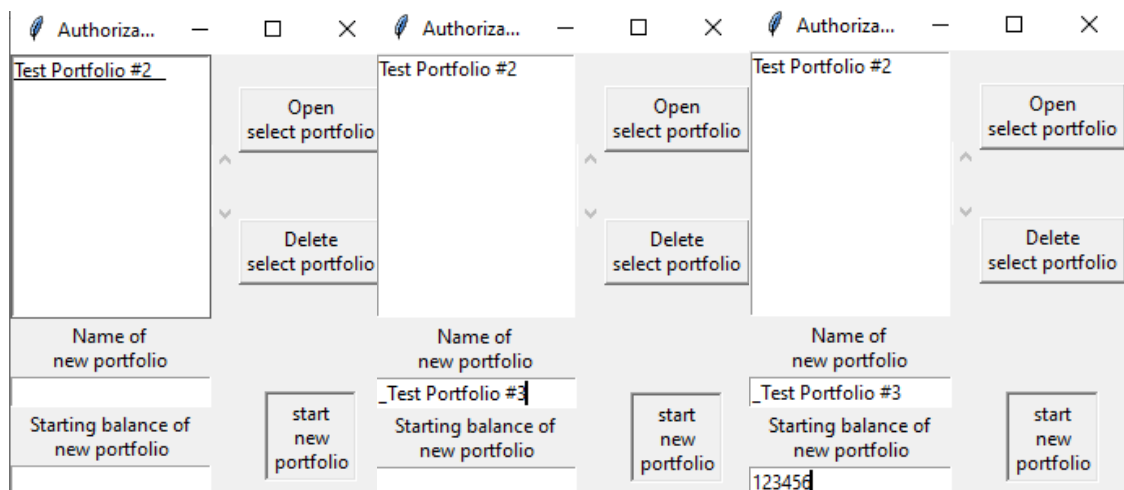


Рисунок 27. Тестування функції створення портфеля активів, частина №1.

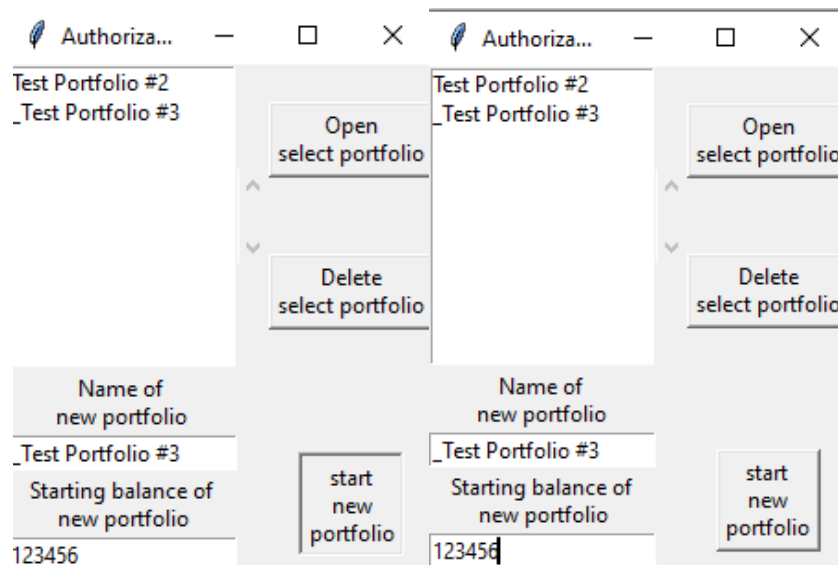


Рисунок 28. Тестування функції створення портфеля активів, частина №2.

Як видно на рисунках (27,28), функція створення портфеля активів працює, а також вона не реагує на спроби створення дублікату портфеля, чи на спроби створення портфеля із неналежним вводом даних.

Виберемо створений портфель та натиснемо кнопку “Open selected portfolio”, ця операція представлена далі на рисунку (29).

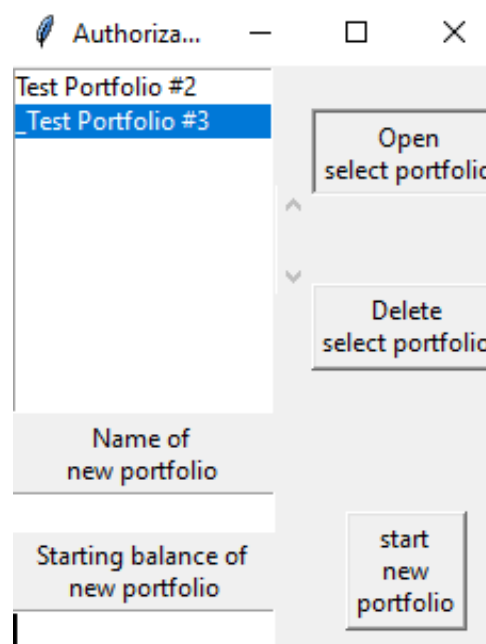


Рисунок 29. Тестування функції відкриття портфеля активів.

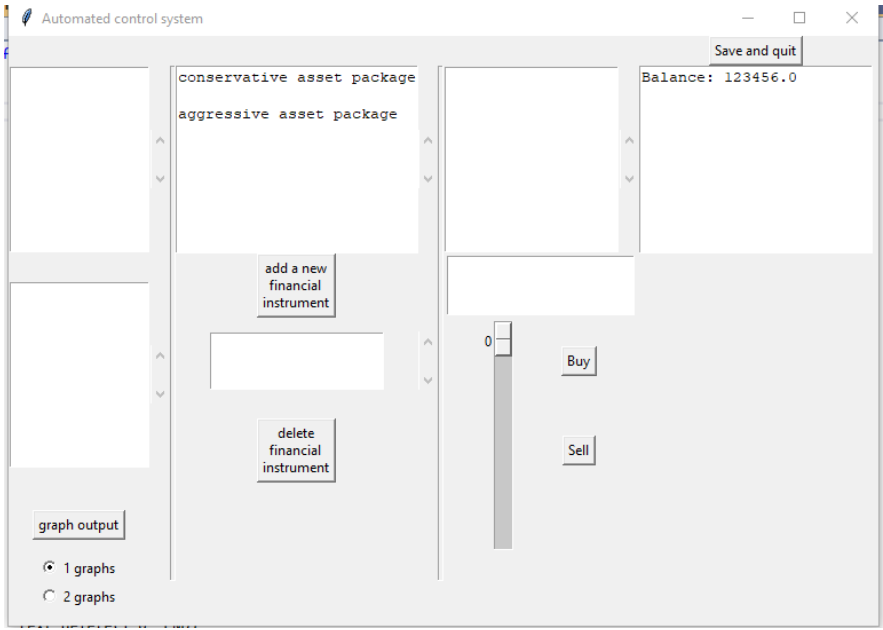


Рисунок 30. Вікно роботи з портфелем активів, частина №1.

На рисунку (30) представлено новий, пустий портфель активів, додаймо у нього кілька фінансових інструментів через підменю, яке можна відкрити натиснувши кнопку “add a new financial instrument”. Далі на рисунку (31) показано підменю для вибору файлу із історією цін для нового фінансового інструменту.

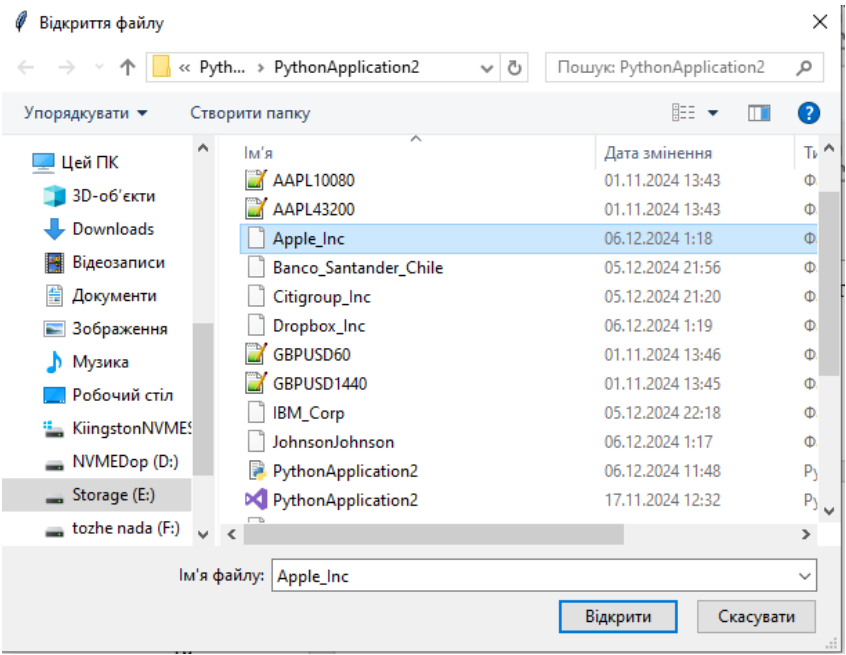


Рисунок 31. Вікно додавання нового фінансового інструменту у портфель.

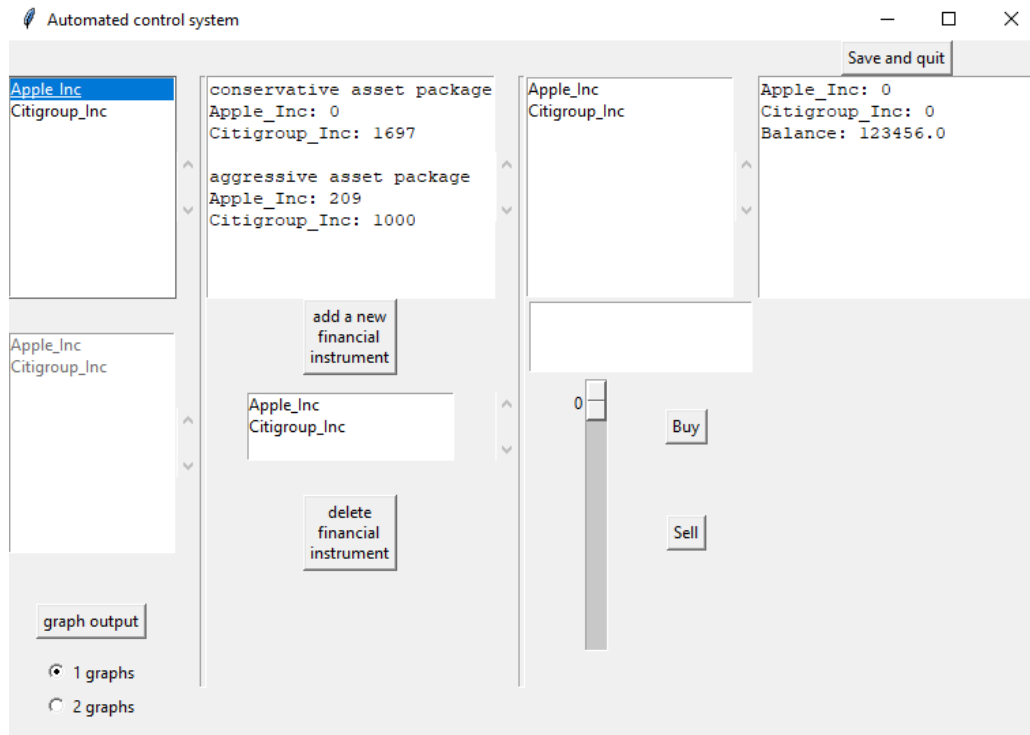


Рисунок 32. Вікно роботи з портфелем активів, частина №2.

На рисунку (32) можна побачити, що із фінансовими інструментами в портфелі система керування вивела у текстовий віджет ряд рекомендацій по розподілу фінансів між наявними фінансовими інструментами, відповідно для складання консервативного та агресивного портфеля активів.

Система керування порекомендувала закупити 0 одиниць фінансового інструменту “Apple Inc” і 1697 одиниці “Citigroup Inc” у рамках консервативного портфелю активів, та 209 і 1000 у рамках агресивного. Для того щоб проаналізувати причини цього, подивимося на графіки часових рядів цих інструментів.

Для того щоб вивести на екран графіки часових рядів, виберемо пункт “2 graphs”, щоб відкрити два графіки, виберемо цікаві нам фінансові інструменти у списках, та натиснемо кнопку “graph output”. Приклад того як це зробити наведено на рисунку (33).

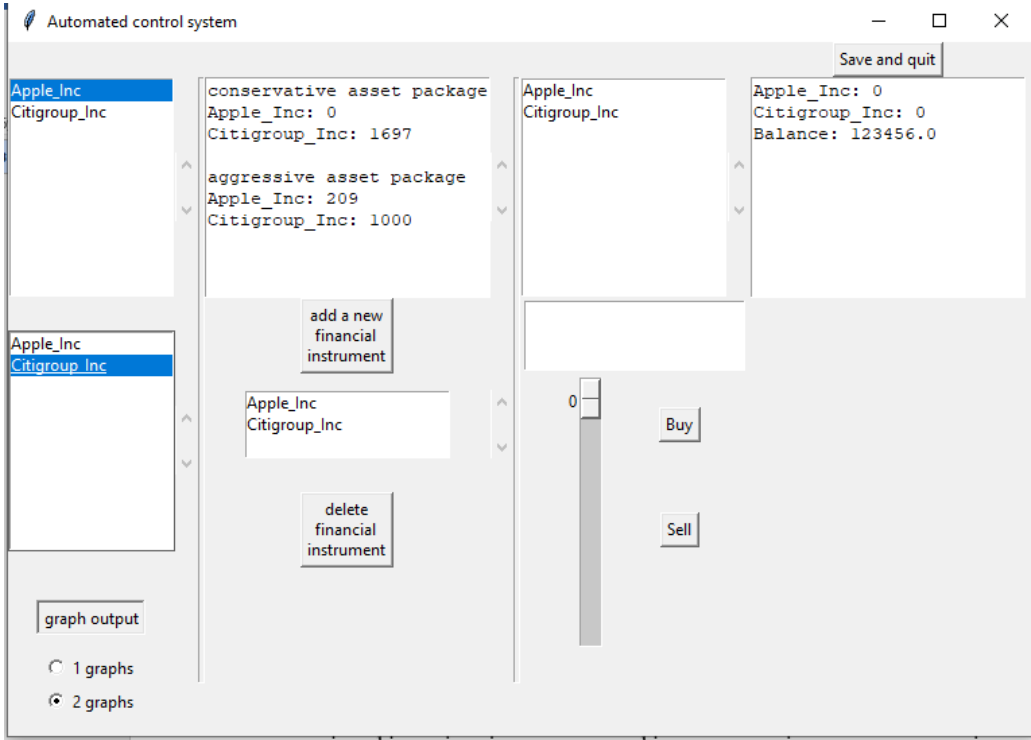


Рисунок 33. Вікно роботи з портфелем активів, частина №3.

На рисунку (34) зображено часові дані по інструментам за останні два місяці, а також прогноз на тиждень вперед.

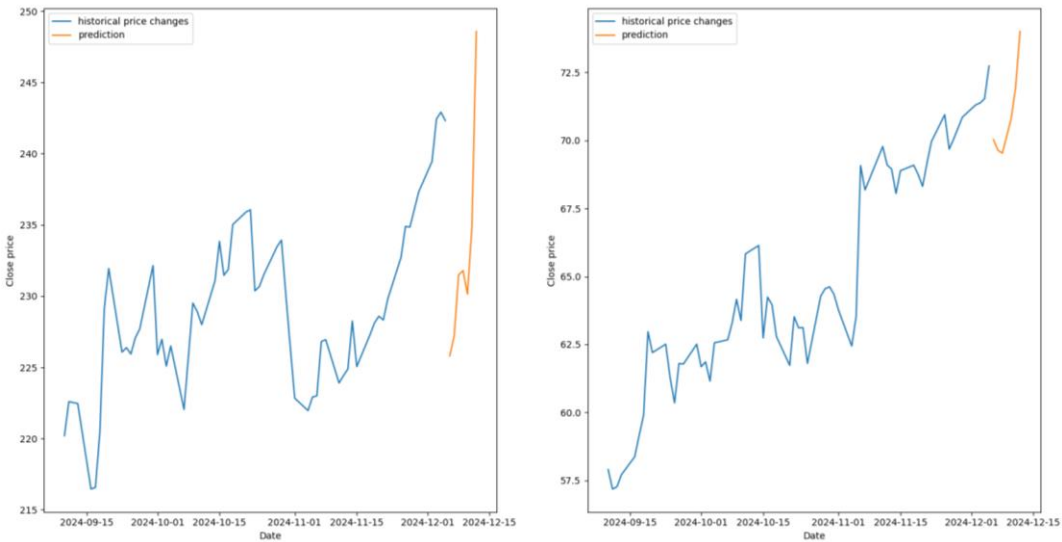


Рисунок 34. Графіки часових рядів для фінансових інструментів “Apple Inc” і “Citigroup Inc” , частина №1.

Проаналізувавши рисунок (34), можна помітити що середнє спадання ціни “Apple Inc” більше ніж “Citigroup Inc”, тому у рамках консервативної

стратегії[36] вкладів належно розподілити фінанси між інструментами із перевагою для “Citigroup Inc”. Також, через особливості роботи алгоритму система відмовляється рекомендувати у консервативний портфель самий швидко-спадаючий фінансовий інструмент. Відповідно система керування і надала рекомендації щодо складання портфеля у співвідношення 0 до 1697.

Наблизимо відрізок графіка за поточну дату, для того щоб зосередитися на актуальній інформації.

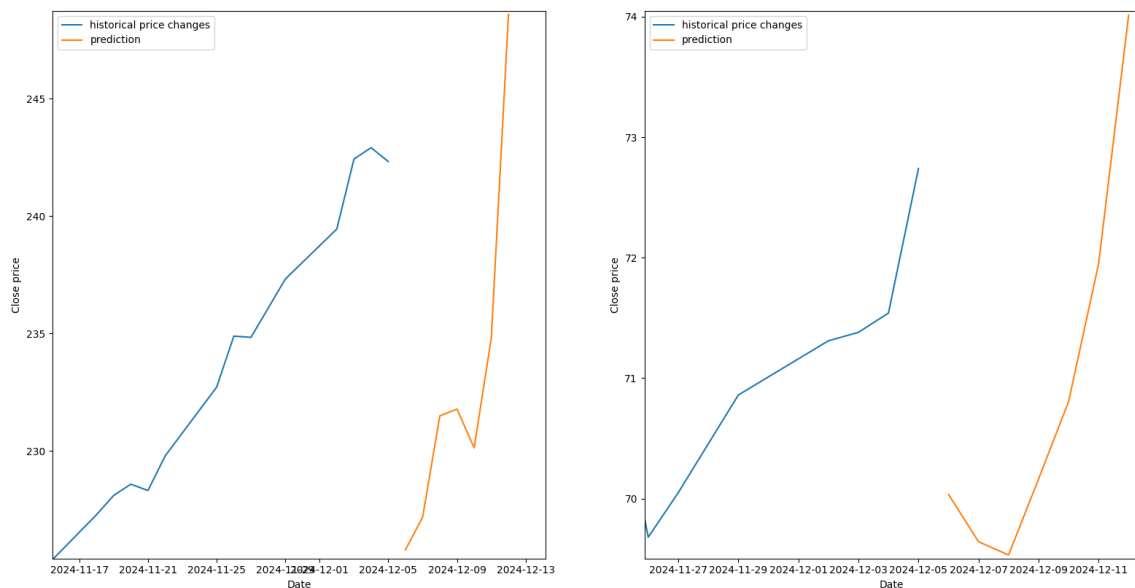


Рисунок 35. Графіки часових рядів для фінансових інструментів “Apple Inc” і “Citigroup Inc” , частина №2.

