

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Система прийняття торгових рішень на фінансових ринках на основі  
нечіткого кластерного аналізу часових рядів»

Виконав: студент 2 курсу, групи ІСТ-24м  
спеціальності 126 – Інформаційні системи та  
технології

(шифр і назва спеціальності)

Володимир ПАВЛЕНКО

(ПІБ студента)

Керівник: к.т.н., доцент кафедри АІТ  
Роман МАСЛІЙ

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

« 15 » 12 2025 р.

Опонент: д.т.н., проф. каф. КСУ  
В'ячеслав КОВТУН

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

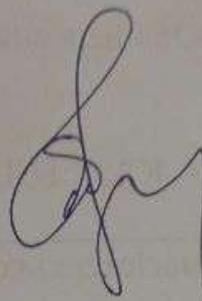
« 15 » 12 2025 р.

Допущено до захисту  
Завідувач кафедри АІТ  
д.т.н., проф. Олег БІСІКАЛО  
(науковий ступінь, вчене звання)

« 16 » 12 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
Рівень вищої освіти \_\_\_\_\_  
Галузь знань – \_\_\_\_\_  
Спеціальність – \_\_\_\_\_  
Освітньо-професійна програма – \_\_\_\_\_  
зображень



**ЗАТВЕРДЖУЮ**

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСКАЛО

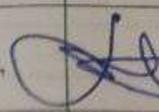
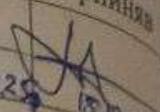
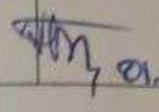
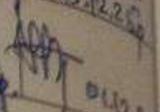
«26» 09 2025 р.

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Павленку Володимиру Віталійовичу  
(ПІБ автора повністю)

1. Тема роботи: Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів.  
Керівник роботи: к.т.н., доцент каф. АІТ Маслій Р. В.  
Затвердженні наказом ВНТУ від «24» вересня 2025 року № 313.
2. Строк подання роботи студентом: до «12» грудня 2025 року.
3. Вихідні дані до роботи: Історичні дані про котирування валютних пар EUR/USD, GBD/USD за 2024-2025 рр; Таймфрейм – 15М (15 хв.); Навчання Fuzzy C-Means з параметрами: c=2 кластери, m=2.0; використати мінімум 5 технічних індикаторів.
4. Зміст текстової частини: Вступ; Дослідження предметної області та існуючих методів; Математичне та алгоритмічне забезпечення системи прийняття торгових рішень; Розробка програмного забезпечення та тестування системи; Економічний розділ; Висновки; Список використаних джерел.
5. Перелік ілюстративного (або графічного) матеріалу: UML-діаграма діяльності. Етап навчання моделі; UML-діаграма діяльності. Етап інференсу моделі; UML-діаграма класів; Архітектура програмної системи; Візуалізація результатів торгівлі; Візуалізація статистичних показників торгівлі.

6. Консультанти розділів роботи

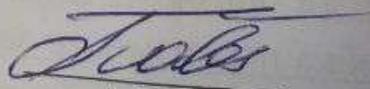
Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 – 3	Роман МАСЛІЙ, к.т.н., доцент кафедри АІТ	 25.09.25	 15.12.25
4	Наталія БУРЕННІКОВА, д.е.н., проф. каф. ЕПтаВМ	 01.11.25	 01.12.25

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	25.09–05.10.2025	вик.
2	Розробка математичного та алгоритмічного забезпечення	05.10 – 25.10.2025	вик.
3	Розробка програмного забезпечення	25.10 – 10.11.2025	вик.
4	Тестування розробленого програмного забезпечення	05.11 – 20.11.2025	вик.
5	Підготовка економічної частини	до 01.12.2025	вик.
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	20.11 – 03.12.2025	вик.
7	Попередній захист роботи	до 03.12.2025	вик.
8	Захист роботи	до 19.12.2025	вик.

Студент

  
(підпис)

Володимир ПАВЛЕНКО  
(прізвище та ініціали)

Керівник роботи

  
(підпис)

Роман МАСЛІЙ  
(прізвище та ініціали)

## АНОТАЦІЯ

УДК 004.4:336.76

Павленко В. В. Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів. Магістерська кваліфікаційна робота зі спеціальності 126 – Інформаційні системи та технології, освітня програма – Інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2025. 139 с.

Укр. мовою. Бібліогр.: 39 назв; рис.: 23; табл. 17.

Магістерська кваліфікаційна робота присвячена розробці автоматизованої торговельної системи для валютного ринку Forex на основі алгоритму нечіткої кластеризації Fuzzy C-Means. Запропонована система дозволяє виявляти латентні закономірності в ринкових даних, генерувати торгові сигнали на основі розрахунку ступеня належності до кластерів BUY та SELL, а також адаптуватися до змінних умов ринку за рахунок періодичного перенавчання моделі.

У даній роботі детально досліджується проблематика алгоритмічної торгівлі, аналізуються методи технічного аналізу та машинного навчання для прогнозування фінансових часових рядів. Особливу увагу приділено математичній формалізації алгоритму Fuzzy C-Means, вибору технічних індикаторів для формування простору ознак, а також проектуванню багатофільтрової системи для зменшення хибних сигналів. Реалізовано повний цикл розробки: від підготовки даних та навчання моделі до інтеграції з торговельною платформою MetaTrader 5 через TCP/IP протокол. Завершальним етапом роботи є backtest-тестування на історичних даних та порівняння ефективності з базовими стратегіями (Buy and Hold, MA Crossover).

**Ключові слова:** Fuzzy C-Means, алгоритмічна торгівля, Forex, машинне навчання, нечітка кластеризація, технічні індикатори, backtesting, MetaTrader 5.

## ABSTRACT

Pavlenko V. V. A system for making trading decisions in financial markets based on fuzzy cluster analysis of time series. Master's Qualification Thesis in the specialty 126 – Information Systems and Technologies, educational program – Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2025. 139 p.

In Ukrainian language. Bibliography: 39 titles; Fig.: 23; table 17.

The master's qualification thesis is dedicated to the development of an automated trading system for the Forex currency market based on the Fuzzy C-Means clustering algorithm. The proposed system enables the identification of latent patterns in market data, generates trading signals based on membership degree calculations for BUY and SELL clusters, and adapts to changing market conditions through periodic model retraining.

This work provides an in-depth study of algorithmic trading challenges, analyzes technical analysis methods and machine learning approaches for financial time series forecasting. Particular attention is given to the mathematical formalization of the Fuzzy C-Means algorithm, selection of technical indicators for feature space construction, and design of a multi-filter system to reduce false signals. A complete development cycle has been implemented: from data preparation and model training to integration with the MetaTrader 5 trading platform via TCP/IP protocol. The final stage involves backtesting on historical data and performance comparison with baseline strategies (Buy and Hold, MA Crossover).

**Keywords:** Fuzzy C-Means, algorithmic trading, Forex, machine learning, fuzzy clustering, technical indicators, backtesting, MetaTrader 5.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ МЕТОДІВ. 9</b>	<b>9</b>
1.1 Аналіз проблеми та постановка задачі.....	9
1.2 Структура та особливості валютного ринку Форекс.....	11
1.3 Характеристики даних фінансових часових рядів.....	13
1.4 Аналіз існуючих методів ідентифікації торгових рішень.....	18
1.4.1 Технічний аналіз як основа для прийняття торгових рішень.....	18
1.4.2 Фундаментальний аналіз валютних ринків.....	21
1.4.3 Методи машинного навчання в алгоритмічній торгівлі.....	22
1.4.4 Методи нечіткої логіки у фінансовому прогнозуванні.....	25
1.4.5 Гібридні підходи як напрямок розвитку торгових систем.....	27
1.5 Аналіз подібних рішень.....	28
1.5.1 Комерційні торгові системи та їх обмеження.....	29
1.5.3 Відкриті бібліотеки та програмні інструменти.....	32
1.5.4 Порівняльний аналіз та виявлення прогалів.....	34
1.6 Висновки до розділу.....	35
<b>2 МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРИЙНЯТТЯ ТОРГОВИХ РІШЕНЬ.....</b>	<b>37</b>
2.2 Алгоритм Fuzzy C-Means.....	37
2.2.1 Постановка задачі нечіткої кластеризації.....	37
2.2.2 Цільова функція та параметри FCM.....	38
2.2.3 Алгоритм FCM.....	38
2.2.4 Ініціалізація та множинні рестарти.....	40
2.2.5 Переваги та обмеження FCM.....	40
2.3 Технічні індикатори та інші ознаки.....	41