Як видно на рисунку (35), на наступний тиждень передбачається зріст обидвох фінансового інструмента “Apple Inc” та “Citigroup Inc”, відповідно до агресивної[36] стратегії система порекомендувала вкладати фінанси у обидва інструмента, розподіл фінансів між інструментами залежить від середнього зростання інструменту за минулі два місяці.

Спробуємо закупити фінансові інструменти. Для цього повернемося до вікна взаємодії із портфелем активів, та виберемо кількість акцій/цінних паперів для трейдингу за допомогою повзунка.

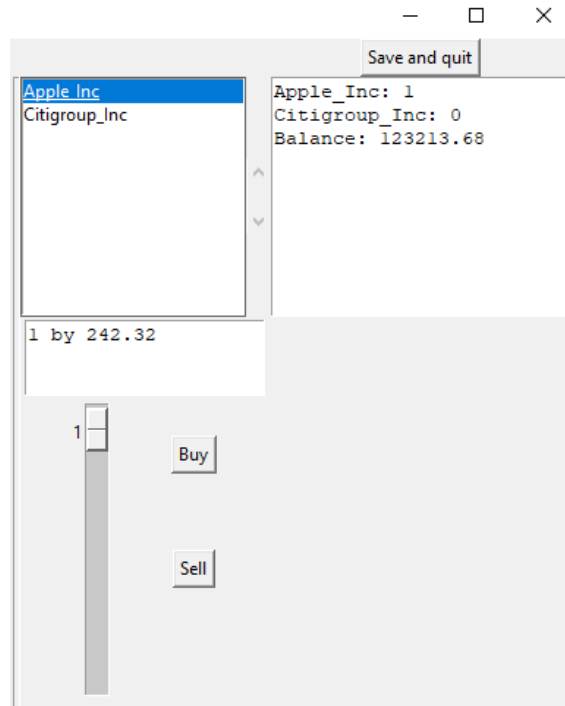


Рисунок 36. Вікно роботи з портфелем активів, частина №4.

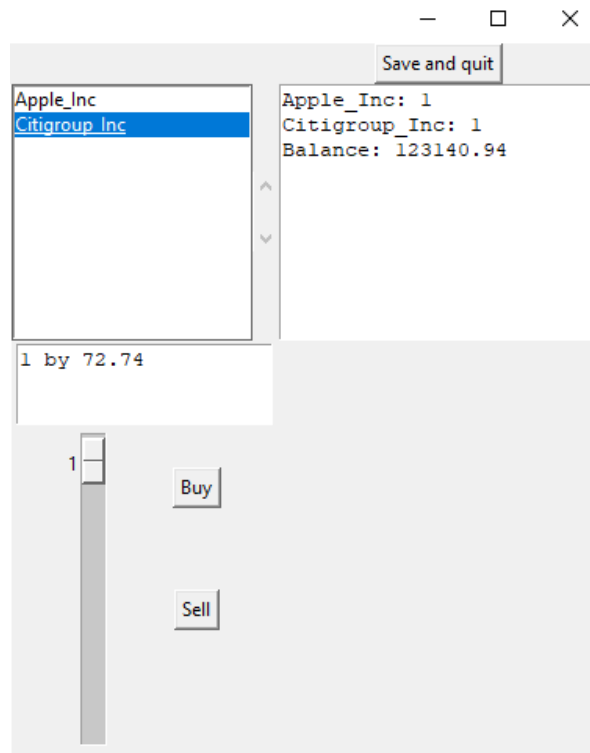


Рисунок 37. Вікно роботи з портфелем активів, частина №5.

На рисунках (36,37) можна побачити по скільки буде коштувати купівля/продаж відповідної кількості обраних фінансових інструментів. Для

проведення купівлі або продажу потрібно натиснути відповідно кнопку “Buy” або “Sell”. Здійснимо продаж закуплених фінансових інструментів.

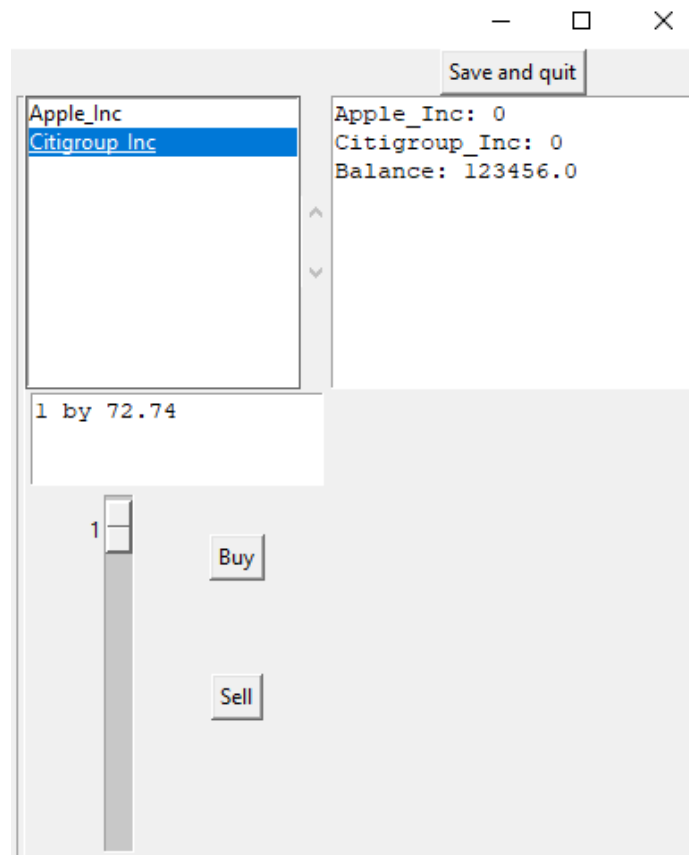


Рисунок 38. Вікно роботи з портфелем активів, частина №6.

Як видно на рисунку (38), функція продажу успішно спрацювала, багів не виявлено.

При виділенні фінансових інструментів із портфеля активів система повинна автоматично продавати усі наявні акції цього інструменту. Для того щоб перевірити цю функцію закупимо фінансові інструменти ще раз, та вилучимо їх із портфеля активів за допомогою кнопки “delete financial instrument”, як показано на рисунку (39).

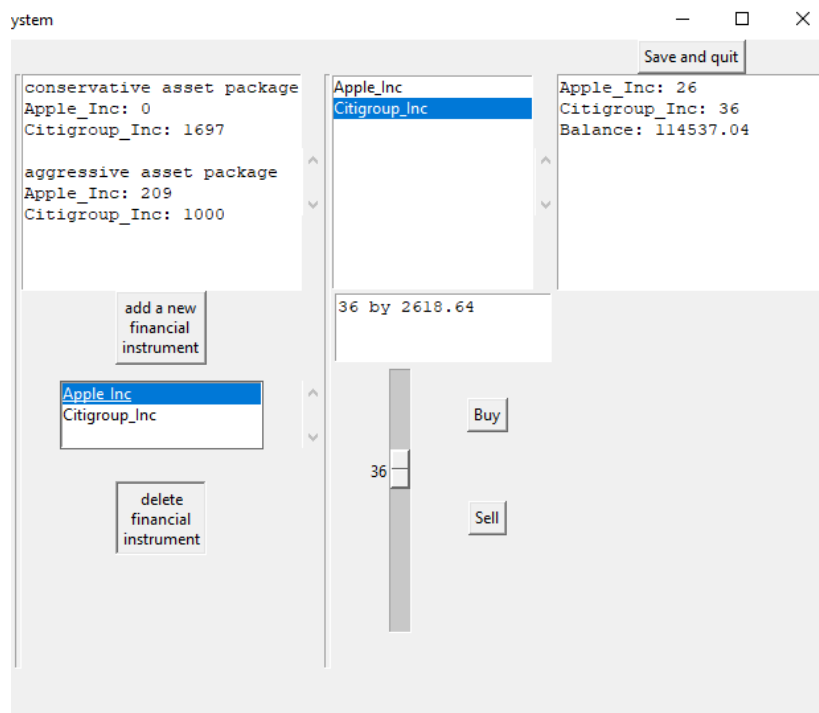


Рисунок 39. Вікно роботи з портфелем активів, частина №7.

Після видалення інструментів значення балансу фінансів повинно було повернутися до початково заданого значення.

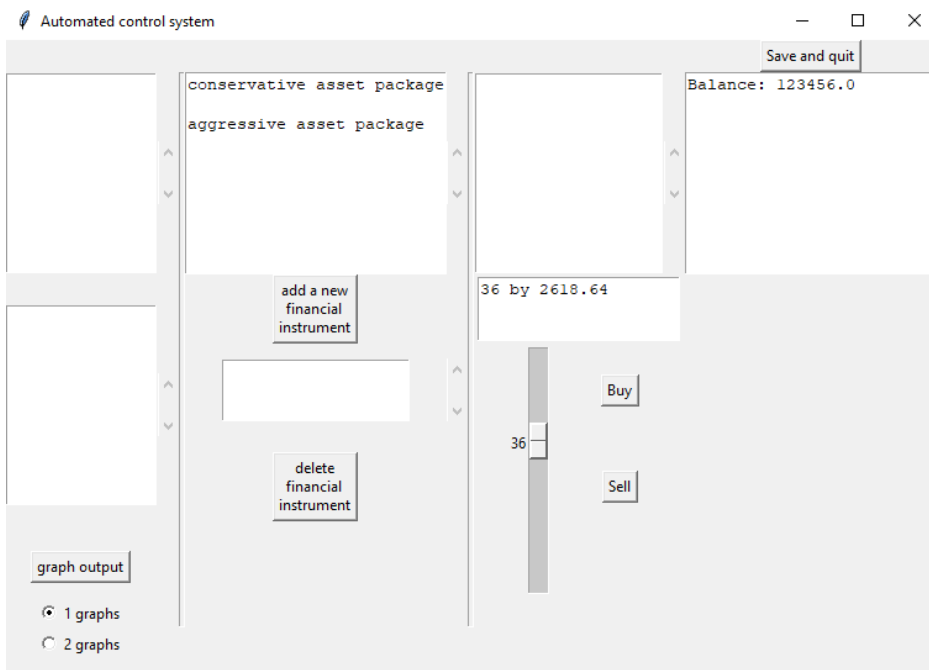


Рисунок 40. Вікно роботи з портфелем активів, частина №8.

На рисунку (40) можна побачити, що при видаленні фінансових інструментів їх запас на балансі був автоматично проданий, а початковий запас фінансів відновився у повному об'ємі.

Робота програми протестована, усі функції працюють коректно.

3.3.2 Тестування ефективності системи

Проведемо експеримент. Для тестування ефективності системи керування, будемо слідувати рекомендаціям системи керування по складанню консервативного та агресивного портфелів на протязі неділі, і подивимося за зміну балансу фінансів.

Для початку додаймо наявні фінансові інструменти до портфеля активів, подивимося як будуть виглядати рекомендації по складанню портфеля активів на рисунку (41).

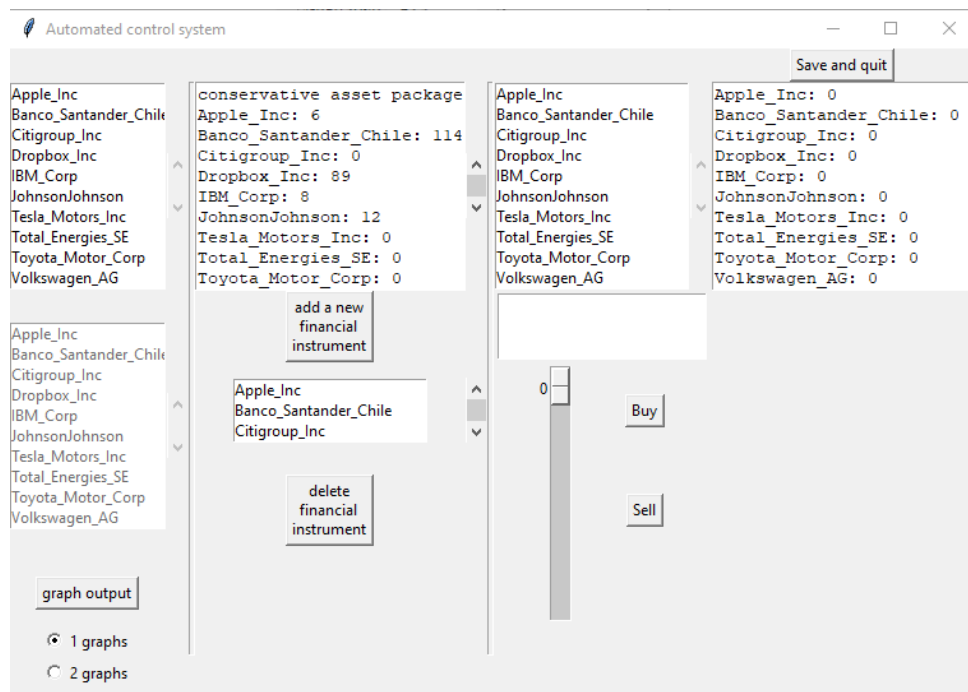


Рисунок 41. Вікно роботи з портфелем активів, частина №9.

Перед тим як перейдемо до закупки інструментів і переходу до наступного кроку, перевіримо графіки часових рядів.

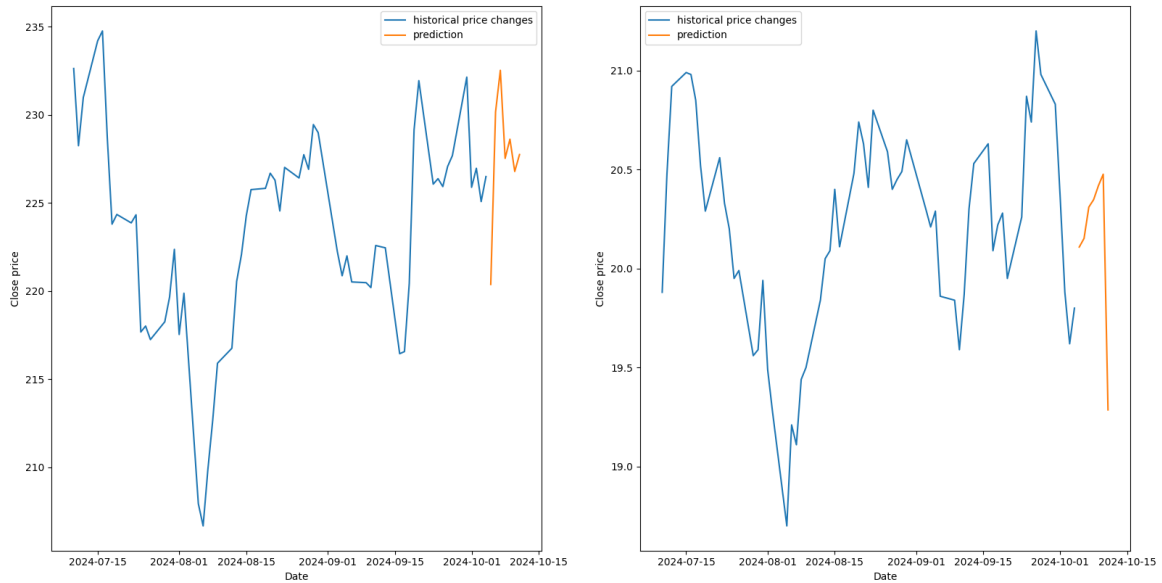


Рисунок 42. Часові ряди цін на “Apple Inc” та “Banco Santander Chile”.

Часові ряди на рисунку (42) не мають ярко вираженого тренду до зростання чи падіння, їх часові проміжки однакові, отже фінансові інструменти придатні для експерименту.

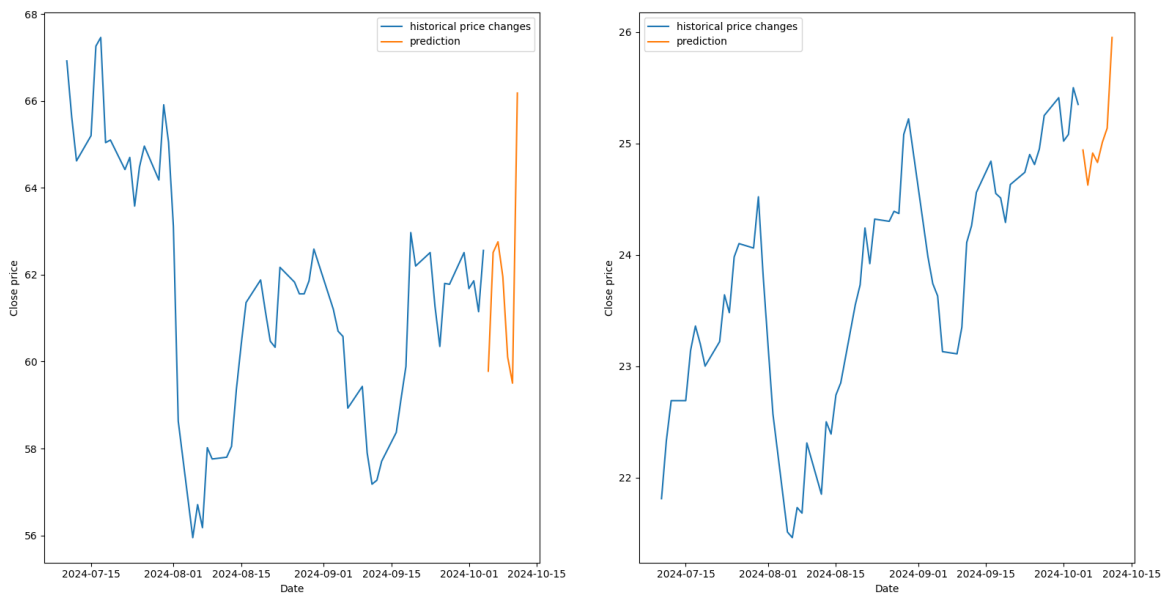


Рисунок 43. Часові ряди цін на “Citigroup Inc” та “Dropbox Inc”.

Часовий ряд “Citigroup Inc”, на рисунку (43), не має ярко вираженого тренду до зростання чи падіння, а “Dropbox Inc” зростає, їх часові проміжки однакові, отже фінансові інструменти придатні для експерименту.

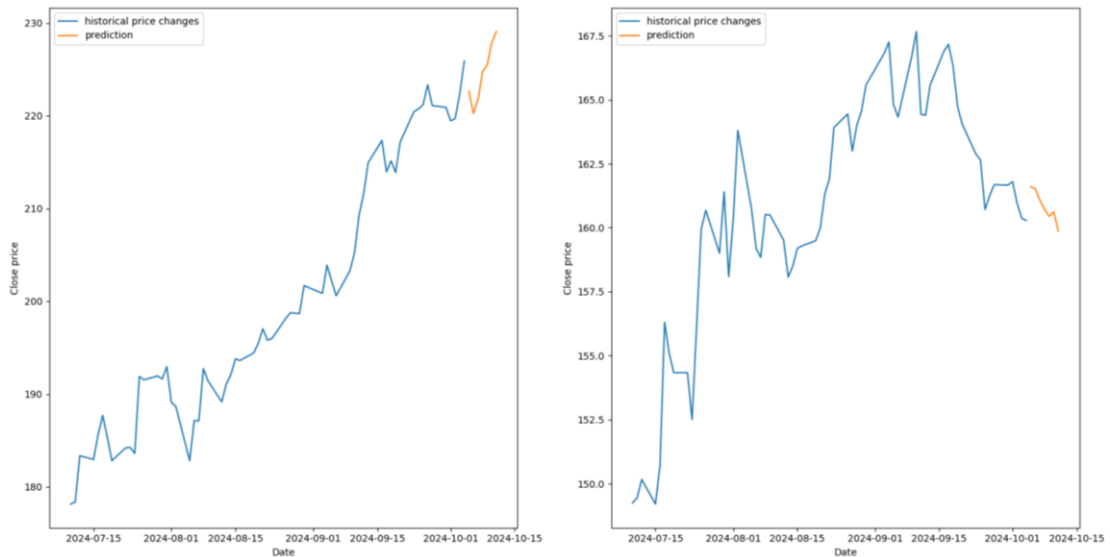


Рисунок 44. Часові ряди цін на “IBM Corp” та “JohnsonJohnson”.

Часовий ряд “IBM Corp” , на рисунку (44), має ярко виражений тренду до зростання, а “JohnsonJohnson” спочатку зростав, проте перейшов у падіння. Часові проміжки графіків однакові, отже фінансові інструменти придатні для експерименту.

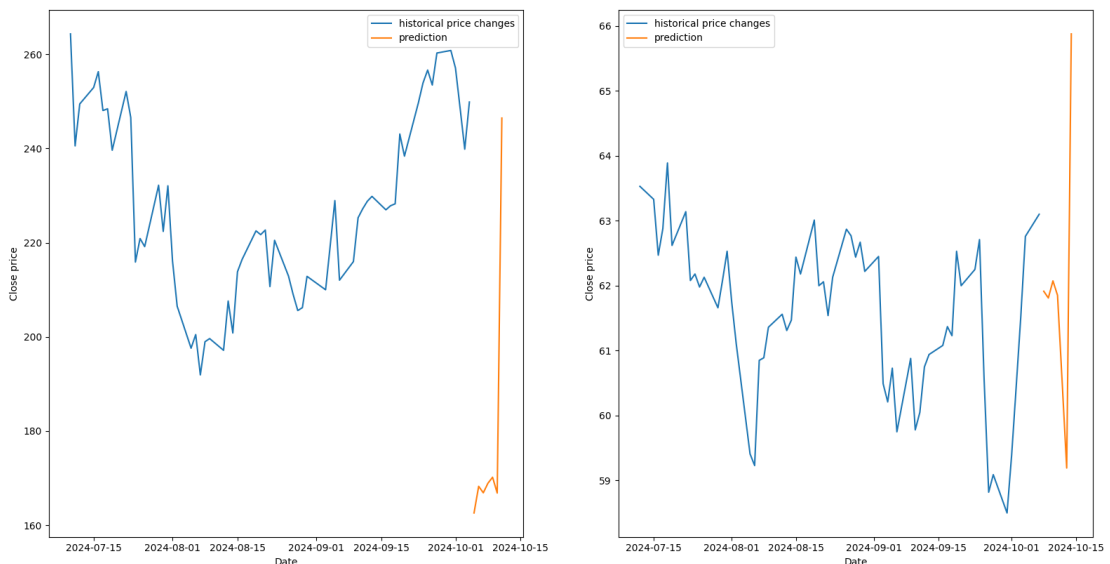


Рисунок 45. Часові ряди цін на “Tesla Motors Inc” та “Total Energies SE”.

Часові ряди на рисунку (45) не мають ярко вираженого тренду до зростання чи падіння, їх часові проміжки однакові, отже фінансові інструменти придатні для експерименту.

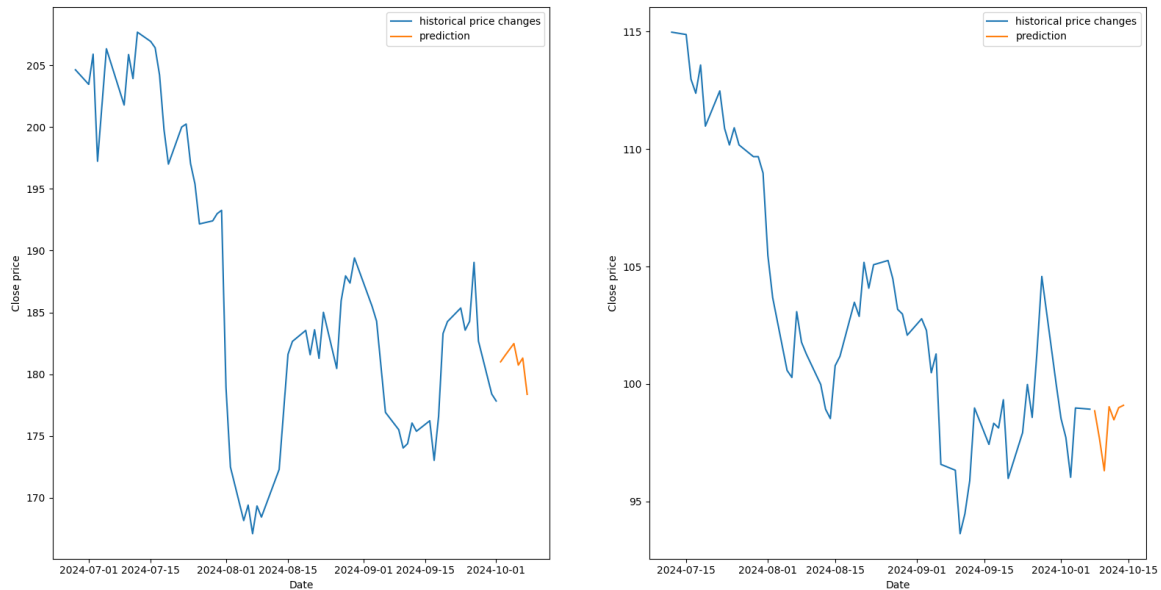


Рисунок 46. Часові ряди цін на “Toyota Motor Corp” та “Volkswagen AG”.

Часові ряди на рисунку (46) мають ярко виражений тренд до падіння, їх часові проміжки однакові, отже фінансові інструменти придатні для експерименту.

Оглянуті на рисунках (42,43,45,45 і 46) фінансові інструменти мають як схильності до зростання, так і до падіння, що і необхідно для експерименту, адже при прибутках через загальний ріст усіх інструментів, або втратах при загальному падінні усіх інструментів, хід експерименту не буде корисний для оцінки ефективності прогнозів системи керування ділінгом.

Перейдемо до експерименту.

Відітнемо від часових рядів відповідну шести тижням кількість записів, щоб імітувати проходження часу повертаючи ці запити, та закупимо фінансові інструменти для портфелю активів за консервативною рекомендацією. Далі будемо імітувати проходження тижнів, повертаючи по сім записів до часових рядів. Перевіримо зміну фінансів на балансі за півтора місяця слідування рекомендаціям системи керування. Початковий склад портфеля активів, із 10000-ним початковим балансом фінансів, можна побачити на рисунку (47).

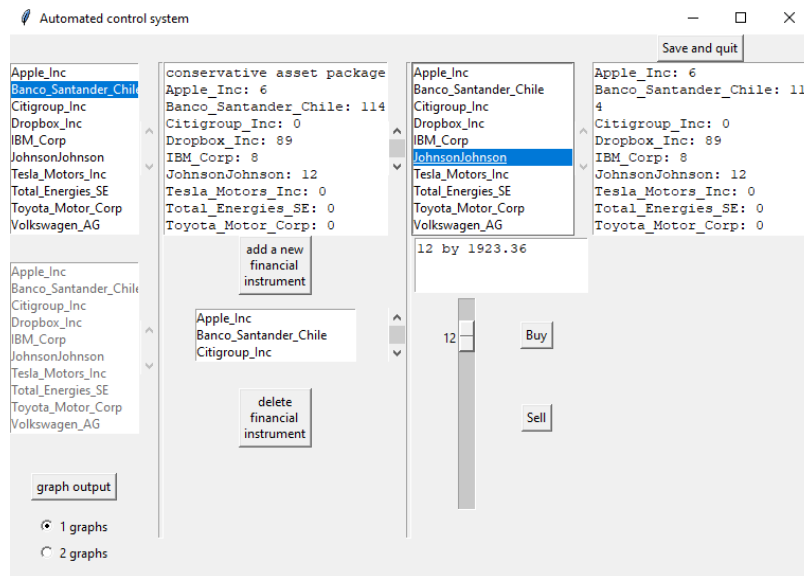


Рисунок 47. Експеримент зі складання консервативного портфеля активів, початок першого тижня.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (48).

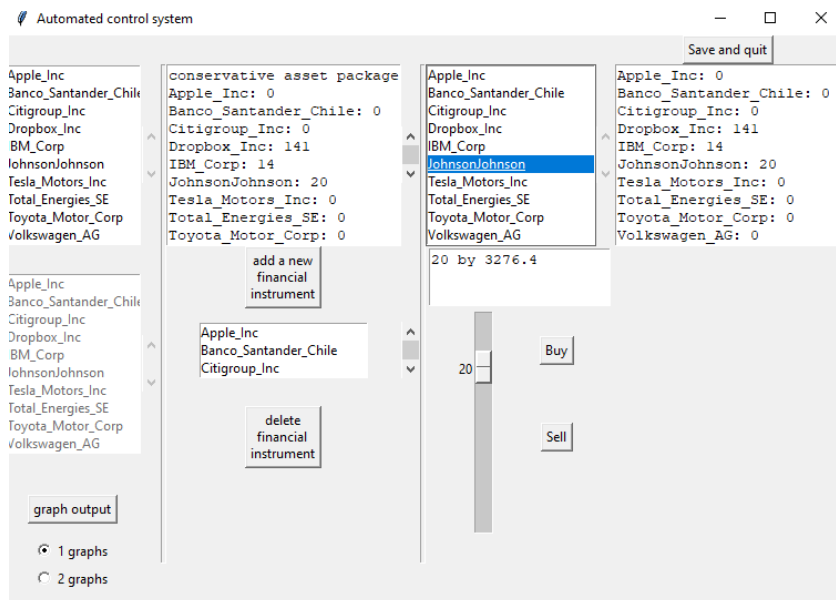


Рисунок 48. Експеримент зі складання консервативного портфеля активів, кінець першого тижня.

Попередня кількість закуплених фінансових інструментів, вкладені у них фінанси, зміна їх цін на кінець першого тижня експерименту представлені далі у таблиці (3.1).

Таблиця 3.1 Цінність вкладів на кінець першого тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	6	1332.24	+ 11.8	+ 70.8
Banco Santander Chile	114	2236.68	+ 0.45	+ 51.3
Citigroup Inc	0	0	+ 0.07	0
Dropbox Inc	89	2253.48	+ 0.97	+ 86.33
IBM Corp	8	1817.44	+ 5.92	+47.36
JohnsonJohnson	12	1914.24	+ 4.3	+4.3
Tesla Motors Inc	0	0	-22.29	0
Total Energies SE	0	0	-2.41	0
Toyota Motor Corp	0	0	-6.14	0
Volkswagen AG	0	0	-2.0	0
Тижневий прибуток	+258.64			
Загальний прибуток	+258.64			

Як видно із таблиці (3.1), усі фінанси було вкладено у зростаючі інструменти, проте найбільша частка фінансів розподілена між найменш-зростаючими інструментами. Подібна поведінка відповідає консервативній стратегії, у рамках якої потрібно мінімізувати ризики а не максимізувати прибутки.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (49).

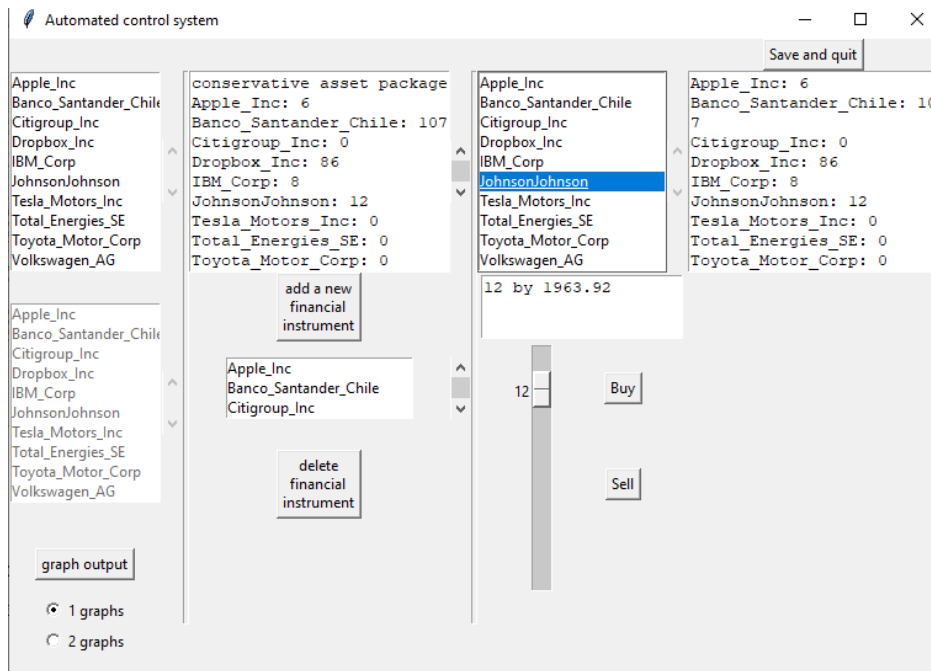


Рисунок 49. Експеримент зі складання консервативного портфеля активів, кінець другого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець другого тижня експерименту представлені далі у таблиці (3.2).

Таблиця 3.2 Цінність вкладів на кінець другого тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	0	0	-0.8	0
Banco Santander Chile	0	0	+0.46	0
Citigroup Inc	0	0	-1.12	0
Dropbox Inc	141	3688.56	-0.34	-47.94
IBM Corp	14	3272.78	-15.45	-216.3
JohnsonJohnson	20	3285.8	-0.63	-12.6
Tesla Motors Inc	0	0	+38.92	0
Total Energies SE	0	0	+0.41	0

Продовження таблиці 3.2.

0	1	2	3	4
Toyota Motor Corp	0	0	-3.71	0
Volkswagen AG	0	0	+2.35	0
Тижневий прибуток	-276.34			
Загальний прибуток	-17.75			

Як видно із таблиці (3.2), фінансові інструменти Dropbox Inc, IBM Corp та JohnsonJohnson, які показували найменше падіння ціни, впали у ціні одночасно. Ці фінансові інструменти відносяться до різних галузей, тому не могли впасти у ціні через якусь спільну причину, отже ми бачимо випадок взаємо непов'язаного падіння ціни фінансових інструментів, які здавалися найбільш стабільними тиждень тому, це можливий, хоча і неймовірний випадок. Не завжди можливо запобігти втратам навіть при намаганні уникнення ризиків. Система не може робити ідеальні рекомендації кожного разу, подивимося як вона поведе себе далі.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (50).

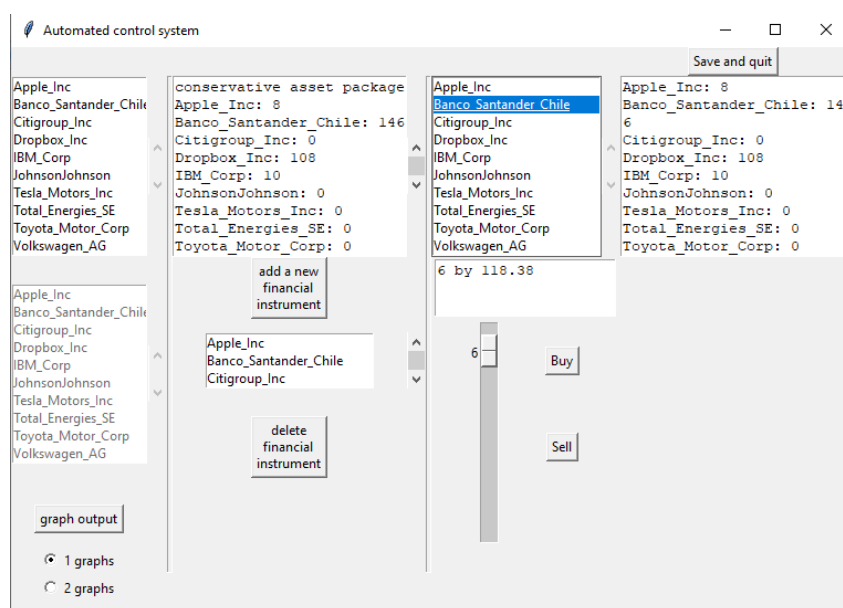


Рисунок 50. Експеримент зі складання консервативного портфеля активів, кінець третього тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець третього тижня експерименту представлені далі у таблиці (3.3).

Таблиця 3.3 Цінність вкладів на кінець третього тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	6	1388.94	-9.53	-57.18
Banco Santander Chile	107	2188.15	-0.72	-77.04
Citigroup Inc	0	0	+ 0.64	0
Dropbox Inc	86	2236.0	+ 0.22	+18.92
IBM Corp	8	1718.16	-8.51	-68.08
JohnsonJohnson	12	1929.12	-2.58	-30.96
Tesla Motors Inc	0	0	-26.42	0
Total Energies SE	0	0	-1.34	0
Toyota Motor Corp	0	0	+ 2.35	0
Volkswagen AG	0	0	-5.25	0
Тижневий прибуток	-214.34			
Загальний прибуток	-316.54			

І як видно із таблиці (3.3), серед порад системи був лише один зростаючий інструмент, і загальний прибуток продовжує уходити у мінус. Це поганий результат, для консервативного портфеля система не може помітити різку зміну тренду ряду фінансових інструментів, та швидко перелаштуватися на інші. Проте це лише за три періоди вкладів, подивимося чи зможе система перекрити втрати за решту відведеного на експеримент часу.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (51).