	3
2.3.1 Relative Strength Index.....	41
2.3.2 Індикатор волатильності ATR.....	43
2.3.3 Moving Average Convergence Divergence.....	44
2.3.4 Moving Average Crossover.....	45
2.2.5 Stochastic Oscillator.....	45
2.2.5 Зміна ціни.....	46
2.4 Математична модель ідентифікації точок входу/виходу на ринку.....	47
2.4.1 Формалізація вхідних даних.....	47
2.4.2 Етап 1: Обчислення вектора ознак.....	47
2.4.3 Етап 2: Нормалізація ознак.....	48
2.4.4 Етап 3: Нечітка кластеризація (навчання та інференс).....	49
2.4.5 Етап 4: Прийняття торгового рішення.....	51
2.4.6 Багатофільтрова система валідації сигналів.....	52
2.4.8 Переваги та обмеження запропонованої математичної моделі.....	54
2.5 Алгоритмічна реалізація системи прийняття рішень.....	56
2.6 Висновки до розділу.....	61
<b>3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ПРИЙНЯТТЯ ТОРГОВИХ РІШЕНЬ.....</b>	<b>63</b>
3.1 Вибір інструментів розробки.....	63
3.2 Архітектура програмного забезпечення.....	65
3.2.1 Шар даних.....	68
3.2.2 Шар формування ознак.....	69
3.2.3 Шар стратегії.....	71
3.2.4 Шар виконання.....	73
3.2.5 Шар оцінки.....	77
3.3 Тестування системи на історичних даних.....	77
3.4 Висновки до розділу.....	83

<b>4 ЕКОНОМІЧНИЙ РОЗДІЛ.....</b>	<b>86</b>
4.1 Проведення комерційного та технологічного аудиту.....	86
4.2 Розрахунок узагальненого коефіцієнта якості розробки.....	87
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	88
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт.	95
4.3.7 Амортизація обладнання, програмних засобів та приміщень.....	96
4.3.8 Паливо та енергія для науково-виробничих цілей.....	97
4.3.9 Службові відрядження.....	99
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	99
4.3.12 Накладні (загальновиробничі) витрати.....	100
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	101
4.5 Висновки до розділу.....	106
<b>ВИСНОВКИ.....</b>	<b>108</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>111</b>
<b>ДОДАТКИ.....</b>	<b>116</b>
Додаток А (обов'язковий) Технічне завдання.....	117
Додаток Б (обов'язковий) Ілюстративна частина.....	123
Додаток В (обов'язковий) Лістинг програмного забезпечення.....	128
Додаток Г (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	139

## ВСТУП

*Актуальність роботи.* У сучасних умовах глобалізації фінансових ринків та стрімкого розвитку інформаційних технологій особливого значення набуває автоматизація торговельних процесів на валютному ринку Forex. Традиційні методи ручної торгівлі, які базуються на суб'єктивному аналізі графіків та емоційному прийнятті рішень, є неефективними, часто призводять до помилок, викликаних людським фактором (страх, жадібність, втома), а також не забезпечують необхідної швидкості реагування на динамічні зміни цін. Впровадження алгоритмічних торговельних систем дозволяє вирішити ряд критичних проблем: емоційність прийняття рішень, недостатню швидкість аналізу великих обсягів даних, неможливість безперервного моніторингу ринку, а також відсутність систематичності в застосуванні торгових правил.

Валютний ринок Forex є найбільшим та найліквіднішим фінансовим ринком у світі з середньодобовим обсягом торгів понад 7 трильйонів доларів США. Його висока волатильність створює як можливості для прибутку, так і значні ризики для трейдерів. Сучасні алгоритмічні системи повинні не лише генерувати торгові сигнали, але й адаптуватися до змінних ринкових умов (тренд, флет, підвищена волатильність), враховувати множину факторів (технічні індикатори, ринкові сесії, макроекономічні події) та забезпечувати ефективне управління ризиками через stop-loss та take-profit механізми.

Методи машинного навчання, зокрема алгоритми кластеризації, демонструють значний потенціал для виявлення прихованих закономірностей у фінансових часових рядах. Алгоритм Fuzzy C-Means (FCM) є особливо перспективним, оскільки, на відміну від жорсткої кластеризації (k-means), він дозволяє кожному об'єкту (ринковому стану) належати до кількох кластерів одночасно з різними ступенями належності. Це відображає реальну природу фінансових ринків, де чіткі межі між станами (бичачий/ведмежий тренд) є

розмитими, а перехідні стани мають високу значущість для прийняття торгових рішень.

Крім того, важливо враховувати зростаючі вимоги до прозорості та відтворюваності результатів алгоритмічних торговельних систем. Використання чітких математичних формулювань, стандартизованих технічних індикаторів (RSI, ATR, MACD) та об'єктивних метрик оцінки ефективності (Sharpe Ratio, Max Drawdown, Win Rate) забезпечує можливість наукового обґрунтування ефективності розробленої системи через порівняння з базовими стратегіями (Buy and Hold, MA Crossover).

Таким чином, впровадження торговельної системи на основі алгоритму Fuzzy C-Means не лише сприяє підвищенню прибутковості торгівлі та мінімізації емоційного впливу на прийняття рішень, але й дозволяє трейдерам отримувати надійну статистичну інформацію для подальшого аналізу, оптимізації параметрів моделі та прийняття обґрунтованих інвестиційних рішень на валютному ринку Forex.

*Метою роботи є розробка автоматизованої торговельної системи для валютного ринку Forex на основі алгоритму нечіткої кластеризації Fuzzy C-Means для підвищення ефективності торгівлі за метриками прибутковості та ризику порівняно з базовими стратегіями.*

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

1. Провести аналіз існуючих підходів до алгоритмічної торгівлі на валютному ринку Forex та методів машинного навчання для прогнозування фінансових часових рядів;
2. Дослідити математичний апарат алгоритму Fuzzy C-Means та його застосування для кластеризації багатовимірних часових рядів;
3. Обґрунтувати вибір технічних індикаторів для формування простору ознак, що описують поточний стан ринку (RSI, ATR, Price Change, MACD Crossover, MA Crossover, Stochastic %K);

4. Розробити математичну модель генерації торгових сигналів на основі розрахунку ступенів належності до кластерів BUY та SELL з використанням гістерезисного фільтра;
5. Спроекувати та реалізувати багатофільтрову систему валідації сигналів (ATR Filter, Trend Filter, Session Filter, Min Position Duration) для зменшення overtrading та хибних сигналів;
6. Розробити програмне забезпечення торговельної системи з використанням Python та бібліотек scikit-learn, scikit-fuzzy, pandas, numpy;
7. Провести backtest-тестування системи на історичних даних валютних пар EURUSD та GBPUSD (таймфрейм M15) з розрахунком ключових метрик ефективності;

*Об'єктом дослідження* є процеси прийняття торгових рішень на валютному ринку Forex на основі аналізу часових рядів ціноутворення.

*Предметом дослідження* є методи нечіткої кластеризації Fuzzy C-Means та їх застосування для побудови автоматизованих торговельних систем на валютному ринку Forex.

*Методи дослідження.* У роботі використовуються такі методи дослідження як теорія нечітких множин, методи машинного навчання без вчителя, технічний аналіз фінансових ринків, методи оптимізації, Backtesting та імітаційне моделювання, статистичний аналіз, об'єктно-орієнтоване проектування, методи мережевого програмування.

*Науково-технічний результат* полягає у створенні підходу до ідентифікації торгових рішень, який поєднує алгоритм нечіткої кластеризації Fuzzy C-Means з багатофільтровою системою валідації сигналів та адаптивними порогами гістерезису, що дозволяє підвищити стабільність торгівлі та зменшити кількість хибних сигналів.

*Практична цінність роботи* полягає у створенні системи прийняття торгових рішень на фінансових ринках, яка може використовуватися для автоматизації торгівлі на валютному ринку Форекс, що дозволяє зменшити вплив емоційного фактору, підвищити швидкість реакції на ринкові зміни та

забезпечити цілодобову роботу без втручання людини.

*Апробація результатів роботи.* Основні положення й результати роботи були представлені автором на Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026) [1].

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ МЕТОДІВ

## 1.1 Аналіз проблеми та постановка задачі

Фінансові ринки, зокрема валютний ринок Forex, характеризуються високою динамікою цінових коливань, що створює можливості для отримання прибутку від торгівлі валютними парами, такими як EUR/USD чи USD/JPY. Ефективність торгівлі значною мірою залежить від точного визначення моментів входу на ринок та виходу з нього, що дозволяє оптимізувати прибуток та мінімізувати потенційні збитки. Традиційні методи прогнозування, такі як ковзна середня, ARMA та ARIMA, спрямовані на оцінку майбутньої ціни на один крок вперед. Проте ці методи мають обмежену застосовність у торгівлі, оскільки не враховують складність ринкових умов та не забезпечують стратегічних рішень щодо управління ризиками.