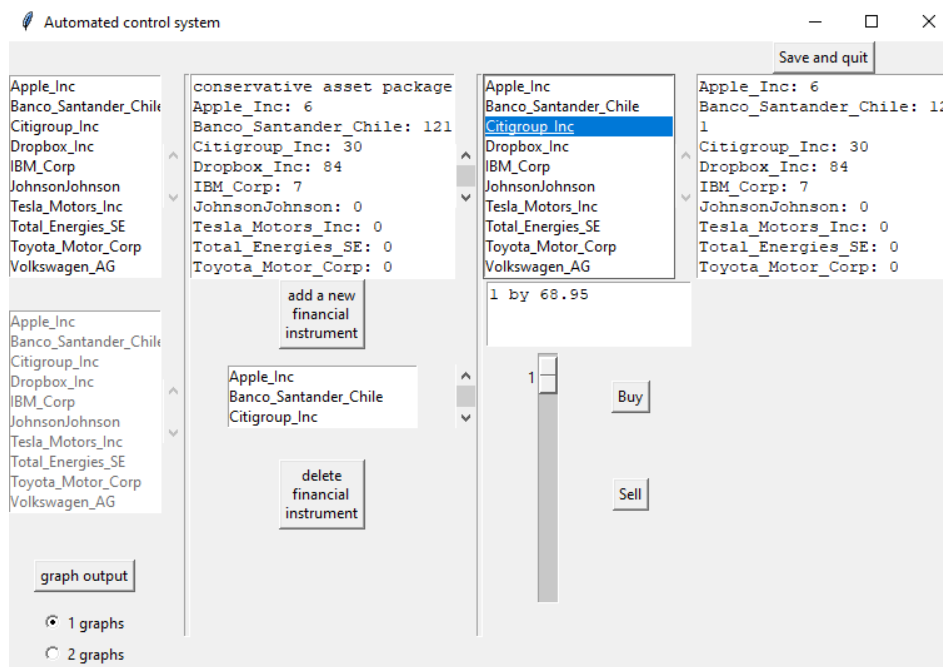


Рисунок 51. Експеримент зі складання консервативного портфеля активів, кінець четвертого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець четвертого тижня експерименту представлені далі у таблиці (3.4).

Таблиця 3.4 Цінність вкладів на кінець четвертого тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	8	1783.28	+1.97	+15.76
Banco Santander Chile	146	2876.2	-0.32	-46.72
Citigroup Inc	0	0	+5.42	0
Dropbox Inc	108	2879.28	+1.28	+138.24
IBM Corp	10	2070.2	+3.55	+35.5
JohnsonJohnson	0	0	-5.05	0
Tesla Motors Inc	0	0	+77.45	0

Продовження таблиці 3.4.

Total Energies SE	0	0	-0.33	0
Toyota Motor Corp	0	0	-0.44	0
Volkswagen AG	0	0	-0.11	0
Тижневий прибуток	+142.78			
Загальний прибуток	-115.44			

На таблиці (3.4) видно, як прибутки компенсують втрати за даний тиждень, що є прикладом успішного випадку консервативної стратегії. Загальний прибуток досі має від'ємне значення, проте порівняно із попереднім тижнем ситуація покращилася.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (52).

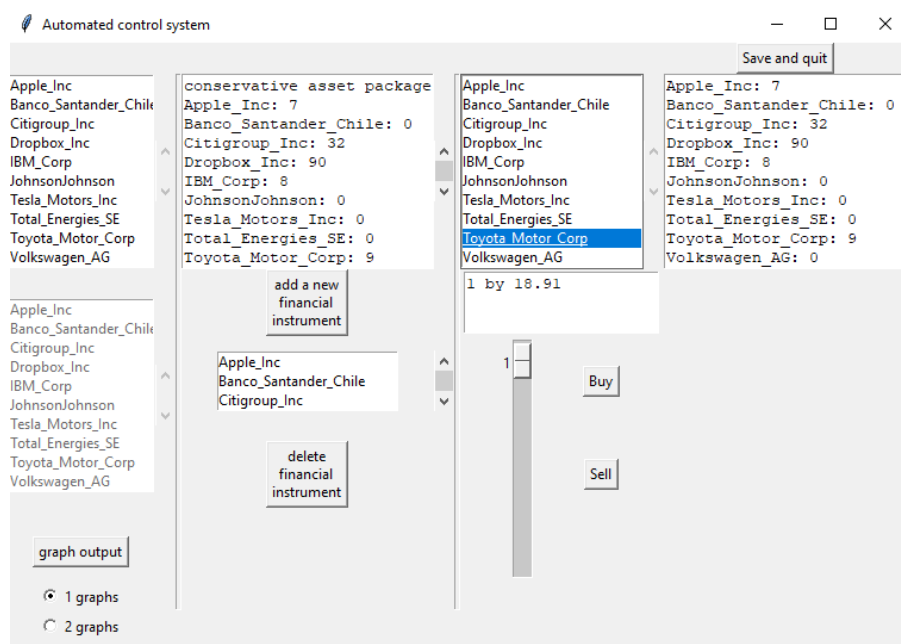


Рисунок 52. Експеримент зі складання консервативного портфеля активів, кінець п'ятого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець п'ятого тижня експерименту представлені далі у таблиці (3.5).

Таблиця 3.5 Цінність вкладів на кінець п'ятого тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	6	1369.5	+1.54	+9.24
Banco Santander Chile	121	2308.68	-0.17	-20.57
Citigroup Inc	30	2041.5	+1.91	+57.3
Dropbox Inc	84	2320.92	+0.3	+30.24
IBM Corp	7	1459.43	+14.53	+101.71
JohnsonJohnson	0	0	+3.31	0
Tesla Motors Inc	0	0	+41.77	0
Total Energies SE	0	0	-1.58	0
Toyota Motor Corp	0	0	+0.93	0
Volkswagen AG	0	0	-1.78	0
Тижневий прибуток	+177.92			
Загальний прибуток	-21.2			

Із таблиці (3.5) видно, що система переважно продовжує рекомендувати вкладатися у фінансові інструменти, які стабільно ростуть. Втрати за період від другого до четвертого тижня майже перекриті, загальні прибутки майже вийшли у плюс.

Тепер, здійснимо продаж усіх запасів наявних фінансових інструментів, та подивимося на загальний баланс фінансів на рисунку (53).

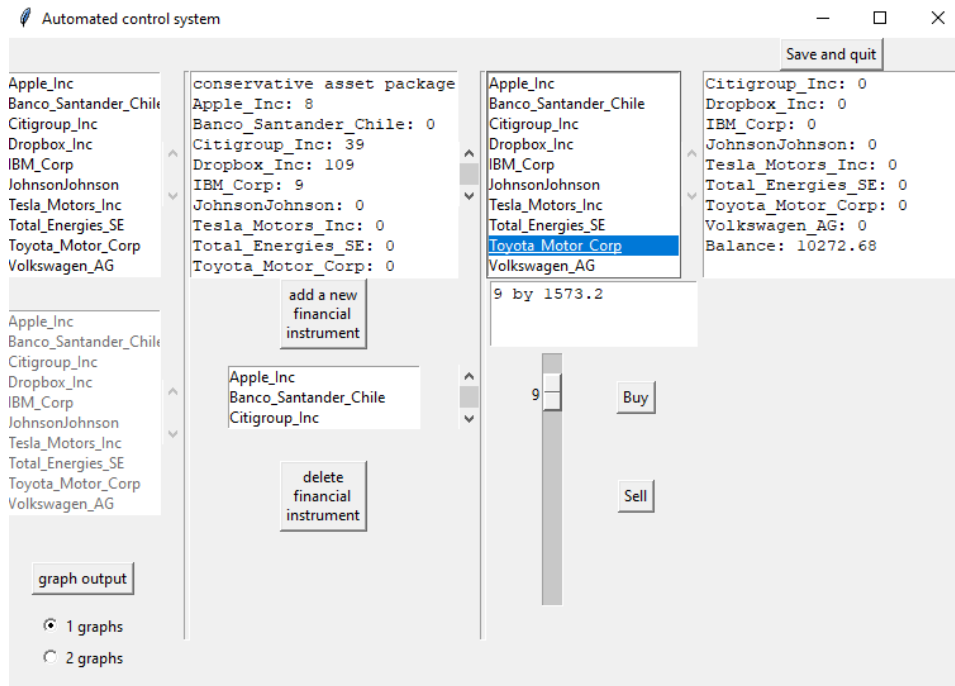


Рисунок 53. Експеримент зі складання консервативного портфеля активів, кінець шостого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець шостого тижня експерименту представлені далі у таблиці (3.6).

Таблиця 3.6 Цінність вкладів на кінець шостого тижня експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	7	1629.04	+ 10.19	+ 71.33
Banco Santander Chile	0	0	-0.04	0
Citigroup Inc	32	2270.4	+ 0.59	+ 18.88
Dropbox Inc	90	2564.1	+ 0.19	+ 17.1
IBM Corp	8	1806.8	+ 7.69	+61.59
JohnsonJohnson	0	0	-5.35	0
Tesla Motors Inc	0	0	+ 15.17	0
Total Energies SE	0	0	-1.17	0

Продовження таблиці 3.6.

Toyota Motor Corp	9	1582.11	-0.99	-8.91
Volkswagen AG	0	0	+1.88	0
Тижневий прибуток	+159.92			
Загальний прибуток	+272.68			

Із таблиці (3.6) видно, що портфель активів успішно вивів загальні прибутки у плюс.

Підсумуємо дані із таблиць (3.1-3.6). Система двічі схилила із рекомендаціями щодо самих найменш-падаючих у ціні фінансових інструментів, проте компенсувала втрати прибутками від успішних випадків. Система не ідеально реалізує консервативну стратегію, проте придатна до використання.

Експеримент починався із 10000 фінансів на балансі, а закінчився із 10272.68 фінансів на балансі. Результуючий баланс більший за початковий на 2.45%, це хороший результат.

Потрібно відзначити, що баланс портфеля поніс значні втрати через неочікуване одночасне падіння ціни Dropbox Inc, IBM Corp та JohnsonJohnson, у які система продовжувала рекомендувати робити вклади, хоча результуючі прибутки перекрили ці втрати. Пара випадків одночасного падіння цін переважної кількості інструментів, у які система рекомендувала вкладати фінанси, також пов'язані з обмеженою кількістю фінансових інструментів у портфелі. При використанні більшої кількості, така проблема повинна зникнути, або виникати значно рідше.

Перевіримо як змінювалася ціна на самий збитковий фінансовий інструмент IBM Corp. Графік історії змін його ціни приведений на рисунку (54).

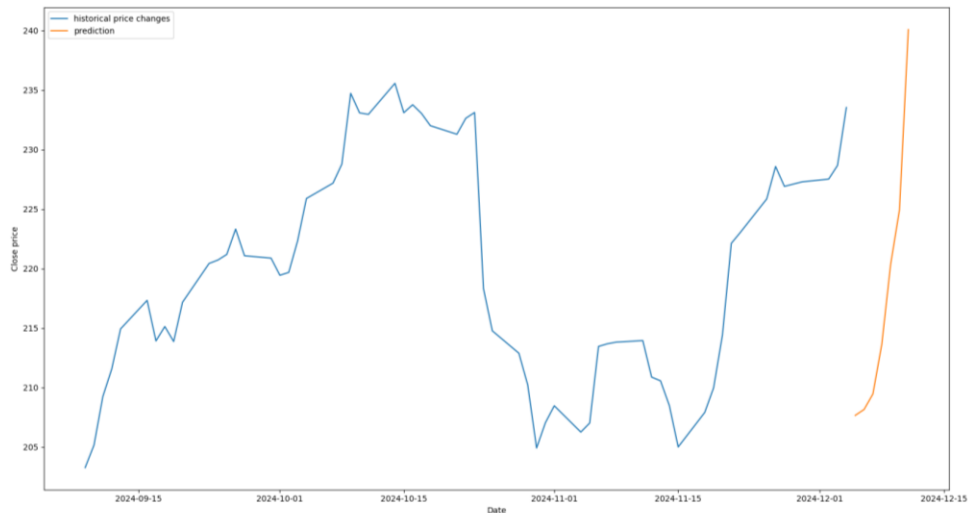


Рисунок 54. Часовий ряд цін на “IBM Corp”.

Збиткові третій і четвертий тижні припадають саме на два провали у ціні, які показані на рисунку (54). Ні до, ні після цих двох провалів фінансовий інструмент не демонструє схильності до різкого падіння ціни, передбачити таке було складно.

Тепер проведемо аналогічний експеримент для рекомендацій по складанню агресивного портфеля активів. Початковий запас фінансів 10000 на балансі. Далі на рисунку (55) представлено початково складений агресивний портфель активів.

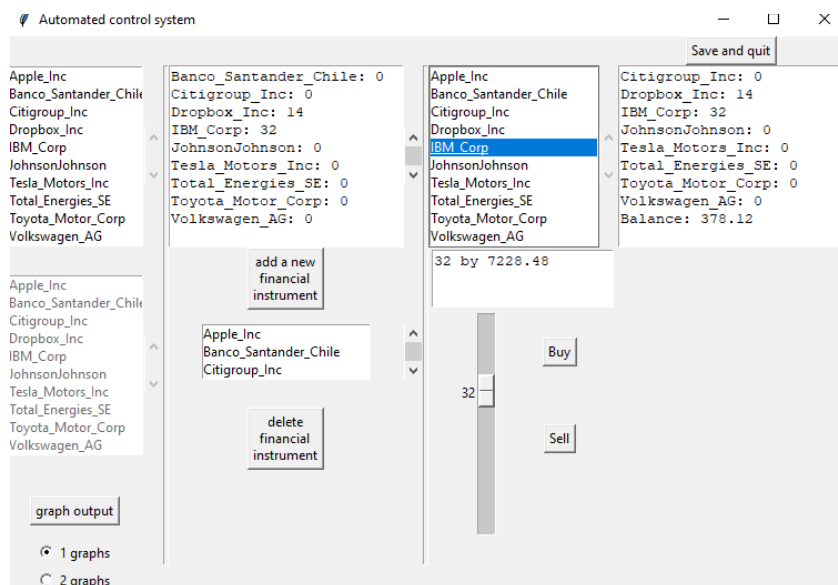


Рисунок 55. Експеримент зі складання агресивного портфеля активів, початок першого тижня.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (56).

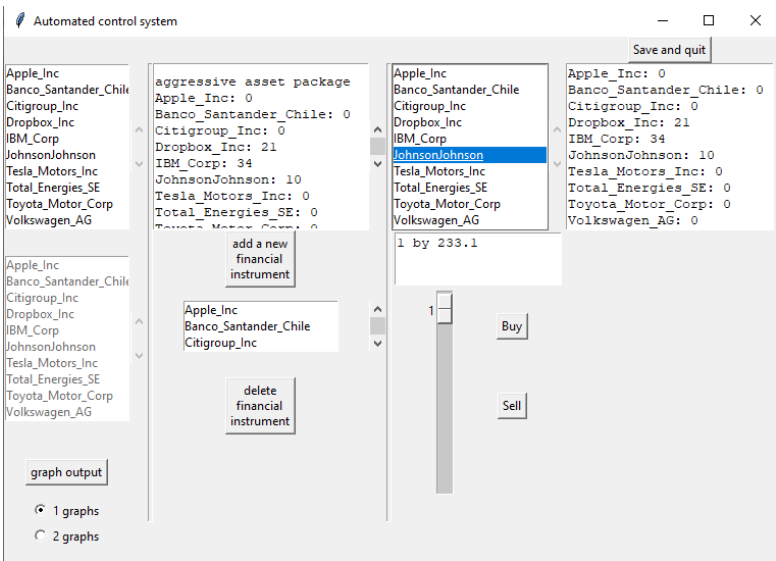


Рисунок 56. Експеримент зі складання агресивного портфеля активів, кінець першого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець першого тижня другого експерименту представлені далі у таблиці (3.7).

Таблиця 3.7 Цінність вкладів на кінець першого тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	9	1998.36	+ 11.8	+ 106.2
Banco Santander Chile	0	0	+ 0.45	0
Citigroup Inc	0	0	+ 0.07	0
Dropbox Inc	14	354.48	+ 0.97	+ 13.58
IBM Corp	32	7269.76	+ 5.92	+ 189.44
JohnsonJohnson	0	0	+ 4.3	0
Tesla Motors Inc	0	0	-22.29	0

Продовження таблиці 3.7.

Total Energies SE	0	0	-2.41	0
Toyota Motor Corp	0	0	-6.14	0
Volkswagen AG	0	0	-2.0	0
Тижневий прибуток	+309.94			
Загальний прибуток	+309.94			

Як видно із таблиці (3.7), система правильно спрогнозувала ріст та падіння трьох фінансових інструментів. Для агресивного портфеля система рекомендувала вкласти більше фінансів у ті фінансові інструменти, які показали більшу зміну ціни, на відмінну від консервативного, який зосереджувався на найменш падаючих інструментах, що призвело до більших прибутків, ніж на таблиці (3.1). Також потрібно відзначити, що перша ітерація складання агресивного портфеля активів призвела до більшого прибутку, ніж консервативний портфель на кінець першого експерименту.

Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (57).

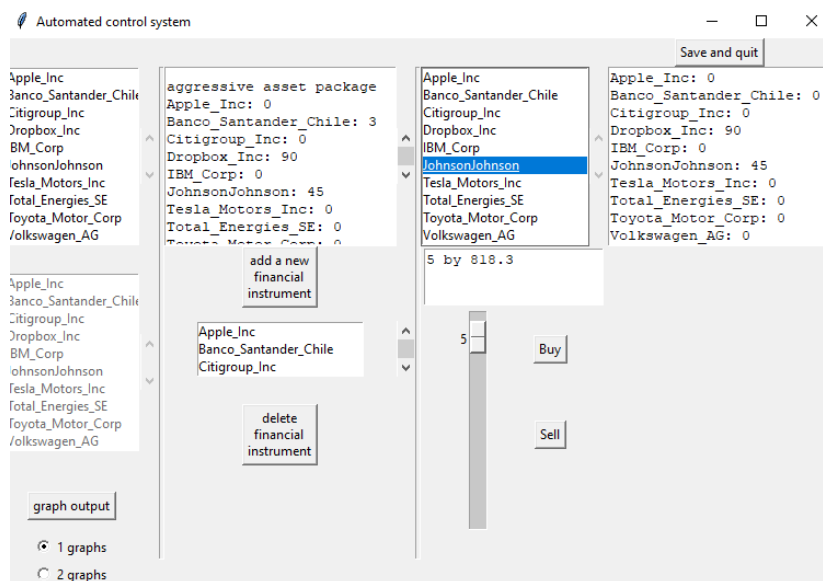


Рисунок 57. Експеримент зі складання агресивного портфеля активів, кінець другого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець другого тижня другого експерименту представлені далі у таблиці (3.8).

Таблиця 3.8 Цінність вкладів на кінець другого тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	0	0	-0.8	0
Banco Santander Chile	0	0	+ 0.46	0
Citigroup Inc	0	0	-1.12	0
Dropbox Inc	21	549.36	-0.34	-7.14
IBM Corp	34	7948.18	-15.45	-525.3
JohnsonJohnson	10	1642.9	-0.63	-6.3
Tesla Motors Inc	0	0	+ 38.92	0
Total Energies SE	0	0	+0.41	0
Toyota Motor Corp	0	0	-3.71	0
Volkswagen AG	0	0	+ 2.35	0
Тижневий прибуток	-538.74			
Загальний прибуток	-204.05			

Порівнюючи дані із таблиці (3.8) із даними попереднього експерименту за цей тиждень у таблиці (3.2), агресивний портфель поніс більші втрати, що співвідноситься і тим, що агресивна стратегія більш ризикована. Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (58).

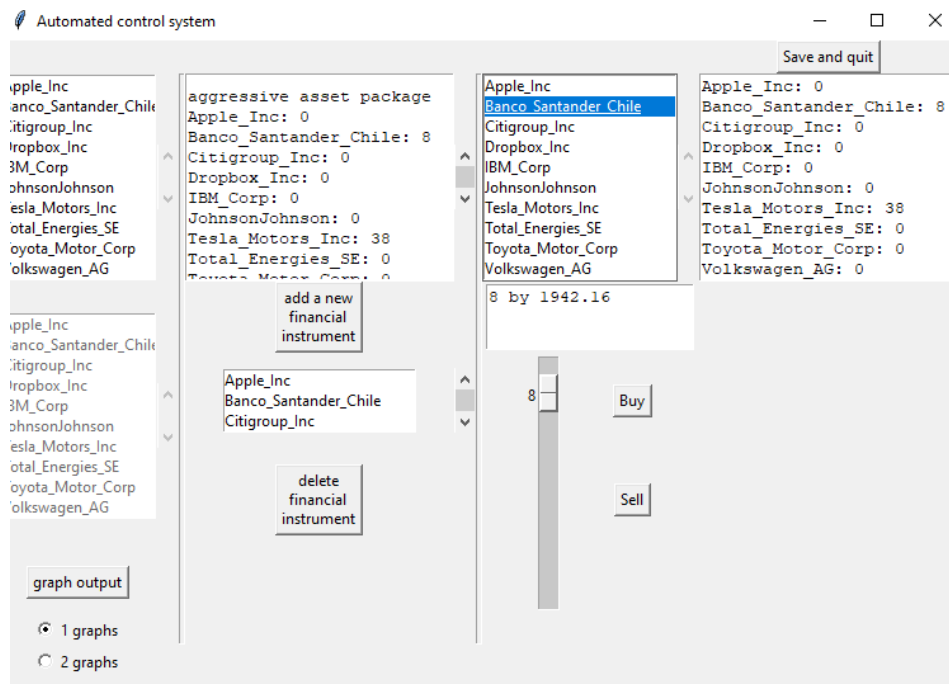


Рисунок 58. Експеримент зі складання агресивного портфеля активів, кінець третього тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець третього тижня другого експерименту представлені далі у таблиці (3.9).

Таблиця 3.9 Цінність вкладів на кінець третього тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	0	0	-9.53	0
Banco Santander Chile	0	0	-0.72	0
Citigroup Inc	0	0	+ 0.64	0
Dropbox Inc	90	2340.0	+ 0.22	+19.8
IBM Corp	0	0	-8.51	0
JohnsonJohnson	45	7234.2	-2.58	-116.1
Tesla Motors Inc	0	0	-26.42	0
Total Energies SE	0	0	-1.34	0

Продовження таблиці 3.9.

Toyota Motor Corp	0	0	+ 2.35	0
Volkswagen AG	0	0	-5.25	0
Тижневий прибуток	-96.3			
Загальний прибуток	-414.65			

Як видно із таблиці (3.9), система досі рекомендувала вкласти фінанси у JohnsonJohnson, ціна на який падає, проте вона передбачила падіння IBM Corp, який приніс найбільше втрат у першому експерименті. Перейдемо до наступного тижня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (59).

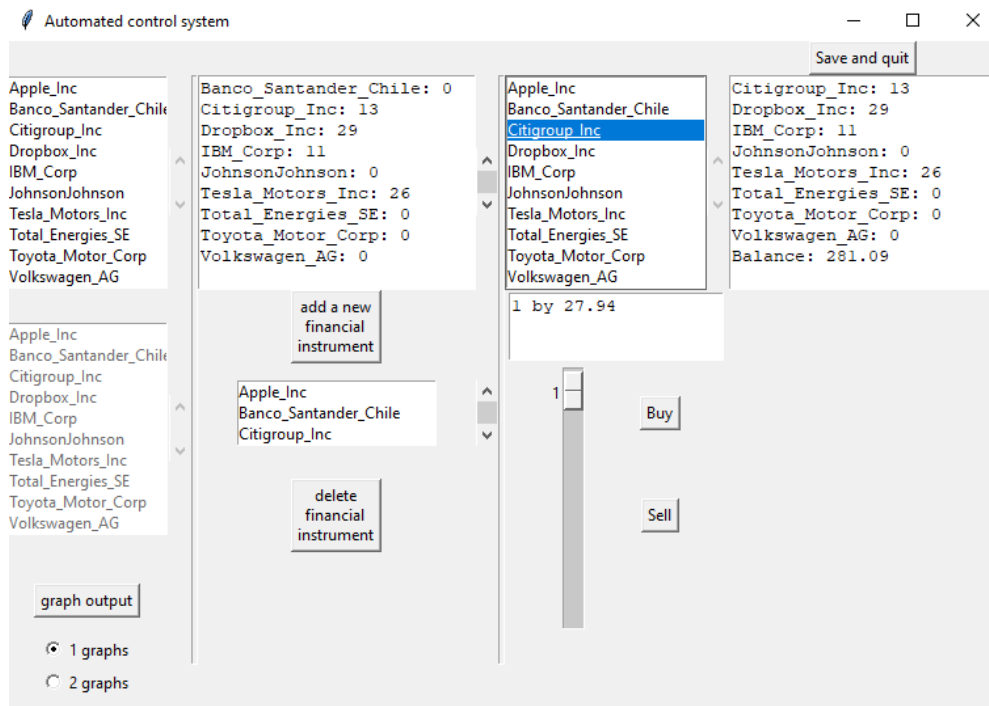


Рисунок 59. Експеримент зі складання агресивного портфеля активів, кінець четвертого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на п'ятий день другого експерименту представлені далі у таблиці (3.10).

Таблиця 3.10 Цінність вкладів на кінець четвертого тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	0	0	+1.97	0
Banco Santander Chile	8	157.6	-0.32	-2.56
Citigroup Inc	0	0	+5.42	0
Dropbox Inc	0	0	+1.28	0
IBM Corp	0	0	+3.55	0
JohnsonJohnson	0	0	-5.05	0
Tesla Motors Inc	38	9554.72	+77.45	+2943.1
Total Energies SE	0	0	-0.33	0
Toyota Motor Corp	0	0	-0.44	0
Volkswagen AG	0	0	-0.11	0
Тижневий прибуток	+2940.54			
Загальний прибуток	+2855.11			

Випадок показаний на таблиці (3.10) є ідеальним прикладом успішного випадку агресивної стратегії, вкладання майже усіх фінансів у один фінансовий інструмент, який як і передбачалося зріс, що призвело до загального прибутку в 2855.11 одиниць. Проте, якби прогноз не справдився, слідування такій рекомендації принесло б величезні втрати. Прибуток на кінець четвертого тижня переважає прибутки від консервативного портфеля активів на порядок.

Перейдемо до наступного дня і складемо заново портфель активів, згідно нових рекомендацій на рисунку (60).

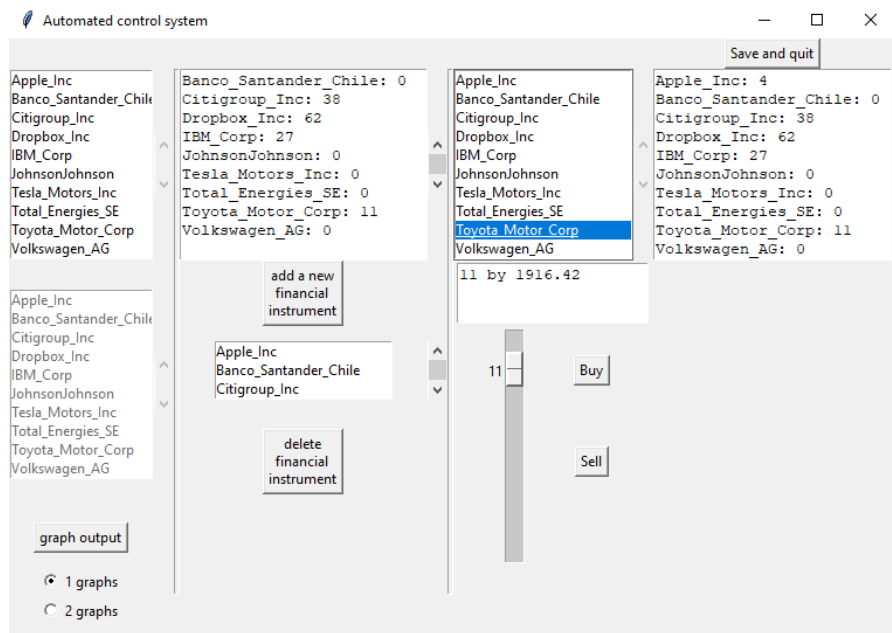


Рисунок 60. Експеримент зі складання агресивного портфеля активів, кінець п'ятого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець п'ятого тижня другого експерименту представлені далі у таблиці (3.11).

Таблиця 3.11 Цінність вкладів на кінець п'ятого тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	0	0	+1.54	0
Banco Santander Chile	0	0	-0.17	0
Citigroup Inc	13	884.65	+1.91	+24.83
Dropbox Inc	29	801.27	+0.3	+10.44
IBM Corp	11	2293.39	+14.53	+159.83
JohnsonJohnson	0	0	+3.31	0
Tesla Motors Inc	26	8072.74	+41.77	+1086.02
Total Energies SE	0	0	-1.58	0

Продовження таблиці 3.11.

Toyota Motor Corp	0	0	+0.93	0
Volkswagen AG	0	0	-1.78	0
Тижневий прибуток	+1281.12			
Загальний прибуток	+3614.26			

Як видно у таблиці (3.11), система успішно передбачила ріст кількох фінансових інструментів, і розподіливши фінанси між ними вона вклала менше фінансів у самий вигідний Tesla Motors Inc. Такий розклад зменшив рівень прибутків порівняно із тижнем відповідно до таблиці (3.10), проте зменшило і ризику. Система пропонує самий ризикований варіант поведінки, вкладання тільки у один/два фінансові інструменти тільки якщо прогнозується ріст тільки настільки малої кількості інструментів.

Тепер, , перейдемо до наступного дня і здійснимо продаж усіх запасів наявних фінансових інструментів, та подивимося на загальний баланс фінансів на рисунку (61).

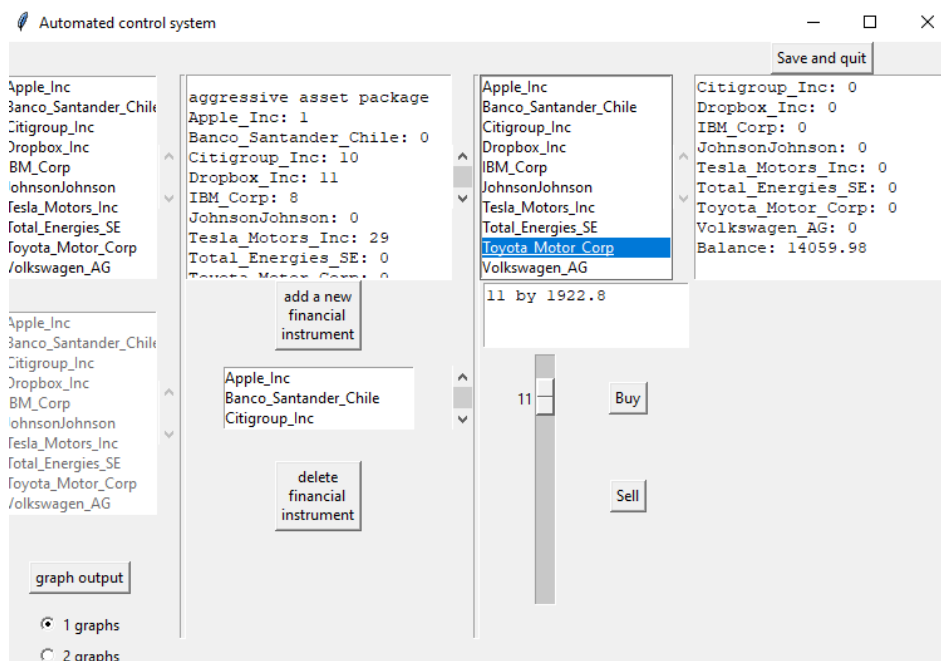


Рисунок 61. Експеримент зі складання агресивного портфеля активів, кінець шостого тижня.

Попередня цінність закуплених фінансових інструментів, та її зміна на кінець шостого тижня другого експерименту представлені далі у таблиці (3.12).

Таблиця 3.12 Цінність вкладів на кінець шостого тижня другого експерименту.

Назва	Кількість закуплених минулого тижня акцій	Вкладені фінанси	Зміна ціни	Прибуток/втрати
Apple Inc	4	930.88	+ 10.19	+ 40.76
Banco Santander Chile	0	0	-0.04	0
Citigroup Inc	38	2696.1	+ 0.59	+ 22.42
Dropbox Inc	62	1766.38	+ 0.19	+ 11.78
IBM Corp	27	6097.95	+ 7.69	+207.63
JohnsonJohnson	0	0	-5.35	0
Tesla Motors Inc	0	0	+ 15.17	0
Total Energies SE	0	0	-1.17	0
Toyota Motor Corp	11	1933.69	-0.99	-10.88
Volkswagen AG	0	0	+1.88	0
Тижневий прибуток	+ 271.7			
Загальний прибуток	+ 4059.98			

Порівнявши рисунки (61) і (53), можна побачити, що слідування рекомендаціям системи керування по складанню агресивного портфеля активів принесло значно більше коштів (загальний баланс фінансів зріс на 40.59%), ніж слідування рекомендаціям по складанню консервативного портфеля (загальний баланс фінансів зріс на 2.72%). Оскільки моделі прогнозу цін використовувались лише при формуванні рекомендацій на агресивний портфель активів, можна зробити висновок про доцільність використання розробленої моделі

передбачення ціни за даними часових рядів. Необхідно зауважити, як показано на таблиці (3.8), модель не завжди робить влучні прогнози, тому сліпе слідування прогнозам, по яким складається агресивний портфель не гарантує переваги прибутків над втратами, проте використання рекомендацій розробленого програмного забезпечення доцільне як допоміжний метод складання портфеля активів.

3.4 Висновки по розділу

У цьому розділі виконано завдання розробки та тестування програмного забезпечення, яке є результатом розробки автоматизованої системи керування ділінгом для трейдингового відділу банку. У розділі наведено UML діаграму діяльності програмного забезпечення та наведено його технологічні вимоги. Також у розділі було наведено опис інтерфейсу ПО, та проведено тестування усіх закладених у програмне забезпечення функцій, по результатам якого весь функціонал програмного забезпечення було визнано працездатним у повному обсязі. Крім того, у рамках тестування придатності ПО формування портфеля активів та його ефективності у цьому, у розділі було проведено експериментальна перевірка, по результатам якої ПО показало результати у збільшенні фінансів у 2.72% і 40.59% для консервативного і агресивного портфелів. У наслідок експерименту ПО було визнано придатним для використання, як допоміжний метод складання портфеля активів.

ВИСНОВКИ

У даній роботі було розглянуто процес фондового ділінгу та описана структура програмного забезпечення для покращення його автоматизації, а саме розподілу фінансів між фінансовими інструментами у портфелі активів.

У першому розділі були виконані завдання дослідження об'єкту автоматизації, дослідження конкурентів, для визначення напрямку дослідження роботи у визначенні цінності фінансових інструментів, та пошук і опис актуальних методів для визначення цінності фінансових інструментів. У процесі їх виконання було побудовано структурну схему об'єкта автоматизації, проведено огляд аналогів систем керування фондовим ділінгом і визначено прогнозування ціни доцільним напрямком, на якому варто зосередити дослідження. Також у розділі описані актуальні методи для визначення цінності фінансових інструментів, тобто методи для прогнозування цін.

Другий розділ даної роботи було присвячено дослідженню методів прогнозування, відповідно для виконання завдання дослідження принципів роботи описаних актуальних методів. За результатами дослідження було відфільтровано усі непридатні для розглядаємої задачі моделі-прогнозування. Із працездатних моделей, була обрана навчена методом "ARIMA" модель, яка була використана у розробці автоматизованої системи керування ділінгом для трейдингового відділу банку.

У третьому розділі було виконано завдання розробки та тестування програмного забезпечення. У розділі було приведено UML діаграму діяльності програмного забезпечення, технологічні вимоги для його працездатності, та результати його тестування. Всі функції, які автор хотів вкласти у розроблену систему керування, були визнані працездатними. Експериментально встановлено, що за допомогою розробленого програмного забезпечення можна збільшити фінанси на 2.72% - 40.59% за півтора місяця слідування рекомендаціям системи керування. Прогнозування цін придатне для формування портфеля активів.

Список літератури

1. Мірошниченко Т. В. Порівняння MetaTrader 4 та MetaTrader 5. *prikhodko*. URL: <https://prikhodko.com.ua/poslugy/finteh-uk/platizhni-licenziyi/porivnyannya-metatrader-4-ta-metatrader-5/> (дата звернення: 09.09.2024).
2. MetaQuotes Ltd Company. *metaquotes*. URL: <https://www.metaquotes.net/en/company> (дата звернення: 10.09.2024).
3. Bloomberg Terminal. *bloomberg*. URL: <https://www.bloomberg.com/professional/support/customer-support/> (дата звернення: 11.09.2024).
4. Zhu Y., Zhou G. Technical analysis: An asset allocation perspective on the use of moving averages☆. *Journal of Financial Economics*. 2009. Vol. 92, no. 3. P. 519–544. URL: <https://doi.org/10.1016/j.jfineco.2008.07.002> (дата звернення: 12.09.2024).
5. Panigrahi A. K., Vachhani K., Chaudhury S. K. Trend identification with the relative strength index (RSI) technical indicator –A conceptual study. *Journal of Management Research and Analysis*. 2021. Vol. 8, no. 4. P. 159–169. URL: <https://doi.org/10.18231/j.jmra.2021.033> (дата звернення: 13.09.2024).
6. Antonio Agudelo Aguirre A., Alfredo Rojas Medina R., Darío Duque Méndez N. Machine learning applied in the stock market through the Moving Average Convergence Divergence (MACD) indicator. *Investment Management and Financial Innovations*. 2020. Vol. 17, no. 4. P. 44–60. URL: [https://doi.org/10.21511/imfi.17\(4\).2020.05](https://doi.org/10.21511/imfi.17(4).2020.05) (дата звернення: 14.09.2024).
7. Americas Trading Conference 2024. *FIX Trading Community v2.0*. URL: <https://www.fixtrading.org/what-is-fix/> (дата звернення: 15.09.2024).
8. Parzen E. An Approach to Time Series Analysis. *The Annals of Mathematical Statistics*. 1961. Vol. 32, no. 4. P. 951–989. URL: <https://doi.org/10.1214/aoms/1177704840> (дата звернення: 16.09.2024).