Основною проблемою є те, що торгівля вимагає не лише точного прогнозування ціни, але й своєчасного визначення оптимальних моментів для відкриття та закриття угод. Цінові коливання між точками входу та виходу можуть бути значними та непередбачуваними, що ускладнює використання одноетапних прогнозів. Дослідження, проведені у сфері фінансового моделювання, наприклад, у статті [2], свідчать, що класичні моделі, оптимізовані за методом найменших квадратів, досягають точності прогнозування напрямку зміни ціни лише на рівні трохи вище 50%, що є недостатнім для ефективної торгівлі на ринках із високою волатильністю, як Forex.

Альтернативним підходом є технічний аналіз, який включає оцінку ключових моментів на графіках цінових рядів із використанням індикаторів, таких як рухомі середні чи рівні підтримки та опору. Цей метод дозволяє враховувати не лише ціну, але й такі фактори, як волатильність та ринкові тренди. Проте, як зазначається в дослідженні [3], традиційні техніки технічного

аналізу забезпечують перевагу лише в 55-60% випадків, що вимагає інтеграції з додатковими аналітичними методами для підвищення ефективності.

Задача ідентифікації торгових рішень полягає у визначенні моментів входу та виходу з ринку, що включає два основні типи транзакцій: купівлю з подальшим продажем за вищою ціною або продаж з подальшою купівлею за нижчою ціною. Своєчасність таких рішень безпосередньо впливає на прибутковість торгових стратегій. Для підвищення ефективності аналізу ринкових даних пропонуються комбіновані підходи, наприклад, використання кількох таймфреймів, де більший масштаб, як-от 4-годинний (H4), слугує фільтром для сигналів меншого масштабу, як-от 15-хвилинний (M15), що зменшує кількість хибних угод, як показано в роботі [4].

Сучасні технології, такі як нейронні мережі та нечітка логіка, дозволяють моделювати складні нелінійні залежності та враховувати невизначеність ринкових даних. Проте їх ефективне застосування потребує поєднання аналітичних методів із практичним досвідом торгівлі, орієнтованим на максимізацію прибутку. У цьому контексті постає задача розробки методів та програмних засобів для ідентифікації торгових рішень у фінансових часових рядах із використанням нечіткого кластерного аналізу. Метою є створення системи, яка базується на нечіткому кластерному аналізі з інтеграцією технічних індикаторів та оцінюється за фінансовими метриками, такими як чистий прибуток, максимальне зниження капіталу та відсоток успішних угод, замість традиційних метрик точності прогнозування.



Рисунок 1.1 – Приклад графіка валютної пари EUR/USD на 4х-годинному таймфреймі з позначками моментів входу "Buy" та виходу "Sell", що ілюструє типові торгові рішення

## 1.2 Структура та особливості валютного ринку Форекс

Валютний ринок Форекс є найбільшим фінансовим ринком світу з щоденним обсягом торгів понад 7,5 трильйонів доларів США [5]. Це децентралізований ринок, що функціонує 24 години на добу п'ять днів на тиждень завдяки послідовному переходу торгових сесій між світовими фінансовими центрами.

Азіатська сесія охоплює Токіо, Сідней та інші азіатські фінансові центри, функціонуючи з 00:00 до 09:00 GMT. Цей період характеризується помірною волатильністю та меншими обсягами торгів порівняно з іншими сесіями. Основна активність зосереджена навколо японської йени та австралійського долара, а торгівля EUR/USD у цей час часто відбувається у вузьких діапазонах.

Європейська сесія (08:00-17:00 GMT) вважається найліквіднішою, адже включає Лондон - найбільший валютний центр світу, а також Франкфурт, Цюрих та інші європейські майданчики. Саме в цей період EUR/USD демонструє найбільшу активність та найвужчі спреди. Лондонська сесія

традиційно встановлює тон для денної торгівлі, оскільки європейські трейдери реагують на азіатські новини та готуються до американського відкриття.

Американська сесія з центром у Нью-Йорку працює з 13:00 до 22:00 GMT і відзначається високою волатильністю, особливо під час виходу важливих економічних новин зі США. Найцікавішим є період перетину з європейською сесією (13:00-17:00 GMT), коли на ринку одночасно активні трейдери з обох континентів, що забезпечує максимальну ліквідність та найбільші цінові рухи.

Унікальною особливістю Форекс є відсутність центрального біржового майданчика. Торгівля здійснюється через мережу банків, брокерів та електронних торгових систем, що створює унікальну екосистему з безперервною ліквідністю та конкурентним ціноутворенням. Ця децентралізована структура забезпечує можливість торгівлі з кредитним плечем до 1:500 для роздрібних інвесторів та мінімальні транзакційні витрати.



Рисунок 1.2 – Активність торгових сесій та волатильність EUR/USD протягом доби

Ієрархія учасників ринку включає центральні банки, які формують монетарну політику та здійснюють валютні інтервенції, великі комерційні

банки як основні постачальники ліквідності, хедж-фонди та інвестиційні банки для спекулятивної торгівлі, корпорації для хеджування валютних ризиків, та роздрібних трейдерів як індивідуальних учасників ринку.

Розвиток технологій кардинально змінив доступ до валютного ринку. Електронні платформи типу MetaTrader 4/5 [6], cTrader [7] дозволили роздрібним трейдерам отримати доступ до інституційних спредів (від 0.1 пункта для EUR/USD), швидкість виконання ордерів до 100 мілісекунд та можливість алгоритмічної торгівлі через Expert Advisors. Це призвело до демократизації валютної торгівлі, коли індивідуальні інвестори змогли конкурувати з професійними трейдерами на більш рівних умовах.

### 1.3 Характеристики даних фінансових часових рядів

Величезну кількість даних, які виробляють фінансові ринки, можна умовно розділити на фундаментальні та ринкові дані. Фундаментальні дані складаються з деталей щодо фінансового стану організації, таких як прибуток, дохід та інші бухгалтерські показники. Для різних фінансових інструментів ринкові дані включають інформацію про ціну та обсяг, а також дані стаття замовлень, які відображають рівні попиту та продажу. Інтервал даних фінансового ринку може бути високочастотним на рівні тиків, що вказує на окремі угоди, або агрегованим на щоденній, щотижневій або щомісячній основі. Високочастотні дані пропонують більш детальне розуміння динаміки ринку, але також створюють труднощі через шум та аномалії, спричинені впливом мікроструктури ринку.

Прогнозування фінансових часових рядів є підмножиною аналізу часових рядів, який загалом є складним завданням. Зокрема, прогнозування фінансових часових рядів включає прогнозування майбутніх значень фінансових показників, таких як ціни на акції, обмінні курси та ціни на товари, на основі історичних даних.

Через свою притаманну складність та специфічні риси ці часові ряди створюють особливі труднощі для прогнозування. Властивості даних фінансових часових рядів включають, але не обмежуються наступним:

1. **Нелінійність та нестационарність:** Нелінійність означає, що дані не можуть бути представлені за допомогою прямої лінії (тобто лінійно); нестационарність означає, що їхні статистичні властивості змінюються з часом [8] (рис. 1.3). Відомо, що дані фінансових часових рядів є нелінійними та часто не мають стаціонарності через наявність тенденцій, сезонності та нерегулярних коливань, спричинених іншими змінними закономірностями, такими як поточні події в економіці, зміни в настроях інвесторів тощо. Нелінійність та нестационарність підвищують складність даних, що створює труднощі при застосуванні традиційних методів аналізу часових рядів, які припускають лінійність або постійні статистичні властивості з часом.