9. Li Y., Pan Y. A novel ensemble deep learning model for stock prediction based on stock prices and news. *International Journal of Data Science and Analytics*. 2021. URL: <https://doi.org/10.1007/s41060-021-00279-9> (дата звернення: 16.09.2024).
10. Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy / H. Yang et al. *SSRN Electronic Journal*. 2020. URL: <https://doi.org/10.2139/ssrn.3690996> (дата звернення: 16.09.2024).
11. Singh R., Srivastava S. Stock prediction using deep learning. *Multimedia Tools and Applications*. 2016. Vol. 76, no. 18. P. 18569–18584. URL: <https://doi.org/10.1007/s11042-016-4159-7> (дата звернення: 16.09.2024).
12. Arora P., Deepali, Varshney S. Analysis of K-Means and K-Medoids Algorithm For Big Data. *Procedia Computer Science*. 2016. Vol. 78. P. 507–512. URL: <https://doi.org/10.1016/j.procs.2016.02.095> (дата звернення: 16.09.2024).
13. Hayes A. Autoregressive Integrated Moving Average (ARIMA) Prediction Model. *Investopedia*. URL: <https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp> (дата звернення: 17.09.2024).
14. Авторегресія - що це таке, визначення та поняття. *economy-pedia*. URL: <https://uk.economy-pedia.com/11036019-autoregression> (дата звернення: 17.09.2024).
15. Довідник по Machine Learning – Mean Square Error (MSE) | База знань IT. *База знань IT технологій*. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/mean-square-error-mse> (дата звернення: 18.09.2024).
16. Довідник по Machine Learning – Mean Absolute Error | База знань IT. *База знань IT технологій*. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/mean-absolute-error> (дата звернення: 18.09.2024).

- 17.Махым Z. Аналіз дерева рішень (Decision Tree Analysis). *Махым Zosym*.
URL: <https://www.maxzosim.com/decision-tree-analysis/> (дата звернення: 19.09.2024).
- 18.Довідник по Machine Learning – Random Forests | База знань ІТ. *База знань ІТ технологій*. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/random-forests> (дата звернення: 19.09.2024).
- 19.On Training Efficiency and Computational Costs of a Feed Forward Neural Network: A Review / A. Laudani et al. *Computational Intelligence and Neuroscience*. 2015. Vol. 2015. P. 1–13.
URL: <https://doi.org/10.1155/2015/818243> (дата звернення: 20.09.2024).
- 20.Довідник по Machine Learning – Recurrent Neural Network | База знань ІТ. *База знань ІТ технологій*. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/recurrent-neural-network> (дата звернення: 20.09.2024).
- 21.GeeksforGeeks. What is LSTM - Long Short Term Memory? - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/> (дата звернення: 21.09.2024).
- 22.Вивчення градієнтного спуску та SGD: ключові алгоритми для оптимізації машинного навчання. Online Coding Bootcamps | Code Labs Academy. URL: <https://codelabsacademy.com/uk/blog/gradient-descent-and-stochastic-gradient-descent-in-machine-learning> (дата звернення: 22.09.2024).
- 23.Встановлення модулів Python. Python documentation. URL: <https://docs.python.org/uk/3/installing/index.html> (дата звернення: 25.11.2024).
- 24.Gilmullin T. GitHub - Tim55667757/MT4ForexParser: Read forex data in MetaTrader 4 .hst-format and convert into .csv file and pandas dataframe. GitHub. URL: <https://github.com/Tim55667757/MT4ForexParser> (date of access: 25.11.2024).

- 25.pandas.DataFrame – pandas 2.2.3 documentation. pandas - Python Data Analysis Library.
URL: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html> (date of access: 25.11.2024).
- 26.Bondarenko O. Pandas: управління даними проєкту (2). Medium.
URL: <https://medium.com/stinopys/pandas-управління-даними-проєкту-2-e1cf6cbc3ee9> (дата звернення: 25.11.2024).
- 27.DecisionTreeRegressor. scikit-learn. URL: <https://scikit-learn.org/dev/modules/generated/sklearn.tree.DecisionTreeRegressor.html> (date of access: 25.11.2024).
- 28.pmdarima.arima.AutoARIMA – pmdarima 2.0.4 documentation. alkaline-ml - Distilling machine learning into its basics. URL: <https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html> (date of access: 25.11.2024).
- 29.Keras documentation: The Sequential model. Keras: Deep Learning for humans. URL: https://keras.io/guides/sequential_model/ (date of access: 25.11.2024).
- 30.Tutorials | TensorFlow Core. TensorFlow. URL: <https://www.tensorflow.org/tutorials> (date of access: 25.11.2024).
- 31.tkinter Python interface to Tcl/Tk. Python documentation. URL: <https://docs.python.org/uk/3/library/tkinter.html> (date of access: 25.11.2024).
- 32.Tkinter python: Як використовувати. FoxmindEd. URL: <https://foxminded.ua/tkinter-python-yak-vykorystovuvaty/> (дата звернення: 25.11.2024).
- 33.Paul Miskew. PYTHON3 TKINTER - TEXTBOX ADDING TEXT, 2018. YouTube. URL: <https://www.youtube.com/watch?v=VBuFaRPxRSc> (date of access: 25.11.2024).

34. Tkinter. kievoit.

URL: <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/tkinter/index.html#Frame> (date of access: 25.11.2024).

35. Python – просто! - grid. Google Drive: Sign-in.

URL: <https://sites.google.com/comp-sc.if.ua/python-easy/tkinter/пакувальники/grid> (дата звернення: 25.11.2024).

36. Зубковський Є. Інвестиційний портфель: що це таке, типи, як скласти. Створюємо капітал для отримання пасивного доходу, Е. Зубковський. URL: <https://e-zubkovskiy.com/blog/shcho-take-investitsiyniy-portfel#4> (дата звернення: 25.11.2024).

37. Червінський Р., Кабачій В. ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДО ПРОГНОЗУВАННЯ ФІНАНСОВИХ ЧАСОВИХ РЯДІВ ВНТУ електронні наукові видання. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20582/17133>

38. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» (кафедра комп'ютерних систем управління) [Електронний ресурс] / уклад. В. М. Дубовой. – Вінниця: ВНТУ, 2023 – (PDF, 45 с.).

39. Ділінг: що означає термін простими словами. Рейтинги банківських послуг України для населення в 2024. URL: https://bankchart.com.ua/spravochniki/finansovyy_slovar/id/130 (дата звернення: 01.12.2024).

ДОДАТКИ

Додаток Б

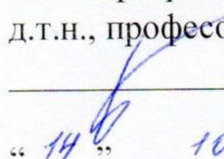
(обов'язковий)

ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., професор


В'ячеслав КОВТУН

“ 14 ” 10 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

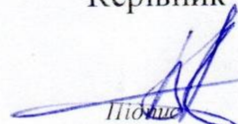
Автоматизована система керуванням фондовим ділінгом для трейдингового
відділу банку.

08-33.МКР.011.00.000 ТЗ

Студент групи _ЗАКІТР-23м_


ПідписРостислав ЧЕРВІНСЬКИЙ
Ім'я ПРІЗВИЩЕ

Керівник _____


ПідписВладислав КАБАЧІЙ
Ім'я ПРІЗВИЩЕ

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Автоматизована система керування фондовим ділінгом.

1.2. Галузь застосування – фондовий ділінг.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ №310 від 17-09-2024р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є розробка програмного забезпечення, яке буде реалізовувати автоматизовану систему керування ділінгом.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» (кафедра комп'ютерних систем управління) [Електронний ресурс] / уклад. В. М. Дубовой. – Вінниця: ВНТУ, 2023 – (PDF, 45 с.).
2. Shubham, Python SciKit Learn Tutorial, 2022 [електронний ресурс] – Режим доступу: <https://www.digitalocean.com/community/tutorials/python-scikit-learn-tutorial#scikit-learn>
3. scikit-learn Machine Learning in Python [електронний ресурс] – Режим доступу: <https://scikit-learn.org/stable/index.html>

4. pmdarima.arima.AutoARIMA – pmdarima 2.0.4 documentation. alkaline-ml - Distilling machine learning into its basics. URL: <https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html> (date of access: 25.11.2024).
5. Top Trending Stocks: US stocks with the highest interest today - Yahoo Finance. Yahoo Finance - Stock Market Live, Quotes, Business & Finance News. URL: <https://finance.yahoo.com/markets/stocks/trending/>

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- навчання моделі прогнозування часових рядів;
- прогноз цінності фінансових інструментів;
- формування рекомендацій щодо складання портфелю активів;

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Персональний комп'ютер на базі WINDOWS 10;
- Інтерпретатор Python;
- Бібліотека mt4forexparser.
- Бібліотека pmdarima.

5.2.2. Умови експлуатації системи:

- регулярне оновлення даних для підтримки їх актуальності.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи. Постановка задач дослідження «03»_09_2024р.
2. Розробка структури програмного забезпечення системи «23»_09_2024 р.
3. Дослідження моделей прогнозування часових рядів «14»_11_2024 р.

4. Розробка та тестування програмного забезпечення системи «26»__11__
2024 р.

6.2 Графічні матеріали:

1. Розробка моделі системи «25»__09__ 2024 р.
2. Розробка блок-схеми програмного забезпечення «28»__09__ 2024 р.
3. Розробка презентації «26»__11__ 2024 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «208»__11__ 2024 р.
- 7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «1»__12__ 2024 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «7»__12__ 2024 р.

Додаток В

(Вибірковий)

Лістинг програми дослідження методів прогнозування часових рядів:

```

from mt4forexparser.MT4ForexParser import MT4ParseToPD as Parser
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from math import sqrt
from sklearn.metrics import mean_squared_error
import pmdarima as pm
import numpy as np
import math
import tensorflow as tf
#from keras.models import Sequential
import os
import pandas
#from keras.layers import LSTM
#from keras.layers import Dense
#from keras.optimizers import Adam
#from keras.metrics import MeanSquaredError
#from keras.metrics import RootMeanSquaredError
from sklearn.metrics import mean_absolute_error
#from keras.callbacks import ModelCheckpoint
#from tensorflow.keras.layers import Embedding
from datetime import date, timedelta
from datetime import datetime
import random

#df = Parser(historyFile="AAPL1440.hst")
df = pandas.read_csv('F1440.csv', delimiter=',')

#df.rename(columns={df.columns[0]: 'date', df.columns[1]: 'time', df.columns[2]: 'open', df.
columns[3]: 'high', df.columns[4]: 'low', df.columns[5]: 'close', df.columns[6]: 'volume'},
inplace=True)
#df.columns[1] = 'time'
print(df)

#df = df.drop(np.where(df.iloc[:,1]!="01:00")[0])#for GBPUSD60 not AAPL1440
tmp = df.iloc[:,0].values

rmse_sum = 0
relative_rmse = list()
relative_rmse_sum = 0
kmeans = KMeans(n_clusters=3)
kmeans.fit(df.iloc[:,2:7])#kmeans.labels_
#plt.scatter(X_, df.iloc[:,5], c=kmeans.labels_)
#plt.xlabel('Date')
#plt.ylabel('Close price')
#plt.show()
Xx = df.iloc[:,2:7].values
#Xx = np.column_stack((Xx,kmeans.labels_))
y = list()
y_pred = list()
direction_accuracy = 0
F = 0
delta_price_1 = list()
delta_price21 = list()

```

```

delta_price31 = list()
delta_price_2 = list()
delta_price22 = list()
delta_price32 = list()
for i in range(1,51):
    model1 = RandomForestRegressor()

    size = int(len(Xx) *0.8+i)
    train, test = Xx[size-60:size], Xx[size+6:int(len(Xx))]
    #train = np.column_stack((train,Xx[size-61:size-1]))
    #test = np.column_stack((test,Xx[size-1:int(len(Xx))-1]))
    history = df.iloc[size-60+7:size+7,5].values
    model1.fit(train,history)
    forecasts = model1.predict(test)
    y.append(test[7][3])
    y_pred.append(forecasts[0])
    tpm = np.mean(df.iloc[:size,5].values)
    F += 1
    if(test[7][3] > test[0][3]):
        delta_price31.append(test[7][3] - test[0][3])
        if forecasts[0] > test[0][3]:
            delta_price_1.append(test[7][3] - test[0][3])
            direction_accuracy +=1
        else:
            delta_price21.append(test[7][3] - test[0][3])
    else:
        delta_price32.append(test[7][3] - test[0][3])
        if forecasts[0] > test[0][3]:
            delta_price_2.append(test[7][3] - test[0][3])
        else:
            delta_price22.append(test[7][3] - test[0][3])
            direction_accuracy +=1

mae = mean_absolute_error(y, y_pred)
mse = mean_squared_error(y, y_pred)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((np.array(y) - np.array(y_pred)) / np.array(y))) * 100
print(f"F = {F}: direction_accuracy: {direction_accuracy}")
direction_accuracy = direction_accuracy/F
print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"MAPE: {mape:.2f}%")
print(f"direction accuracy: {direction_accuracy*100:.2f}%")
print(f"price growth in cases of price growth forecast: {sum(delta_price_1):.3f}")
print(f"price growth in cases of a price drop forecast: {sum(delta_price21):.3f}")
print(f"total price increase: {sum(delta_price31):.3f}\n")
print(f"price drop in cases of price growth forecast: {sum(delta_price_2):.3f}")
print(f"price drop in cases of price drop forecast: {sum(delta_price22):.3f}")
print(f"total price drop: {sum(delta_price32):.3f}\n")
print(f"benefit in cases of price growth forecast:
{sum(delta_price_1)+sum(delta_price_2):.3f}")
print(f"benefit in cases of price drop forecast:
{sum(delta_price21)+sum(delta_price22):.3f}")
print(f"total price benefit: {sum(delta_price31)+sum(delta_price32):.3f}\n")

y = list()
y_pred = list()
direction_accuracy = 0
delta_price_1 = list()
delta_price21 = list()
delta_price31 = list()
delta_price_2 = list()
delta_price22 = list()
delta_price32 = list()
direction = list()

```



```

#max1 = 0
#max2 = 0
F = 0
for i in range(1,51):
    size = int(len(Xx) *0.8+i)
    train, test = Xx[size-60:size], Xx[size+6:int(len(Xx))]
    history = df.iloc[7+size-60:size+7,5].values
    #train = np.column_stack((train,Xx[size-61:size-1]))
    #test = np.column_stack((test,Xx[size-1:int(len(Xx))-1]))
    model2 = pm.auto_arima(history,train)
    forecasts = model2.predict(len(test),test)
    y_pred.append(forecasts[0])
    y.append(test[7][3])
    F +=1
    if(test[7][3] > test[0][3]):
        delta_price31.append(test[7][3] - test[0][3])
        if forecasts[0] > test[0][3]:
            delta_price_1.append(test[7][3] - test[0][3])
            direction_accuracy +=1
        else:
            delta_price21.append(test[7][3] - test[0][3])
    else:
        delta_price32.append(test[7][3] - test[0][3])
        if forecasts[0] > test[0][3]:
            delta_price_2.append(test[7][3] - test[0][3])
        else:
            delta_price22.append(test[7][3] - test[0][3])
            direction_accuracy +=1

mae = mean_absolute_error(y, y_pred)
mse = mean_squared_error(y, y_pred)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((np.array(y) - np.array(y_pred)) / np.array(y))) * 100
print(f"F = {F}: direction_accuracy: {direction_accuracy}")
direction_accuracy = direction_accuracy/F

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"MAPE: {mape:.2f}%")
print(f"direction accuracy: {direction_accuracy*100:.2f}%")
print(f"price growth in cases of price growth forecast: {np.sum(delta_price_1):.3f}")
print(f"price growth in cases of a price drop forecast: {np.sum(delta_price21):.3f}")
print(f"total price increase: {np.sum(delta_price31):.3f}\n")
print(f"price drop in cases of price growth forecast: {np.sum(delta_price_2):.3f}")
print(f"price drop in cases of price drop forecast: {np.sum(delta_price22):.3f}")
print(f"total price drop: {sum(delta_price32):.3f}\n")
print(f"benefit in cases of price growth forecast:
{sum(delta_price_1)+sum(delta_price_2):.3f}")
print(f"benefit in cases of price drop forecast:
{sum(delta_price21)+sum(delta_price22):.3f}")
print(f"total price benefit: {sum(delta_price31)+sum(delta_price32):.3f}\n")

```


Додаток Г

(вибірковий)

Лістинг автоматизованої системи керування ділінгом:

```

from tkinter import*
from tkinter import filedialog
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
from mt4forexparser.MT4ForexParser import MT4ParseToPD as Parser
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestRegressor
import pmdarima as pm
from datetime import date, timedelta
from datetime import datetime
import re
import sys
import pandas

class user:
    def __init__(self):
        file1 = open("users BD.txt")
        content1 = ' '
        self.portfolio_names = list()
        while 1:
            content1 = file1.readline()
            if content1=='':
                break
            self.portfolio_names.append(content1)
            content2 = file1.readline()
        file1.close()
        self.is_open = False

    def open_portfolio(self,portfolio_iterator):
        #self.rpog =
        prognos([["AAPL1440.hst"],["00:00"]],[["GBPUSD60.hst"],["01:00"]]])
        self.is_open = True
        self.asset_names = list()
        self.asset_times = list()
        self.asset_test_remove = list()
        self.assets = {}
        self.balance = 0
        self.current_portfolio = self.portfolio_names[portfolio_iterator]
        file1 = open("users BD.txt")
        content = file1.readline()
        while content != self.current_portfolio:
            content = file1.readline()
        content = file1.readline()
        file1.close()
        i = content.find(" ")
        i += content[i:].find(" ")
        self.balance = float(content[:i])
        i+=1
        #ef = list()
        while content[i:].find(" ") != -1:
            #if content[i:].find(" ") == -1:
            #    break
            asset_name = content[i:i+content[i:].find(" ")]
            self.asset_names.append(asset_name)
            i += content[i:].find(" ")
            i+=1
            self.asset_times.append(content[i:i+content[i:].find(" ")]])
            i += content[i:].find(" ")

```

```

        i+=1
        self.asset_test_remove.append(int(content[i:i+content[i:].find(" ")]))
        i += content[i:].find(" ")
        i+=1
        self.assets[asset_name] = int(content[i:i+content[i:].find(" ")])
        i += content[i:].find(" ")
        i+=1
        #ef.append(float(content[i:i+content[i:].find(" ")]))
        #i += content[i:].find(" ")
        #i+=1

        #print(content[i:])
        self.asset_test_global_remove = int(content[i:])
        for i in range(len(self.asset_names)):
            self.asset_test_remove[i] += self.asset_test_global_remove
        #print(i)
        prognos_input = []
        for i in range(0,len(self.asset_names)):

prognos_input.append([[self.asset_names[i]+".csv"],[self.asset_test_remove[i]],[self.a
sset_times[i]]])#, [ef[i]]
            self.rpog = prognos(prognos_input)

    def get_prognos(self):
        return self.rpog

    def get_portfolio_of_assets(self, text):
        absolute_decrease = self.rpog.get_absolute_decrease()
        average_relative_advantage = self.rpog.get_average_relative_advantage()
        #effectiveness_of_method_for_tool =
self.rpog.get_effectiveness_of_method_for_tool()
        todays_instrument_profitability_index =
self.rpog.get_todays_instrument_profitability_index()
        full_balance = self.balance

        if len(absolute_decrease) > 1:
            minim = min(absolute_decrease)
            for i in range(0,len(absolute_decrease)):
                absolute_decrease[i] -= minim
                if average_relative_advantage[i] == 0:
                    absolute_decrease[i] = 0
        else:
            if len(absolute_decrease) == 1:
                if average_relative_advantage[0] == 0:
                    absolute_decrease[0] = 0

        for i in range(0,len(self.asset_names)):
            full_balance +=
self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[-1,5]
            sum1 = sum(absolute_decrease)
            sum2 = 0
            for i in range(0,len(self.asset_names)):

sum2+=average_relative_advantage[i]*todays_instrument_profitability_index[i]
            if sum1 == 0:
                per_asset1 = 0
            else:
                per_asset1 = full_balance / sum1
            if sum2 == 0:
                per_asset2 = 0
            else:
                per_asset2 = full_balance / sum2
        text.config(state = NORMAL)
        text.delete(1.0, END)
        text.insert(INSERT, 'conservative asset package\n')