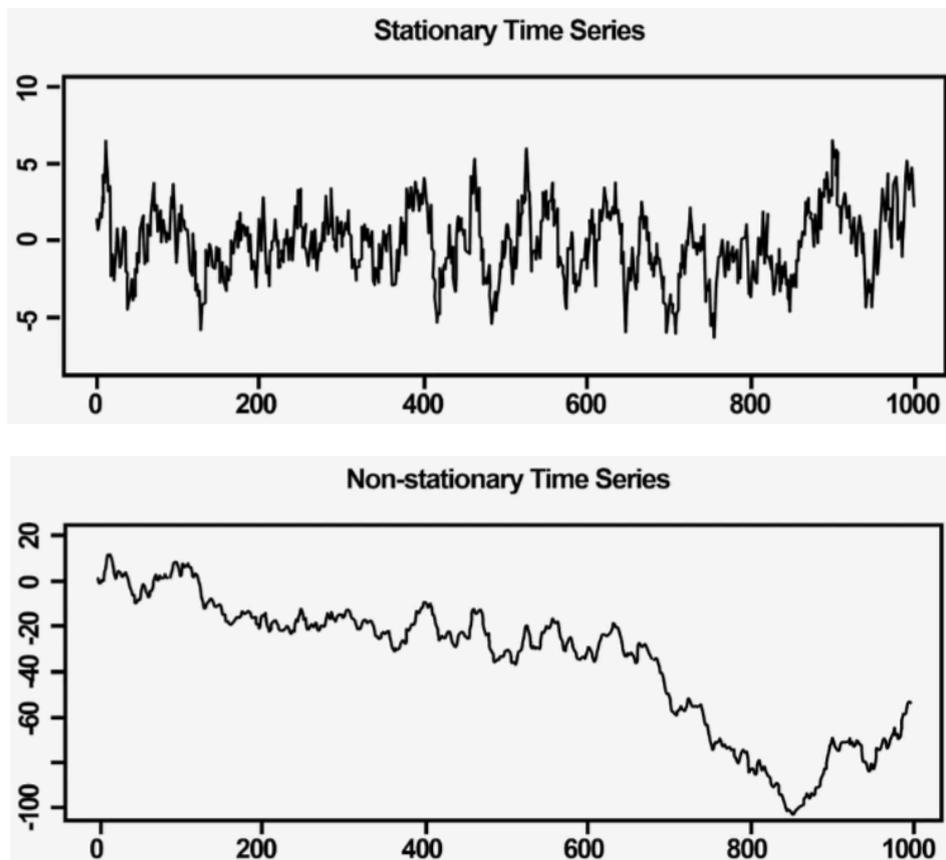


Рисунок 1.3 – Стаціонарність та нестационарність часових рядів

**2. Короткострокові та довгострокові залежності:** Фінансові часові ряди мають внутрішню часову природу, оскільки кожен запис має залежності, короткострокові або довгострокові, від попередніх записів. Короткострокові залежності стосуються внутрішньоденних коливань та швидких рухів ринку, тоді як довгострокові залежності стосуються тенденцій та закономірностей, які тривають тижні, місяці або навіть роки. Було виявлено, що нехтування часовими залежностями може призвести до поганої прогнозованої ефективності [9, 10].

**3. Асиметрія, товсті хвости та степеневий спад:** Традиційно багато фінансових моделей, включаючи деякі моделі ціноутворення опціонів та моделі ризику, спиралися на припущення про нормальні розподіли, що було поставлено під сумнів численними дослідниками та фінансовими фактами в останні роки. Натомість асиметричні розподіли та товсті хвости часто спостерігаються у фінансових часових рядах, наприклад, у прибутковості акцій. Емпірично встановлено, що розподіл ймовірностей цінової прибутковості має степеневий спад у хвостах [11, 12]. Це означає, що екстремальні події, такі як обвали або сплески ринку, трапляються частіше, ніж передбачає нормальний розподіл. Традиційні моделі, що припускають нормальні розподіли, можуть не точно відображати ці виняткові події.

**4. Волатильність:** Волатильність стосується ступеня варіації або дисперсії фінансового часового ряду. Фінансові ринки відомі тим, що переживають періоди високої волатильності, за якими слідує відносно спокійніші періоди (рис. 1.4). Інше явище, яке спостерігається, називається кластеризацією волатильності, яке виникає, коли періоди високої волатильності змінюються іншими з аналогічно високою волатильністю, і навпаки [13]. Кластеризацію волатильності можна кількісно визначити як степеневий запад автокореляційної функції абсолютної цінової дохідності [12].

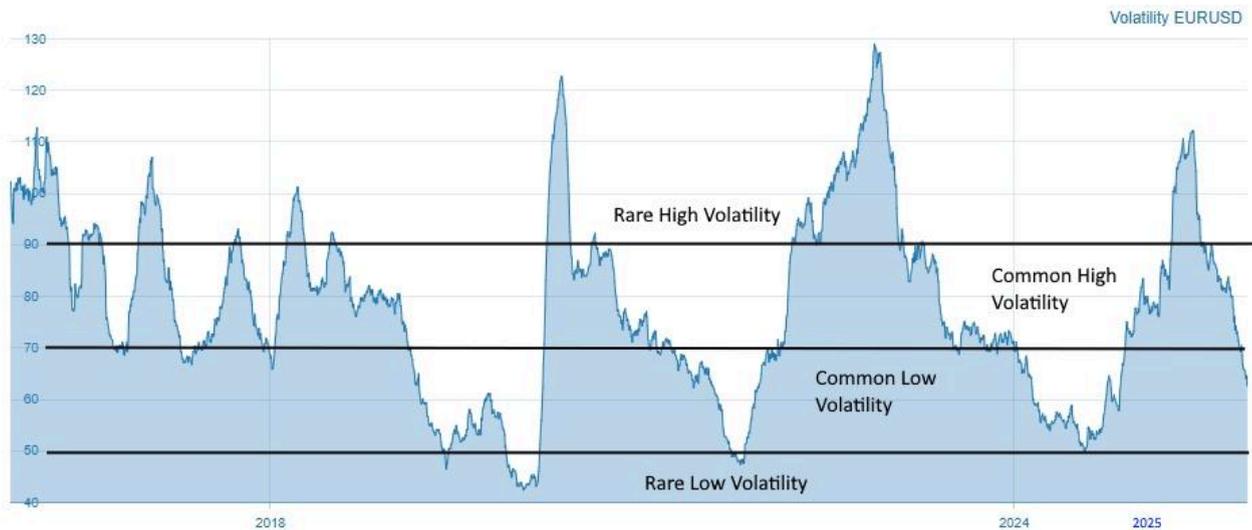


Рисунок 1.4 – Волатильність валютної пари EUR/USD

**5. Автокореляція та взаємна кореляція:** Автокореляція у фінансових часових рядах може виникати, стосуючись кореляції поточних значень часового ряду з його історичними значеннями [14]. Основною причиною може бути довгостроковий вплив новин, настроїв ринку тощо. Крім того, численні фінансові активи або інструменти можуть демонструвати взаємозалежність, що відомо як взаємна кореляція [15]. Взаємна кореляція між фінансовими часовими рядами показує, як зміни ціни одного активу впливають на ціну іншого.

**6. Ефекти левериджу:** Ефекти левериджу описують негативний зв'язок між вартістю активів та волатильністю. Спостерігається, що негативні шоки, як правило, мають більший вплив на волатильність, ніж позитивні шоки такої ж величини [12]. У фінансових часових рядах зв'язок між дохідністю та волатильністю може бути асиметричним, причому рухи вниз призводять до збільшення волатильності.

**7. Поведінкова та подієва динаміка:** Поведінка інвесторів, настрої та психологічні змінні, а також події (рис. 1.5), що впливають на ринок, такі як оголошення про прибутки, урядові рішення, геополітичні події та прес-релізи, впливають на фінансові ринки [16]. Ці поведінкові та подієві характеристики можуть спричинити значні зрушення та ірраціональні рухи у фінансових часових рядах.



Рисунок 1.5 – Значне падіння пари BTC/USD після заяви Трампа в жовтні 2025 року

Зазвичай ефективне включення предметних знань, включаючи ці статистичні властивості, у прогностичні моделі має вирішальне значення для точного аналізу та прогнозування фінансових часових рядів. Останнім часом моделі глибокого навчання та гібридні підходи все частіше розробляються та використовуються для вирішення цих складнощів та фіксації основної динаміки фінансових ринків.

Альтернативні джерела даних, які включають нетрадиційні дані, такі як настрої з соціальних мереж та моделі веб-трафіку, набули популярності в останні роки для завдання прогнозування фінансових часових рядів. Ці альтернативні джерела даних мають на меті надати додаткову інформацію, яка може не бути одразу відображена в традиційних фундаментальних або ринкових даних.

## 1.4 Аналіз існуючих методів ідентифікації торгових рішень

Розвиток автоматизованих торгових систем супроводжується постійним пошуком оптимальних методів ідентифікації точок входу та виходу з ринку. Історично склалися три основні напрямки аналізу фінансових ринків: технічний, фундаментальний та методи на основі штучного інтелекту. Кожен з цих підходів має власну філософію, математичний апарат та область застосування. У даному підрозділі проведемо детальний аналіз існуючих методів, визначимо їхні переваги та обмеження, а також обґрунтуємо доцільність використання нечіткої кластеризації для вирішення поставленої задачі.