```

```

inp = int()

tmp = 0
#print("conservative asset package")
for i in range(0,len(self.asset_names)):
    #for q in range(7,1,-1):

        #inp = input()
        #print(int(inp)*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
q,5])
        #print(-self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
q,5]+self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-(q-1),5])
        #print(int(inp)*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-(q-
1),5])
        #print(int(inp)*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-(q-
1),5] - int(inp)*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-q,5])
        tmp +=
self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
1,5] -
self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
7,5]
        #print(self.assets[self.asset_names[i]])

#print(self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(
i))-7,5])
        #print(self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-1,5]-
self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-7,5])

#print(self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(
i))-1,5] -
self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
7,5])
        #print(full_balance - 10000)

        #print(per_asset1*absolute_decrease[i]/self.rpog.get_DF(i).iloc[-1,5])
        text.insert(INSERT,self.asset_names[i]+' :
'+str(int(per_asset1*absolute_decrease[i]/self.rpog.get_DF(i).iloc[-1,5]))+'\n')
        #print("aggressive asset package")
        text.insert(INSERT,'\naggressive asset package\n')
        for i in range(0,len(self.asset_names)):

#print(todays_instrument_profitability_index[i]*per_asset2*average_relative_advantage[
i]/self.rpog.get_DF(i).iloc[-1,5])
        #print(self.assets[self.asset_names[i]])

#print(self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(
i))-7,5])
        #print(self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-1,5]-
self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-7,5])

#print(self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(
i))-1,5] -
self.assets[self.asset_names[i]]*self.rpog.get_DF(i).iloc[len(self.rpog.get_DF(i))-
7,5])

        text.insert(INSERT,self.asset_names[i]+' :
'+str(int(todays_instrument_profitability_index[i]*per_asset2*average_relative_advanta
ge[i]/self.rpog.get_DF(i).iloc[-1,5]))+'\n')
        #print(tmp)
        #print(full_balance - 10000)

        text.config(state = DISABLED)

def get_portfolio_names(self):
    return self.portfolio_names

```

```

def get_balans_and_portfolio_of_assets(self, text):
    text.config(state = NORMAL)
    text.delete(1.0, END)
    for i in self.asset_names:
        text.insert(INSERT, i + ': ')
        text.insert(INSERT, str(self.assets[i]) + '\n')
    text.insert(INSERT, 'Balance: ' +
str(float("{:.2f}".format(self.balance))) + '\n')
    text.config(state = DISABLED)

def get_price(self, iterator):
    self.rpog.get_DF(iterator).iloc[-1,5]

def get_assets_names(self):
    return self.asset_names

def set_new_asset(self):
    sourceDir = filedialog.askopenfilename()
    i = 0
    while sourceDir[i:].find('/') != -1:
        i += sourceDir[i:].find('/')
        i += 1
    if (sourceDir[len(sourceDir)-4:] != '.csv'):
        return -1
    for I in range(0, len(self.asset_names)):
        if sourceDir[i:len(sourceDir)-4] == self.asset_names[I]:
            return -1
    self.asset_names.append(sourceDir[i:len(sourceDir)-4])
    Df = pandas.read_csv(sourceDir[i:len(sourceDir)],
delimiter=',') #Parser(historyFile=sourceDir[i:len(sourceDir)])
    self.asset_times.append(Df.iloc[0,1])
    self.asset_test_remove.append(0)
    self.assets[sourceDir[i:len(sourceDir)-4]] = 0
    prognos_input = []
    prognos_input.append([[sourceDir[i:len(sourceDir)-
4]+".csv"], [0], [Df.iloc[0,1]]]) #, [ef[i]]
    self.rpog.set_new_model(prognos_input)
    return 1
    print(sourceDir[i:len(sourceDir)-4])

def delete_asset(self, iterator):
    if self.assets[self.asset_names[iterator]] > 0:
        self.balance +=
self.assets[self.asset_names[iterator]] * self.rpog.get_DF(iterator).iloc[-1,5]
    self.rpog.delete_model(iterator, self.asset_names[iterator])
    del self.assets[self.asset_names[iterator]]
    self.asset_names.pop(iterator)
    self.asset_times.pop(iterator)
    self.asset_test_remove.pop(iterator)

def save_portfolio(self):
    file1 = open("users BD.txt")
    write_content = ''
    content1 = ''
    while 1:
        content1 = file1.readline()
        if content1 == '':
            break
        write_content = write_content + content1
        if content1 == self.current_portfolio:
            content1 = file1.readline()
            write_content = write_content + str(self.balance)
            for i in range(len(self.asset_names)):
                write_content = write_content + ' ' + self.asset_names[i] + ' ' +
self.asset_times[i] + ' ' + str(self.asset_test_remove[i] -
self.asset_test_global_remove) + ' ' + str(self.assets[self.asset_names[i]])

```

```

        write_content = write_content + ' ' +
str(self.asset_test_global_remove) + '\n'
    else:
        content2 = file1.readline()
        write_content = write_content + content2
    file1.close()
    with open("users BD.txt",'w') as file1:
        file1.write(write_content)

def is_portfolio_exists(self,enter):
    file1 = open("users BD.txt")
    content1 = ' '
    while 1:
        content1 = file1.readline()
        if content1=='':
            break
        if content1==enter:
            file1.close()
            return True
        else:
            content2 = file1.readline()
    file1.close()
    return False

def create_portfolio(self,enter1,enter2):
    file1 = open("users BD.txt")
    write_content = ''
    content1 = ' '
    while 1:
        content1 = file1.readline()
        if content1=='':
            break
        write_content = write_content + content1
    file1.close()
    write_content = write_content + enter1 + '\n' + enter2 + ' ' + '0' + '\n'
    self.portfolio_names.append(enter1 + '\n')
    with open("users BD.txt",'w') as file1:
        file1.write(write_content)

def delete_portfolio(self,enter1):
    file1 = open("users BD.txt")
    write_content = ''
    content1 = ' '

    while 1:
        content1 = file1.readline()
        if content1=='':
            break
        if content1!=self.portfolio_names[enter1]:
            write_content = write_content + content1
            content2 = file1.readline()
            write_content = write_content + content2
        else:
            content2 = file1.readline()
    file1.close()
    self.portfolio_names.pop(enter1)
    with open("users BD.txt",'w') as file1:
        file1.write(write_content)

class prognos:
    def __init__(self, df):
        #df = Parser(historyFile="AAPL1440.hst")
        self.DF = list()

```

```

self.Dictionary = {}
for i in range(0,len(df)):
    Df = pandas.read_csv(df[i][0][0],
delimiter=',')#Parser(historyFile=df[i][0][0])
    Df = Df.drop(np.where(Df.iloc[:,1]!=df[i][2][0])[0])
    Df = Df.iloc[:Df.shape[0]-df[i][1][0],]
    #Df = Df.drop(Df.iloc[-1])#Df.drop(np.where(Df.iloc[Df.size-2:,])[0])
#Df.iloc[:Df.size-2,]#df[i][1][0]
    self.Dictionary[df[i][0][0][:len(df[i][0][0])-4]] = i
    self.DF.append(Df)
self.absolute_decrease = list()
self.average_relative_advantage = list()
self.relative_advantage = list()
self.forecasts = list()
self.effectiveness_of_method_for_tool = list()
self.todays_instrument_profitability_index = list()
self.train_size = 62

for i in range(0,len(self.DF)):
    delta_price = list()
    self.absolute_decrease.append(0)
    self.relative_advantage.append(0)
    self.average_relative_advantage.append(0)
    self.todays_instrument_profitability_index.append(0)
    self.effectiveness_of_method_for_tool.append(0)
    self.forecasts.append([])
    X = self.DF[i].iloc[:,2:7].values
    size = len(X)-7
    train, test = X[size-self.train_size:size], X[size+6:size+7]
    history = self.DF[i].iloc[size-self.train_size+7:size+7,5].values
    model = pm.auto_arima(history,train)
    forecasts1 = model.predict(len(train),train)
    forecasts2 = model.predict(1,test)
    self.forecasts[i] = forecasts1[len(forecasts1)-6:len(forecasts1)]
    self.forecasts[i] = np.append(self.forecasts[i],[forecasts2[0]], axis=0)
    for I in range(size-self.train_size,size):
        delta_price.append(self.DF[i].iloc[I+1:I+2,5].values[0] -
self.DF[i].iloc[I:I+1,5].values[0])
        if self.DF[i].iloc[I+1:I+2,5].values[0] -
self.DF[i].iloc[I:I+1,5].values[0] < 0:
            self.absolute_decrease[i] += self.DF[i].iloc[I+1:I+2,5].values[0]
- self.DF[i].iloc[I:I+1,5].values[0]

    if np.mean(delta_price) < 0:
        self.average_relative_advantage[i] = 0
    else:
        self.average_relative_advantage[i] = np.mean(delta_price)

    if self.DF[i].iloc[-1,5] < forecasts2[0]:
        self.todays_instrument_profitability_index[i] = 1
    else:
        self.todays_instrument_profitability_index[i] = 0

def set_new_model(self,df):
    for i in range(0,len(df)):
        Df = pandas.read_csv(df[i][0][0],
delimiter=',')#Parser(historyFile=df[i][0][0])
        Df = Df.drop(np.where(Df.iloc[:,1]!=df[i][2][0])[0])
        Df = Df.iloc[:Df.shape[0]-df[i][1][0],]
        self.DF.append(Df)
        self.Dictionary[df[i][0][0][:len(df[i][0][0])-4]] = len(self.DF)-1
    for i in range(len(self.DF)-1,len(self.DF)):
        delta_price = list()
        self.absolute_decrease.append(0)
        self.relative_advantage.append(0)
        self.average_relative_advantage.append(0)

```

```

self.todays_instrument_profitability_index.append(0)
self.effectiveness_of_method_for_tool.append(0)
self.forecasts.append([])
X = self.DF[i].iloc[:,2:7].values
size = len(X)-7
train, test = X[size-self.train_size:size], X[size+6:size+7]
history = self.DF[i].iloc[size-self.train_size+7:size+7,5].values
model = pm.auto_arima(history,train)
forecasts1 = model.predict(len(train),train)
forecasts2 = model.predict(1,test)
self.forecasts[i] = forecasts1[len(forecasts1)-6:len(forecasts1)]
self.forecasts[i] = np.append(self.forecasts[i],[forecasts2[0]], axis=0)
for I in range(size-self.train_size,size):
    delta_price.append(self.DF[i].iloc[I+1:I+2,5].values[0] -
self.DF[i].iloc[I:I+1,5].values[0])
    if self.DF[i].iloc[I+1:I+2,5].values[0] -
self.DF[i].iloc[I:I+1,5].values[0] < 0:
        self.absolute_decrease[i] += self.DF[i].iloc[I+1:I+2,5].values[0]
- self.DF[i].iloc[I:I+1,5].values[0]
    if np.mean(delta_price) < 0:
        self.average_relative_advantage[i] = 0
    else:
        self.average_relative_advantage[i] = np.mean(delta_price)
if self.DF[i].iloc[-1,5] < forecasts2[0]:
    self.todays_instrument_profitability_index[i] = 1
else:
    self.todays_instrument_profitability_index[i] = 0

def delete_model(self,iterator,asset_names):
    del self.Dictionary[asset_names]
    self.DF.pop(iterator)
    self.absolute_decrease.pop(iterator)
    self.relative_advantage.pop(iterator)
    self.average_relative_advantage.pop(iterator)
    self.todays_instrument_profitability_index.pop(iterator)
    self.effectiveness_of_method_for_tool.pop(iterator)
    self.forecasts.pop(iterator)
def get_todays_instrument_profitability_index(self):
    return self.todays_instrument_profitability_index
def get_effectiveness_of_method_for_tool(self):
    return self.effectiveness_of_method_for_tool
def get_first_forecast(self,iterator):
    return self.forecasts[iterator][self.train_size]
def get_relative_advantage(self):
    return self.relative_advantage
def get_absolute_decrease(self):
    return self.absolute_decrease
def get_average_relative_advantage(self):
    return self.average_relative_advantage
def get_DF(self,iterator):
    return self.DF[iterator]
def get_forecasts(self,iterator):
    return self.forecasts[iterator]
def get_train_size(self):
    return self.train_size

def write__():
    # створення функції для друку фрази
    print("I like Tkinter!")
def graph_output(A,b):
    if len(listbox1.curselection())==1 and len(listbox2.curselection())==1 and
var1.get():
        B1 = listbox1.curselection()[0]
        B2 = listbox2.curselection()[0]
        tmp = A.get_DF(B1).iloc[len(A.get_DF(B1)).iloc[:,0]]-
A.get_train_size():,0].values

```

```

X1_ = []#np.append(A.get_DF(B1).iloc[len(A.get_DF(B1).iloc[:,0])-30:,0],
'tomorrow')#
X1__ = []
for i in tmp:
    X1_ = np.append(X1_,datetime.strptime(i, '%Y.%m.%d').date())
    plt.subplot(1,2,1)
    plt.plot(X1_, A.get_DF(B1).iloc[len(A.get_DF(B1).iloc[:,0])-
A.get_train_size():len(A.get_DF(B1).iloc[:,0]),5].values, label='historical price
changes')
    tmp = A.get_DF(B1).iloc[-1,0]
    for i in range(1,8):
        X1__ = np.append(X1__,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(i))
        #X1_ = np.append(X1_,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(7))# 'tomorrow'
        #for i in range(1,20):
            #X1_ = np.append(X1_,X1_[len(X1_)-1] + timedelta(i))
        plt.plot(X1__, A.get_forecasts(B1), label='prediction')
        plt.xlabel('Date')
        plt.ylabel('Close price')
        plt.legend()
        tmp = A.get_DF(B2).iloc[len(A.get_DF(B2).iloc[:,0])-
A.get_train_size():,0].values
        X2_ = []#np.append(A.get_DF(B2).iloc[len(A.get_DF(B2).iloc[:,0])-30:,0],
'tomorrow')#
        X2__ = []
        for i in tmp:
            X2_ = np.append(X2_,datetime.strptime(i, '%Y.%m.%d').date())
            plt.subplot(1,2,2)
            plt.plot(X2_, A.get_DF(B2).iloc[len(A.get_DF(B2).iloc[:,0])-
A.get_train_size():len(A.get_DF(B2).iloc[:,0]),5].values, label='historical price
changes')
            tmp = A.get_DF(B2).iloc[-1,0]
            for i in range(1,8):
                X2__ = np.append(X2__,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(i))
                #X2_ = np.append(X2_,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(7))# 'tomorrow'
                #for i in range(1,20):
                    #X2_ = np.append(X2_,X2_[len(X2_)-1] + timedelta(i))
                plt.plot(X2__, A.get_forecasts(B2), label='prediction')
                plt.xlabel('Date')
                plt.ylabel('Close price')
                plt.legend()
                plt.show()
            elif len(listbox1.curselection())==1:
                B = listbox1.curselection()[0]
                tmp = A.get_DF(B).iloc[len(A.get_DF(B).iloc[:,0])-
A.get_train_size():,0].values
                X_ = []
                X__ = []
                for i in tmp:
                    X_ = np.append(X_,datetime.strptime(i, '%Y.%m.%d').date())
                    plt.plot(X_, A.get_DF(B).iloc[len(A.get_DF(B).iloc[:,0])-
A.get_train_size():len(A.get_DF(B).iloc[:,0]),5].values, label='historical price
changes')
                    tmp = A.get_DF(B).iloc[-1,0]
                    for i in range(1,8):
                        X__ = np.append(X__,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(i))
                        #X_ = np.append(X_,datetime.strptime(tmp, '%Y.%m.%d').date() +
timedelta(7))# 'tomorrow'
                        #for i in range(1,20):
                            #X_ = np.append(X_,X_[len(X_)-1] + timedelta(i))
                        plt.plot(X__, A.get_forecasts(B), label='prediction')
                        plt.xlabel('Date')

```



```

        plt.ylabel('Close price')
        plt.tight_layout()
        plt.legend()
        plt.show()
def DISABLED_():
    listbox2.config(state = DISABLED,selectmode = BROWSE)
def ENABLED():
    listbox2.config(state = NORMAL, selectmode = SINGLE)

def Calculate_the_price(val):
    if len(listbox3.curselection())==1:
        text3.config(state = NORMAL)
        text3.delete(1.0, END)
        text3.insert(INSERT,str(S1.get()) + ' by
'+str(float("{:.2f}".format(S1.get()*User.rpog.get_DF(listbox3.curselection()[0])).iloc
[-1,5])))
        text3.config(state = DISABLED)

def buy_():
    if len(listbox3.curselection())==1:
        if S1.get()*User.rpog.get_DF(listbox3.curselection()[0]).iloc[-1,5] <=
User.balance:
            User.balance -=
S1.get()*User.rpog.get_DF(listbox3.curselection()[0]).iloc[-1,5]
            User.assets[list1[listbox3.curselection()[0]]] += S1.get()
            User.get_balans_and_portfolio_of_assets(text2)
def sell_():
    if len(listbox3.curselection())==1:
        if S1.get() <= User.assets[list1[listbox3.curselection()[0]]]:
            User.balance +=
S1.get()*User.rpog.get_DF(listbox3.curselection()[0]).iloc[-1,5]
            User.assets[list1[listbox3.curselection()[0]]] -= S1.get()
            User.get_balans_and_portfolio_of_assets(text2)

def select_portfolio_name():
    if len(listbox.curselection())==1:
        P = listbox.curselection()[0]
        authorization.quit()
        authorization.destroy()
        User.open_portfolio(P)

def delete_select_portfolio_name():
    if len(listbox.curselection())==1:
        P = listbox.curselection()[0]
        User.delete_portfolio(P)
        listbox.delete(listbox.curselection()[0],listbox.curselection()[0])

def start_portfolio():
    if entry1.get() == '' or entry2.get() == '':
        return
    if User.is_portfolio_exists(entry1.get()+'\n') == False:
        User.create_portfolio(entry1.get(),entry2.get())
        listbox.insert(END, entry1.get())
        #authorization.quit()
        #authorization.destroy()
        #User.open_portfolio(P)

def set_new_asset_in_portfolio():
    if User.set_new_asset() == 1:
        User.get_portfolio_of_assets(text)
        User.get_balans_and_portfolio_of_assets(text2)
        list_ = User.get_assets_names()
        listbox1.insert (END, list_[len(list_)-1])
        listbox2.config(state = NORMAL)
        listbox2.insert (END, list_[len(list_)-1])