### 1.4.1 Технічний аналіз як основа для прийняття торгових рішень

Технічний аналіз є найпоширенішим підходом серед трейдерів і базується на фундаментальному припущенні, що вся інформація про актив вже відображена в його ціні. Цей підхід виник ще в кінці XIX століття завдяки роботам Чарльза Доу, який сформулював теорію Dow Theory – перший системний підхід до аналізу ринкових тенденцій. З того часу технічний аналіз еволюціонував, інтегруючи математичні методи статистики, теорії ймовірностей та обробки сигналів [3].

Філософія технічного аналізу ґрунтується на трьох ключових постулатах. По-перше, ціна враховує все – це означає, що всі економічні, політичні, психологічні та інші фактори вже відображені в поточному ціновому рівні. По-друге, ціни рухаються в трендах, які мають тенденцію до продовження. По-третє, історія повторюється – поведінка учасників ринку циклічна, що дозволяє використовувати історичні патерни для прогнозування майбутніх рухів.

Графічний аналіз та розпізнавання патернів

Одним з найдавніших напрямків технічного аналізу є графічний аналіз, який передбачає візуальне виявлення характерних формацій на цінових графіках. Трейдери ідентифікують тренди (висхідні, низхідні або бічні), фігури продовження тренду (такі як прапорці, вимпели, трикутники) та фігури розвороту (голова і плечі, подвійна вершина/дно). Особливе місце займає японський свічковий аналіз (Japanese Candlesticks), розроблений ще в XVIII столітті рисовими трейдерами Японії. Свічкові патерни, такі як hammer (молот), doji (дожі), engulfing (поглинання), надають інформацію про психологію учасників ринку в конкретний момент часу [3].

Однак графічний аналіз має суттєве обмеження – високий ступінь суб'єктивності. Різні трейдери можуть по-різному інтерпретувати ту саму графічну формацію, що унеможливорює повну автоматизацію такого підходу. Відсутність строгих математичних критеріїв ідентифікації патернів призводить до неузгодженості в прийнятті торгових рішень.

#### Технічні індикатори та їх математична основа

Для подолання суб'єктивності графічного аналізу було розроблено систему технічних індикаторів – математичних функцій, обчислених на основі історичних даних про ціну та обсяг торгів. Технічні індикатори можна класифікувати за їхнім призначенням.

Трендові індикатори призначені для ідентифікації напрямку та сили ринкового тренду. Ковзні середні (Moving Averages) – найпростіша, але водночас найпопулярніша група індикаторів. Просте ковзне середнє (SMA) обчислюється як арифметичне середнє цін за визначений період, тоді як експоненційне ковзне середнє (EMA) надає більшу вагу останнім спостереженням. Класичний торговий сигнал генерується при перетині ціною ковзного середнього або при перетині швидкого МА з повільним (відоме як Golden Cross для бичачого сигналу та Death Cross для ведмежого). Індикатор MACD (Moving Average Convergence Divergence), розроблений Джеральдом Аппелем, виявляє зміну імпульсу тренду через різницю між двома експоненційними ковзними середніми [3].

Осцилятори особливо корисні в умовах флетового (бічного) ринку, коли ціна коливається в певному діапазоні без чіткого тренду. Індекс відносної сили RSI (Relative Strength Index), запропонований Уелсом Уайлдером у 1978 році, вимірює швидкість і величину цінових змін, приймаючи значення в діапазоні від 0 до 100. Традиційно рівні вище 70 інтерпретуються як зона перекупленості (потенційний сигнал до продажу), а рівні нижче 30 – як зона перепроданості (сигнал до купівлі). Стохастичний осцилятор (Stochastic Oscillator), розроблений Джорджем Лейном, порівнює поточну ціну закриття з діапазоном цін за визначений період, виявляючи моменти потенційного розвороту [3].

Індикатори волатильності вимірюють ступінь коливань ціни, що є критично важливим для управління ризиками. Смуги Боллінджера (Bollinger Bands) будуються як ковзне середнє плюс-мінус два стандартних відхилення, створюючи динамічний конверт навколо ціни. Розширення смуг сигналізує про зростання волатильності, тоді як звуження вказує на період консолідації, після якого часто настає значний ціновий рух. Середній істинний діапазон ATR (Average True Range) використовується не для генерації торгових сигналів, а для калібрування рівнів Stop-Loss відповідно до поточної волатильності ринку [3].

#### Обмеження технічного аналізу

Незважаючи на широке застосування, технічний аналіз має ряд фундаментальних обмежень. Більшість індикаторів є лагуючими (lagging indicators) – вони запізнюються за ринком, оскільки базуються на минулих даних. Це призводить до ситуації, коли сигнал генерується вже після того, як значна частина руху вже відбулася. У флетовому ринку технічні індикатори генерують багато хибних сигналів (false signals), що призводить до надмірної торгівлі (overtrading) та збитків від комісій. Крім того, технічний аналіз не враховує фундаментальні фактори, такі як рішення центральних банків щодо процентних ставок або макроекономічні шоки, які можуть кардинально змінити ринкову ситуацію [3].

Суб'єктивний вибір параметрів індикаторів (наприклад, період для RSI чи довжина ковзного середнього) створює проблему оптимізації під історичні дані

(curve-fitting), що погіршує узагальнюючу здатність системи на нових даних. Емпіричні дослідження показують, що стратегії, оптимізовані на історичних даних, часто демонструють значно гіршу ефективність на out-of-sample періоді.

#### 1.4.2 Фундаментальний аналіз валютних ринків

На протипагу технічному аналізу, фундаментальний аналіз фокусується на визначенні внутрішньої (справедливої) вартості активу на основі економічних, фінансових та геополітичних факторів. Для валютного ринку Форекс фундаментальний аналіз має свою специфіку, оскільки валютний курс відображає відносну економічну силу двох країн.

##### Макроекономічні драйвери валютних курсів

Процентні ставки центральних банків є одним з найважливіших факторів, що впливають на валютні курси. Згідно з теорією паритету процентних ставок (Interest Rate Parity), валюта країни з вищою процентною ставкою повинна зміцнюватися відносно валюти країни з нижчою ставкою через приплив іноземного капіталу. Трейдери уважно стежать за рішеннями центральних банків (Federal Reserve, Європейський центральний банк, Банк Англії) та риторикою їхніх представників. "Яструбина" (hawkish) риторика, що вказує на ймовірне підвищення ставок, зміцнює валюту, тоді як "голубина" (dovish) позиція призводить до ослаблення [3].

Інфляція безпосередньо впливає на купівельну спроможність валюти. Висока інфляція призводить до девальвації валюти, оскільки зменшується її реальна вартість. Центральні банки використовують процентні ставки як інструмент контролю інфляції – підвищення ставок охолоджує економіку та знижує інфляцію. Показники інфляції, такі як Індекс споживчих цін (CPI) та Індекс цін виробників (PPI), є ключовими макроекономічними релізами, що викликають значну волатильність на валютному ринку.

Зростання валового внутрішнього продукту (ВВП) сигналізує про економічну силу країни і, як правило, супроводжується зміцненням національної валюти. Інвестори віддають перевагу валютам країн з стабільним економічним зростанням. Торговий баланс (співвідношення експорту та імпорту) також відіграє важливу роль – профіцит торгового балансу створює попит на національну валюту від іноземних покупців, що призводить до її зміцнення. Рівень безробіття є індикатором здоров'я ринку праці та економіки загалом. Публікація звіту Non-Farm Payrolls (зміна зайнятості в несільськогосподарському секторі США) щомісяця викликає одні з найсильніших коливань на валютному ринку.

Обмеження фундаментального аналізу для автоматизованих систем.

Попри важливість для розуміння довгострокових трендів, фундаментальний аналіз має суттєві обмеження для застосування в автоматизованих торгових системах. Він не надає конкретних точок входу та виходу з ринку – знання про те, що валюта недооцінена, не дає відповіді на питання, коли саме відбудеться корекція. Ринки можуть тривалий час реагувати ірраціонально, керуючись спекулятивними настроями (market sentiment), що суперечить фундаментальній оцінці. Інформація про економічні показники часто публікується з запізненням (дані ВВП за квартал стають доступними через місяць), що знижує її актуальність. Найважливіше – формалізація фундаментальних факторів у вигляді математичних правил для автоматизованої системи є вкрай складною задачею [3].

#### 1.4.3 Методи машинного навчання в алгоритмічній торгівлі

Революція в обчислювальних потужностях та доступності великих обсягів даних у XXI столітті призвела до активного застосування методів машинного навчання (Machine Learning, ML) у фінансовій індустрії. На відміну

від традиційних підходів, ML-алгоритми здатні автоматично виявляти складні нелінійні залежності в даних без явного програмування правил.