```

```

        listbox2.config(state = DISABLED)
        listbox3.insert (END, list_[len(list_)-1])
        listbox4.insert (END, list_[len(list_)-1])

def delete_asset_in_portfolio():
    if len(listbox4.curselection())==1:
        User.delete_asset(listbox4.curselection()[0])
        User.get_portfolio_of_assets(text)
        User.get_balans_and_portfolio_of_assets(text2)
        listbox1.delete (listbox4.curselection()[0])
        listbox2.config(state = NORMAL)
        listbox2.delete (listbox4.curselection()[0])
        listbox2.config(state = DISABLED)
        listbox3.delete (listbox4.curselection()[0])
        listbox4.delete (listbox4.curselection()[0])

def save_and_quit():
    User.save_portfolio()
    MyWindow.quit()
    MyWindow.destroy()

def check_num(newval):
    print(f'check_num({newval=!a})')
    return re.match('[0-9]*$', newval) is not None

User = user()
P_name = int()
authorization = Tk()
authorization.title("Authorization")

listbox = Listbox (authorization,height = 10, selectmode = SINGLE,exportselection = 0)
list1 = User.get_portfolio_names()
for i in list1:
    listbox.insert (END, i)
select_portfolio = Button(authorization,text='Open\nselect portfolio',command =
select_portfolio_name)#open_portfolio
del_portfolio = Button(authorization,text='Delete\nselect portfolio',command =
delete_select_portfolio_name)#open_portfolio
entry1 = Entry(authorization)
check_num_wrapper = (authorization.register(check_num), '%P')
entry2 = Entry(authorization, validate='key', validatecommand=check_num_wrapper)
l1 = Label(text="Name of\nnew portfolio")
l2 = Label(text="Starting balance of\n new portfolio")
start_new_portfolio = Button(authorization,text='start\nnew\nportfolio',command =
start_portfolio)
scrollbar = Scrollbar (authorization)
scrollbar ['command'] = listbox.yview
listbox['yscrollcommand'] = scrollbar.set

scrollbar.grid(row = 0,column=1,rowspan=2)
listbox.grid(row = 0,column=0,rowspan=2)
select_portfolio.grid(row = 0,column=2)
del_portfolio.grid(row = 1,column=2)
l1.grid(row = 2,column=0)
entry1.grid(row = 3,column=0)
l2.grid(row = 4,column=0)
entry2.grid(row = 5,column=0)
start_new_portfolio.grid(row = 3,column=2,rowspan=3)
authorization.mainloop()
if User.is_open == False:
    sys.exit()

#print(User.get_prognos().get_average_deviation())
#print(User.get_prognos().get_relative_advantage())

MyWindow = Tk()

```

```

MyWindow.title("Automated control system")
MyWindow.geometry("760x515")

listbox1 = Listbox (MyWindow,height = 10, selectmode = SINGLE,exportselection = 0)
list1 = User.get_assets_names()
listbox2 = Listbox (MyWindow,height = 10, selectmode = SINGLE,exportselection = 0)
listbox3 = Listbox (MyWindow,width = 25,height = 10, selectmode =
SINGLE,exportselection = 0)
listbox4 = Listbox (MyWindow,width = 25,height = 3, selectmode =
SINGLE,exportselection = 0)
for i in list1:
    listbox1.insert (END, i)
    listbox2.insert (END, i)
    listbox3.insert (END, i)
    listbox4.insert (END, i)
var1 = BooleanVar()
check2= Radiobutton(MyWindow, text="2 graphs", variable=var1, value=True,
command=ENABLED)
check1 = Radiobutton(MyWindow, text="1 graphs", variable=var1, value=False,
command=DISABLED_) # команда command прив'язує подію
butt = Button(MyWindow,text='graph output')
New_asset_butt = Button(MyWindow,text='add a
new\nfinancial\ninstrument',command=set_new_asset_in_portfolio)
Del_asset_butt =
Button(MyWindow,text='delete\nfinancial\ninstrument',command=delete_asset_in_portfolio
)
buy_butt = Button(MyWindow,text='Buy',command=buy_)
sell_butt = Button(MyWindow,text='Sell',command=sell_)
quit_butt = Button(MyWindow,text='Save and quit',command=save_and_quit)
text = Text (MyWindow,width = 26,height = 10,state = DISABLED)
text2 = Text (MyWindow,width = 25,height = 10,state = DISABLED)
text3 = Text (MyWindow,width = 20,height = 3,state = DISABLED)
scrollbar1 = Scrollbar (MyWindow)
scrollbar2 = Scrollbar (MyWindow)
scrollbar3 = Scrollbar (MyWindow)
scrollbar4 = Scrollbar (MyWindow)
scrollbar5 = Scrollbar (MyWindow)
S1 = Scale(MyWindow,length = 200,from_=0,to=100,command = Calculate_the_price)
separator = Frame(MyWindow,width = 5, height=450, bd=1, relief=SUNKEN)
separator2 = Frame(MyWindow,width = 5, height=450, bd=1, relief=SUNKEN)
#com = ttk.Combobox(MyWindow, values = list1)
#calculate_the_price_butt = Button(MyWindow,text='Calculate\nPrice')

User.get_portfolio_of_assets(text)
User.get_balans_and_portfolio_of_assets(text2)
scrollbar1 ['command'] = text.yview # Прив'язування 1
text ['yscrollcommand'] = scrollbar1.set # Прив'язування 2
scrollbar2 ['command'] = listbox1.yview
listbox1['yscrollcommand'] = scrollbar2.set
scrollbar3 ['command'] = listbox2.yview
listbox2['yscrollcommand'] = scrollbar3.set
scrollbar4 ['command'] = listbox3.yview
listbox3['yscrollcommand'] = scrollbar4.set
scrollbar5 ['command'] = listbox4.yview
listbox4['yscrollcommand'] = scrollbar5.set
butt.config(command=lambda A = User.get_prognos(),B = list1: graph_output(A,B))
# command установлює дію на кнопку
listbox2.config(state = DISABLED)

New_asset_butt.grid(row=2, column=3)
Del_asset_butt.grid(row=4, column=3)
listbox4.grid(row = 3,column=3)
S1.grid(row=3, column=6,rowspan=3)
butt.grid(row = 5,column=0)
listbox1.grid (row = 1,column=0)
listbox2.grid (row = 2,column=0,rowspan=3)

```

```
listbox3.grid(row = 1,column=6,columnspan=2)
text3.grid(row = 2,column=6,columnspan=3)
buy_buttn.grid(row = 3,column=7)
sell_buttn.grid(row = 4,column=7)
check1.grid(row = 6,column=0)
check2.grid(row = 7,column=0)
text.grid(row = 1,column=3)
text2.grid(row = 1,column=10)
scrollbar1.grid(row = 1,column=4)
scrollbar2.grid(row = 1,column=1)
scrollbar3.grid(row = 2,column=1,rowspan=3)
scrollbar4.grid(row = 1,column=8)
scrollbar5.grid(row = 3,column=4)
quit_buttn.grid(row = 0,column=9,columnspan=2)
separator.grid(row = 0,column=2,rowspan=8)
separator2.grid(row = 0,column=5,rowspan=8)

MyWindow.mainloop()

print("good work")
```

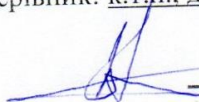
Додаток Д

(Обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

Студент 2 курсу, групи ЗАКІТР-23мРостислав ЧЕРВІНСЬКИЙ

Керівник: к.т.н., доцент кафедри АІВТ

Владислав КАБАЧІЙ

Вінниця 2024р.

Автоматизована система керування фондовим ділінгом для трейдингового відділу банку

Підготував студент групи ЗАКІТР-23м
Червінський Ростислав

Рисунок Д-1

Мета і завдання роботи

Об'єкт дослідження: Ділінг фінансових інструментів.

Мета роботи: розробка програмного забезпечення, яке буде реалізовувати автоматизовану систему керування ділінгом.

Завдання дослідження:

- Вивчити літературу по даній темі.
- Дослідити об'єкт автоматизації.
- Дослідити конкурентів, для обґрунтування напрямку дослідження у роботі.
- Провести огляд методів прогнозування часових рядів.
- Дослідити принципи роботи оглянутих методів.
- Розробити та протестувати програмне забезпечення.

Методи дослідження:

- Вивчення спеціальної літератури.
- Узагальнення і систематизація матеріалів по даній тематиці.
- Створення моделей для прогнозу цін фінансових інструментів.
- Порівняння вигідності при вкладах у фінансові інструменти постійно, у випадку прогнозованого росту фінансових інструментів, та у випадках прогнозованого падіння фінансових інструментів.

Рисунок Д-2

Структурний опис об'єкта

Спочатку працівник банку (1) розпочинає роботу на своєму робочому місці (2). Працівник дистанційно зв'язується із клієнтами банку (3) для обговорення наданих послуг. Працівник використовує програмне забезпечення, робота якого ґрунтується на даних із підготовленої іншим відділом банку бази (4). Враховуючи поради програмного забезпечення, щодо купівлі продажу цінних паперів, працівник зв'язується із депозитарієм (7) та підготтовлює до продажу цінні папери із портфелю активів банку (5) та/або клієнта (6). Далі працівник виконує продаж або купівлю цінних паперів на біржі (8), у випадку купівлі передає цінні папери на зберігання у депозитарій (7).



Рисунок Д-3

Системи-аналоги



MetaTrader це популярна торгова платформа, яка розроблена компанією MetaQuotes Ltd.

MetaTrader дозволяє трейдерам здійснювати угоди купівлі та продажу активів (валюти, акції, ф'ючерси, метали та інші фінансові інструменти) у режимі реального часу.

Трейдери можуть налаштовувати шаблони для аналізу та отримувати торгові сигнали на основі попередньо визначених умов.

Платформа дозволяє створювати автоматизовані торгові стратегії за допомогою експертних радників. Ці програми можуть виконувати торгові операції автоматично на основі заданих правил і умов.

Bloomberg Terminal — це провідна платформа для фінансових професіоналів, яка надає доступ до потужних інструментів для аналізу ринкових даних, проведення торгових операцій, отримання новин, і підтримує широкий спектр фінансових інструментів. Bloomberg Terminal використовується банками, інвесторами, трейдерами, аналітиками та корпораціями для прийняття рішень на фінансових ринках.

Платформа надає новини в реальному часі, висвітлюючи події в економіці, фінансах, політиці та бізнесі. Bloomberg є одним із провідних постачальників фінансових новин, що дозволяє користувачам бути в курсі останніх змін.

Bloomberg Terminal пропонує інструменти для аналізу ринкових тенденцій, оцінки економічних показників і відстеження корпоративних фінансів.

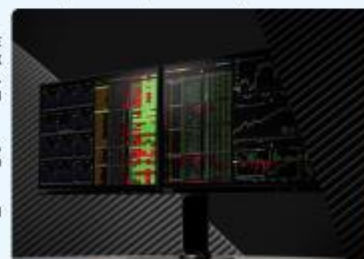


Рисунок Д-4

Часові ряди

Часовий ряд — це ряд точок даних, проіндексованих (або перелічених, або відкладених на графіку) в хронологічному порядку. Найчастіше часовий ряд є послідовністю, взятою на рівновіддалених точках в часі, які йдуть одна за одною. Таким чином, він є послідовністю даних дискретного часу.



Графік історії змін закриваючої ціни з часом.

Рисунок Д-5

Дослідження методів прогнозування часових рядів

ARIMA

MAE: 4.612124112470352
MSE: 33.49911149029436
RMSE: 5.787841695338113
MAPE: 6.56%
direction accuracy: 58.00%

price growth in cases of price growth forecast: 152.840
price growth in cases of a price drop forecast: 57.690
total price increase: 210.230
price drop in cases of price growth forecast: -90.100
price drop in cases of price drop forecast: -72.830
total price drop: -163.010
benefit in cases of price growth forecast: 62.300
benefit in cases of price drop forecast: -15.140
total price benefit: 47.220



MAE: 5.0237500000000015
MSE: 39.40397951440003
RMSE: 6.2772589172663595
MAPE: 7.06%
direction accuracy: 45.00%

Random forest

price growth in cases of price growth forecast: 109.590
price growth in cases of a price drop forecast: 100.640
total price increase: 210.230
price drop in cases of price growth forecast: -110.400
price drop in cases of price drop forecast: -52.530
total price drop: -163.010
benefit in cases of price growth forecast: -0.890
benefit in cases of price drop forecast: 48.110
total price benefit: 47.220

Рисунок Д-6



Рисунок Д-7

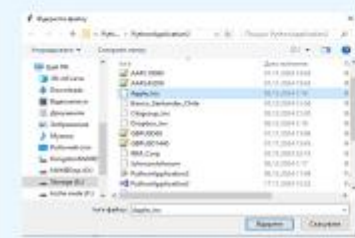


Рисунок Д-8

Інтерфейс



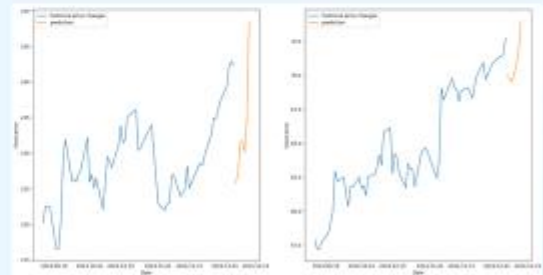
Вікно роботи з портфелем активів



Вікно додавання нового фінансового інструменту у портфель



Початкове вікно вибору портфеля активів

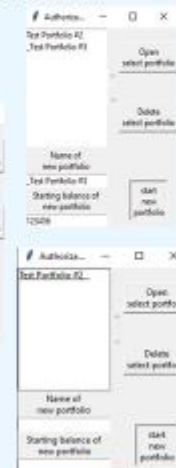
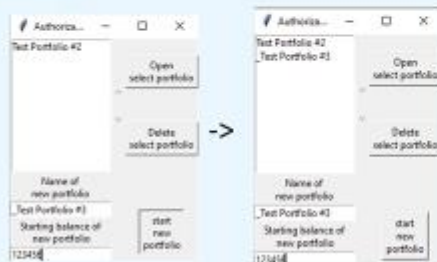


Графіки часових рядів для фінансових інструментів

Рисунок Д-9

Тестування розробленого ПО

Інтерфейс ПО стійкий до не коректного використання



Відсутність результату

Рисунок Д-10

Додавання та вилучення фінансових інструментів

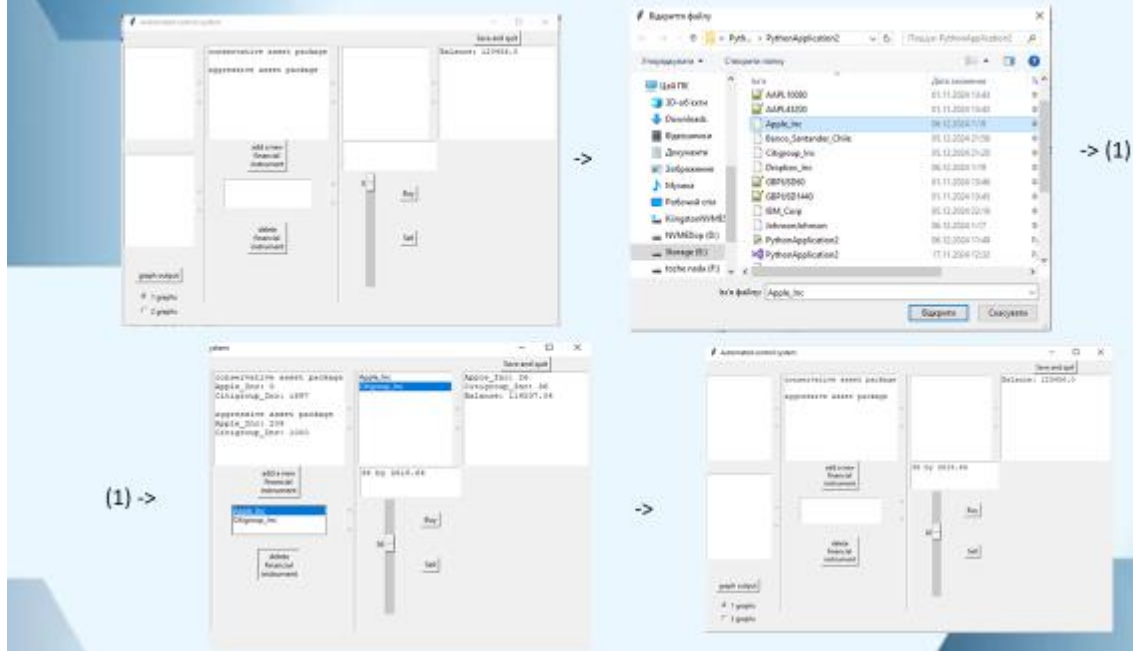


Рисунок Д-11

Великий експеримент

Консервативна стратегія

Агресивна стратегія

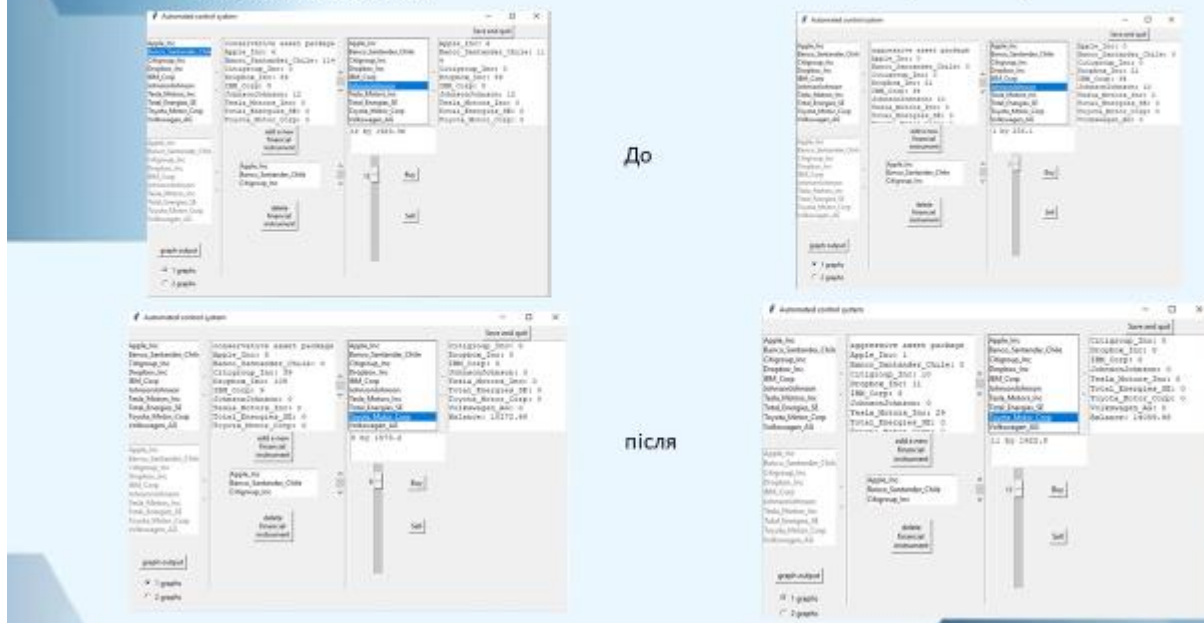


Рисунок Д-12

Висновки

В магістерській кваліфікаційній роботі була поставлена задача розробки автоматизованої системи ділінгу. У ході роботи було локалізовано напрямок дослідження на прогнозуванні цін фінансових інструментів за допомогою часових рядів. За результатами дослідження моделей прогнозування, побудованими методами машинного навчання, було обрано модель "ARIMA" як найкращу для поставленої задачі. У ході роботи було розроблено програмне забезпечення, яке реалізує автоматизовану систему керування ділінгом для трейдингового відділу банку. По результатам тестування ПО було визнано працездатним у повному обсязі.

Рисунок Д-13