#### Класичні алгоритми машинного навчання

Лінійна та логістична регресія є найпростішими ML-методами, що використовуються для прогнозування майбутньої ціни або ймовірності руху в певному напрямку. Лінійна регресія моделює ціну як лінійну комбінацію предикторів (технічних індикаторів, лагованих цін). Однак фундаментальне обмеження цих методів полягає в припущенні лінійності залежностей, що рідко виконується на фінансових ринках. Крім того, мультиколінеарність між індикаторами (висока кореляція між предикторами) призводить до нестабільності оцінок параметрів [17].

Дерева рішень (Decision Trees) та їх ансамблі, зокрема Random Forest, здатні моделювати нелінійні залежності та взаємодії між змінними [18]. Random Forest створює множину дерев рішень на випадкових підвибірках даних та ознак, після чого усереднює їхні прогнози, що значно зменшує проблему перенавчання (overfitting). Важливою перевагою є можливість оцінки важливості ознак (feature importance), що допомагає зрозуміти, які індикатори найбільше впливають на прогноз.

Метод опорних векторів (Support Vector Machines, SVM) використовується для класифікації напрямку руху ціни (вгору/вниз). SVM знаходить оптимальну гіперплощину, що максимізує відстань до найближчих точок класів (margin). Kernel trick дозволяє SVM моделювати нелінійні залежності через неявне відображення даних у простір вищої розмірності. SVM показують хороші результати на датасетах малого та середнього обсягу, але їх обчислювальна складність зростає квадратично з розміром даних [19].

#### Нейронні мережі та глибоке навчання

Штучні нейронні мережі, інспіровані структурою людського мозку, є потужними універсальними апроксиматорами нелінійних функцій. Багатошарові перцептрони (Multi-Layer Perceptrons, MLP) складаються з кількох шарів нейронів, де кожен нейрон обчислює зважену суму входів з

наступною нелінійною функцією активації. Однак MLP не враховують темпоральну структуру часових рядів, оскільки обробляють вхідні дані як незалежні спостереження.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) спеціально розроблені для роботи з послідовностями. Вони мають зворотні зв'язки, що дозволяє "пам'ятати" попередні стани. Однак класичні RNN страждають від проблеми зникаючого градієнта (vanishing gradient), що унеможливорює навчання довгострокових залежностей. Архітектура LSTM (Long Short-Term Memory), запропонована Hochreiter & Schmidhuber (1997), вирішує цю проблему за допомогою спеціальних гейтів (gates), що контролюють потік інформації [20].

Згорткові нейронні мережі (Convolutional Neural Networks, CNN), класично застосовувані для обробки зображень, знаходять несподіване застосування у фінансах. Дослідники перетворюють часові ряди на зображення (наприклад, графіки свічок) та застосовують CNN для виявлення візуальних патернів. Інший підхід полягає у використанні 1D-згорток для виявлення локальних темпоральних патернів у часових рядах [21].

Навчання з підкріпленням як парадигма для торгівлі

Навчання з підкріпленням (Reinforcement Learning, RL) пропонує принципово інший підхід, де торгова система (агент) навчається оптимальній поведінці через взаємодію з середовищем (ринком). Агент спостерігає стан ринку (ціни, індикатори, поточна позиція), виконує дію (купити, продати, утримувати) та отримує винагороду (прибуток/збиток). Мета навчання – знайти політику (policy), що максимізує кумулятивну винагороду.

Перевагою RL є безпосередня оптимізація цільової метрики (прибуток) з урахуванням транзакційних витрат та інших ринкових реалій. Алгоритми Q-Learning, Deep Q-Networks (DQN), Actor-Critic методи (A2C, A3C, PPO) активно досліджуються в контексті торгівлі. Однак RL має значні недоліки: потребує величезних обсягів даних для навчання, схильний до нестабільності (велика варіантність результатів між запусками), складність налаштування

гіперпараметрів. Крім того, RL-агенти часто експлуатують особливості симуляційного середовища, які не присутні в реальній торгівлі [22].

#### Проблема перенавчання та узагальнення

Фундаментальною проблемою застосування машинного навчання до фінансових часових рядів є перенавчання (overfitting) – модель занадто добре "запам'ятовує" тренувальні дані, втрачаючи здатність узагальнювати на нових даних. Це особливо критично для фінансів через нестационарність часових рядів – статистичні властивості ринку змінюються з часом, тому минулі патерни не обов'язково повторюватимуться в майбутньому.

Дослідження Sezer et al. (2020) у систематичному огляді застосування глибокого навчання до фінансових часових рядів виділило основні виклики: нестационарність, низьке відношення сигнал/шум, обмежений обсяг даних (порівняно з задачами комп'ютерного зору), складність оцінки реальної ефективності через look-ahead bias та survivorship bias [23].

#### 1.4.4 Методи нечіткої логіки у фінансовому прогнозуванні

Нечітка логіка (Fuzzy Logic), запропонована Лотфі Заде в 1965 році, пропонує альтернативну парадигму для роботи з невизначеністю. На відміну від класичної булевої логіки, де твердження є або істинним, або хибним, нечітка логіка дозволяє проміжні ступені істинності. Це відповідає природі фінансових ринків, де поняття "висока волатильність", "сильний тренд", "перекупленість" не мають чітких меж.

##### Теоретичні основи нечіткої логіки.

Центральним поняттям є нечітка множина (Fuzzy Set), де елемент має ступінь належності  $\mu(x) \in [0, 1]$  до множини. Наприклад, для поняття "перекупленість" на основі індикатора RSI можна визначити функцію належності:  $\mu_{\text{overbought}}(\text{RSI}=50) = 0$  (ринок абсолютно не перекуплений),  $\mu_{\text{overbought}}(\text{RSI}=70) = 0.5$  (помірна перекупленість),  $\mu_{\text{overbought}}(\text{RSI}=90) = 1$

(повна перекупленість). Така градуальність краще відображає реальність, де ринкові стани змінюються поступово, а не стрибкоподібно [24].

Нечіткі правила (Fuzzy Rules) формалізують експертні знання у вигляді конструкцій IF-THEN з нечіткими умовами. Наприклад: "ЯКЩО RSI є ВИСОКИМ І MACD є ПОЗИТИВНИМ, ТО Сигнал є СИЛЬНА\_КУПІВЛЯ". Процес прийняття рішення складається з трьох етапів: фазифікація (перетворення чітких значень індикаторів у ступені належності до нечітких множин), нечітка інференція (застосування нечітких правил та агрегація результатів), дефазифікація (перетворення нечіткого виходу в чітке торгове рішення) [25].

Нечітка кластеризація Fuzzy C-Means.

Алгоритм Fuzzy C-Means (FCM), розроблений Джеймсом Без деком у 1981 році, є найпопулярнішим методом нечіткої кластеризації. На відміну від жорсткої кластеризації (наприклад, K-Means), де кожна точка даних належить тільки одному кластеру, FCM дозволяє часткову належність – кожна точка має ступінь належності до всіх кластерів, причому сума ступенів належності дорівнює 1 [26].

Застосування FCM у торгівлі має глибокий економічний зміст. Фінансові ринки рідко перебувають у "чистих" станах – замість чіткого розділення на "тренд" та "флет" реальність часто є проміжною. FCM дозволяє ідентифікувати ринкові режими (бичачий ринок, ведмежий ринок, консолідація, висока/низька волатильність) та визначити ступінь належності поточного стану до кожного режиму. Це надає цінну інформацію для прийняття торгових рішень – наприклад, якщо ступінь належності до кластеру "сильний бичачий тренд" становить 0.85, це є надійнішим сигналом для відкриття позиції на купівлю, ніж при значенні 0.55.

Lai et al. (2006) використали FCM для кластеризації торгових днів фондового ринку на основі технічних індикаторів та адаптації торгової стратегії до виявленого режиму. Результати показали покращення ефективності порівняно з універсальною стратегією [27]. Boyacıoglu & Avcı (2010)

застосували FCM до фінансових показників компаній для прогнозування кредитних рейтингів та ідентифікації потенційних банкрутств, досягнувши високої точності класифікації [28].

Переваги нечіткої логіки для фінансових застосувань

Нечітка логіка має ряд переваг, що роблять її привабливою для фінансових застосувань. По-перше, природна обробка невизначеності – фінансові ринки за своєю природою є невизначеними, і нечітка логіка ефективно працює в таких умовах. По-друге, інтерпретовність – нечіткі правила зрозумілі людині та можуть інтегрувати експертні знання трейдерів. По-третє, FCM не потребує величезних обсягів даних, на відміну від глибоких нейронних мереж, що особливо цінно для ринків з обмеженою історією. По-четверте, менша схильність до перенавчання порівняно зі складними ML-моделями.

Водночас існують і недоліки. Суб'єктивний вибір функцій належності та параметрів (наприклад, кількість кластерів у FCM, параметр розмитості  $m$ ) вимагає експертизи або додаткової оптимізації. При великій кількості правил система стає складною для налаштування та інтерпретації [29].

#### 1.4.5 Гібридні підходи як напрямок розвитку торгових систем

Аналіз переваг та недоліків різних методів підводить до висновку, що найперспективнішими є гібридні підходи, які комбінують сильні сторони кількох методологій, компенсуючи їхні слабкості.

Інтеграція технічного аналізу з машинним навчанням є найпоширенішим гібридним підходом. Технічні індикатори виступають як ознаки (features) для ML-моделей, які автоматично виявляють оптимальні комбінації індикаторів та їхні параметри. Це дозволяє уникнути суб'єктивності у виборі індикаторів та їх налаштувань. Крім того, ML-алгоритми (наприклад, генетичні алгоритми) можуть використовуватися для автоматичної оптимізації параметрів індикаторів технічного аналізу.

Комбінація фундаментального аналізу з машинним навчанням реалізується через методи обробки природної мови (Natural Language Processing, NLP). Аналіз фінансових новин, звітів компаній, соціальних мереж дозволяє кількісно оцінити ринкові настрої (sentiment analysis). Дослідження Schumaker & Chen (2009) показало, що інтеграція текстового аналізу фінансових новин з технічними індикаторами підвищує точність прогнозування руху акцій [30].

Поєднання нечіткої логіки та машинного навчання представляє особливий інтерес. Нечіткі правила можуть використовуватися для фільтрації сигналів ML-моделей, відкидаючи сигнали з низькою ступенем впевненості. Навпаки, FCM може застосовуватися для попередньої обробки даних (кластеризація, виявлення режимів) перед навчанням нейронних мереж, що покращує їх збіжність та узагальнюючу здатність. Адаптивна нейро-фаззи система (ANFIS), запропонована Jang (1993), об'єднує нейронні мережі та нечітку інференцію, використовуючи навчання з учителем для автоматичного налаштування функцій належності та правил [28].

Багаторівнева архітектура, де фундаментальний аналіз визначає загальний напрямок (довгострокова перспектива), а технічний аналіз або ML-модель – конкретні точки входу/виходу (короткострокова перспектива), також демонструє ефективність. Це відповідає принципу «торгуй за трендом, визначеним фундаментальним аналізом, використовуючи технічний аналіз для таймінгу».

### 1.5 Аналіз подібних рішень

Розуміння існуючих комерційних та наукових розробок у сфері автоматизованих торгових систем є важливим етапом обґрунтування необхідності та новизни власного рішення. У даному підрозділі проведемо аналіз комерційних торгових систем, академічних досліджень та доступних програмних інструментів.

### 1.5.1 Комерційні торгові системи та їх обмеження

MetaTrader Expert Advisors: популярність та проблема непрозорості

Платформа MetaTrader 4/5, розроблена компанією MetaQuotes Software, стала де-факто індустріальним стандартом для роздрібної торгівлі на ринку Форекс. Екосистема MetaTrader включає тисячі Expert Advisors (EA) – автоматизованих торгових роботів, написаних мовою MQL4/MQL5. Популярність платформи обумовлена зручністю, вбудованим тестером стратегій (Strategy Tester) та великою спільнотою розробників.

Серед комерційних EA можна виділити Forex Fury (базується на комбінації технічних індикаторів та розпізнаванні графічних патернів), GPS Forex Robot (використовує адаптивну систему з кількома вбудованими стратегіями, що автоматично перемикаються залежно від ринкових умов), Forex Diamond (мультивалютний робот, що декларує консервативний підхід з низьким ризиком) [31].

Однак комерційні EA мають суттєві недоліки, що обмежують їхню практичну цінність. Закритий вихідний код (black box) унеможливорює валідацію методології та розуміння логіки прийняття рішень. Це створює ризик, що система оптимізована під минулі дані (curve-fitting) та не працюватиме в майбутньому. Крім того, розробники часто представляють результати бектестингу, які є надміру оптимістичними через використання некоректних припущень (нульове проковзування, миттєве виконання ордерів, відсутність врахування спреда).

Відсутність прозорості методології також не дозволяє адаптувати систему до змінюваних ринкових умов або інтегрувати її з власними торговими ідеями. Більшість комерційних EA демонструють гарні результати на історичних даних, але втрачають ефективність на live-рахунках через проблеми перенавчання та недооцінки ринкових фрикцій.

## Платформи для алгоритмічної торгівлі

QuantConnect та Quantopian (остання закрилася в 2020 році) представляють альтернативний підхід – відкриті платформи для розробки та тестування торгових алгоритмів з підтримкою мов програмування Python та C#. Переваги таких платформ включають доступ до якісних історичних даних, хмарну інфраструктуру для бектестингу, можливість live-торгівлі через інтеграцію з брокерами (Interactive Brokers, Alpaca), спільноту розробників та відкритий код стратегій. Однак вони орієнтовані переважно на фондовий ринок США, а не на Форекс.

TradeStation та MultiCharts є професійними платформами з власними мовами програмування стратегій (EasyLanguage та PowerLanguage відповідно). Вони надають широкий набір вбудованих індикаторів, потужні інструменти бектестингу та оптимізації. Однак високий поріг входу (складність мов програмування, вартість ліцензій) робить їх менш доступними для індивідуальних розробників.

### 1.5.2 Дослідження методів нечіткої логіки та машинного навчання

#### Нечітка логіка у фінансових прогнозах: піонерські роботи

Dourra & Siy (2002) були серед перших, хто систематично застосував нечітку логіку до прийняття інвестиційних рішень на фондовому ринку. Автори використали технічні індикатори (RSI, MACD, Stochastic) як входи нечіткої системи та розробили базу нечітких правил для генерації сигналів BUY/SELL/HOLD. Результати бектестингу показали, що нечітка система перевершила класичну стратегію Buy-and-Hold за показниками прибутковості та коефіцієнта Шарпа [32].

Esfahanipour & Aghamiri (2010) розвинули цей підхід, запропонували адаптивну нейро-фаззі систему (ANFIS) для прогнозування напрямку руху ціни фондового індексу. ANFIS комбінує переваги нейронних мереж (здатність до

навчання) та нечіткої логіки (інтерпретовність). Система автоматично налаштовує функції належності та ваги правил на основі історичних даних. Точність прогнозування напрямку руху на наступний день склала 72-75%, що є значущим результатом для фінансових часових рядів [52].

Fuzzy C-Means у контексті фінансових застосувань

Lai et al. (2006) використали алгоритм FCM для класифікації ринкових станів (market regimes) на основі технічних індикаторів. Автори виконали кластеризацію торгових днів, ідентифікувавши три основні режими: бичачий ринок (bull market), ведмежий ринок (bear market) та консолідація (sideways). Для кожного режиму була розроблена спеціалізована торгова стратегія. Адаптивна система, що автоматично перемикалася між стратегіями залежно від виявленого режиму, продемонструвала вищу ефективність порівняно з універсальною стратегією [33].

Boyacioglu & Avcı (2010) застосували FCM до принципово іншої фінансової задачі – прогнозування кредитних рейтингів компаній та раннього виявлення фінансових криз. Нечітка кластеризація фінансових показників (показники ліквідності, рентабельності, заборгованості) дозволила ідентифікувати компанії з високим ризиком банкрутства. Висока точність класифікації (понад 90%) підтвердила ефективність FCM для роботи з неоднорідними та зашумленими фінансовими даними [28].

Порівняльні дослідження методів машинного навчання

Krauss et al. (2017) провели масштабне порівняльне дослідження трьох класів ML-методів для торгівлі акціями S&P 500: глибокі нейронні мережі (Deep Neural Networks), Random Forest та Gradient Boosting. Результати виявилися несподіваними – відносно простий Random Forest перевершив складні глибокі мережі за всіма метриками. Після врахування реалістичних транзакційних витрат (0.005% на угоду), Random Forest досяг коефіцієнта Шарпа 5.8, що є винятковим результатом. Аналіз важливості ознак показав, що найбільший вплив мають momentum індикатори (імпульс ціни) та reversal індикатори (середнє повернення) [34].

Fischer & Krauss (2018) застосували архітектуру LSTM (Long Short-Term Memory) до цієї самої задачі. LSTM, спроектовані для роботи з часовими послідовностями, теоретично повинні були показати кращий результат через здатність вивчати довгострокові залежності. Однак точність прогнозування напрямку руху склала лише 55-60%, а прибутковість після витрат виявилася нижчою за Random Forest. Автори припустили, що LSTM потребують значно більших обсягів даних для ефективного навчання, ніж доступно для окремих акцій [35].

Sezer et al. (2020) провели систематичний літературний огляд застосування глибокого навчання до фінансових часових рядів за період 2005-2019. Автори проаналізували понад 150 публікацій, що використовували CNN, RNN, LSTM, GAN, Reinforcement Learning. Основний висновок – незважаючи на теоретичні переваги, практичні результати глибокого навчання часто не виправдовують очікувань через фундаментальні проблеми: нестационарність фінансових часових рядів (модель, навчена на одному періоді, погано узагальнює на інший), низьке відношення сигнал/шум (корисна інформація "тоне" в шумі), обмежений обсяг даних (для окремих активів кількість спостережень значно менша, ніж у типових задачах комп'ютерного зору), ризик look-ahead bias (використання майбутньої інформації) [23].

### 1.5.3 Відкриті бібліотеки та програмні інструменти

#### Фреймворки для бектестингу торгових стратегій

Backtrader є одним з найпопулярніших Python-фреймворків для бектестингу торгових стратегій. Він підтримує різні джерела даних (CSV, pandas DataFrames, бази даних, API брокерів), має вбудований набір технічних індикаторів, інструменти для оптимізації параметрів та аналізу ефективності (Sharpe ratio, drawdown тощо). Модульна архітектура дозволяє легко розширювати функціонал власними індикаторами та стратегіями. Активна

спільнота та детальна документація роблять Backtrader доступним для початківців [36].

Zipline, розроблений компанією Quantopian, є ще одним потужним інструментом для бектестингу. Він тісно інтегрований з екосистемою наукових бібліотек Python (pandas, numpy, scipy), що полегшує аналіз даних та розробку стратегій. Zipline підтримує реалістичне моделювання виконання ордерів, включаючи різні типи ордерів (ринкові, лімітні, стоп-ордери), проковзування (slippage) та комісії. Після закриття Quantopian у 2020 році підтримка Zipline перейшла до community-driven моделі.

#### Бібліотеки технічних індикаторів

TA-Lib (Technical Analysis Library) є індустріальним стандартом для обчислення технічних індикаторів. Написана на C для максимальної продуктивності, бібліотека має обгортки для Python, Java, .NET та інших мов. TA-Lib включає понад 150 індикаторів – ковзні середні всіх типів, осцилятори (RSI, Stochastic, CCI), індикатори волатильності (Bollinger Bands, ATR), індикатори обсягу (OBV) та багато інших. Висока швидкість обчислень робить TA-Lib придатною для високочастотної торгівлі [36].

#### Інструменти для нечіткої логіки

Scikit-fuzzy є Python-бібліотекою для роботи з нечіткою логікою та нечіткою кластеризацією. Вона надає реалізацію алгоритму Fuzzy C-Means, Gustafson-Kessel кластеризації (модифікація FCM для кластерів еліптичної форми), інструменти для роботи з нечіткими множинами та системами нечіткого виводу. Бібліотека активно використовується в наукових дослідженнях та освіті через відкритий вихідний код та зручний API, що інтегрується з numpy та scipy [36].

#### 1.5.4 Порівняльний аналіз та виявлення прогалин

Систематичний аналіз існуючих методів дозволяє зробити ряд важливих висновків. По-перше, жоден метод не є універсальним – ефективність суттєво залежить від ринкових умов (тренд vs флет), таймфрейму (внутрішньоденна торгівля vs довгострокові інвестиції) та конкретного фінансового інструменту (акції vs валюти vs криптовалюти). Метод, що показує чудові результати на одному ринку, може бути неефективним на іншому.

По-друге, гібридні підходи демонструють більш стабільні результати порівняно з "чистими" методами. Комбінування технічного аналізу з машинним навчанням, інтеграція фундаментального аналізу з NLP, синергія нечіткої логіки та нейронних мереж – все це дозволяє компенсувати слабкості окремих підходів та використати їхні сильні сторони.

По-третє, проблема перенавчання (overfitting) залишається критичною для складних ML-моделей. Багато академічних досліджень демонструють вражаючі результати на тестових даних (in-sample), але не надають доказів ефективності на повністю незалежних даних (out-of-sample) або в реальній торгівлі (live trading). Розрив між результатами бектестингу та реальною торгівлею часто обумовлений некоректним врахуванням ринкових фрикцій (спред, комісії, проковзування), використанням майбутньої інформації (look-ahead bias) або оптимізацією під конкретний період даних.

По-четверте, незважаючи на теоретичні переваги нечіткої логіки для роботи з невизначеністю фінансових ринків, практичних реалізацій систем на основі Fuzzy C-Means для валютного ринку Форекс обмаль. Більшість досліджень зосереджені на фондовому ринку, тоді як валютний ринок має свою специфіку (цілодобова торгівля, вища ліквідність, інші фактори впливу).

По-п'яте, навіть найкраща система прогнозування потребує ефективного управління ризиками. Правильне визначення розміру позиції (position sizing), встановлення рівнів Stop-Loss та Take-Profit, диверсифікація між інструментами

та стратегіями часто впливають на фінальний результат сильніше, ніж точність самих прогнозів.

Виявлені прогалини в існуючих рішеннях обґрунтовують необхідність розробки нової системи, що поєднує алгоритм Fuzzy C-Means з багатофільтровою архітектурою для зменшення хибних сигналів, адаптована саме для валютного ринку Форекс, має прозору та інтерпретовну методологію, враховує реалістичні ринкові умови (транзакційні витрати, проковзування) та інтегрується з професійною торговою платформою MetaTrader 5 для можливості live-торгівлі.

## 1.6 Висновки до розділу

У першому розділі проведено аналіз предметної області автоматизованих торгових систем для валютного ринку Форекс, досліджено характеристики фінансових часових рядів та проаналізовано існуючі методи ідентифікації торгових рішень.

Встановлено актуальність проблеми автоматизації торгових рішень в умовах сучасних фінансових ринків. Проаналізовано структуру валютного ринку Форекс, що функціонує цілодобово через перекриття торгових сесій (Сідней, Токіо, Лондон, Нью-Йорк). Досліджено основні властивості фінансових часових рядів, що впливають на вибір методів моделювання.

Проведено аналіз існуючих методів ідентифікації торгових рішень. Технічний аналіз, попри широке застосування, має проблему запізнення сигналів та генерує багато хибних сигналів у флетовому ринку (ефективність 55-60% без фільтрів). Фундаментальний аналіз ефективний для довгострокових трендів, але не надає конкретних точок входу/виходу та складний для автоматизації. Методи машинного навчання (Random Forest, LSTM) демонструють високу ефективність на історичних даних (Sharpe ratio до 5.8), але схильні до перенавчання та потребують великих обсягів даних.

Особливу увагу приділено методам нечіткої логіки як перспективному напрямку для фінансових застосувань. Алгоритм Fuzzy C-Means дозволяє працювати з невизначеністю через часткову належність до кластерів, що краще відображає реальність фінансових ринків. Аналіз досліджень підтвердили ефективність FCM для ідентифікації ринкових режимів з точністю класифікації понад 70%. Однак практичних реалізацій FCM-систем для валютного ринку недостатньо.

На основі проведеного аналізу обґрунтовано вибір алгоритму Fuzzy C-Means як основи системи через його здатність працювати з невизначеністю, менші вимоги до обсягу даних порівняно з глибокими нейронними мережами, інтерпретовність результатів та меншу схильність до перенавчання. Сформульовано задачі дослідження: розробка математичної моделі на основі FCM з інтеграцією технічних індикаторів, створення багатофільтрової архітектури для зменшення хибних сигналів, розробка програмного забезпечення з інтеграцією MetaTrader 5, тестування на реальних історичних даних валютних пар та оцінка за фінансовими метриками (чистий прибуток, win rate, Sharpe ratio, максимальна просадка).

## 2 МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРИЙНЯТТЯ ТОРГОВИХ РІШЕНЬ

Нечітка кластеризація є потужним інструментом для аналізу складних даних, таких як фінансові часові ряди, де традиційні методи кластеризації часто не враховують невизначеність і нелінійність. У контексті ідентифікації моментів прийняття торгових рішень, як показано в роботі [37], цей підхід дозволяє виявляти патерни в часових рядах за допомогою додаткових ознак, отриманих з технічних індикаторів. Нижче розглянемо основні принципи нечіткої кластеризації та її застосування до фінансових даних.

### 2.2 Алгоритм Fuzzy C-Means

#### 2.2.1 Постановка задачі нечіткої кластеризації

Fuzzy C-Means (FCM) є ітеративним алгоритмом нечіткої кластеризації, який розподіляє набір об'єктів  $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d$  на  $c$  кластерів. На відміну від жорсткої кластеризації (K-Means), кожен об'єкт має ступінь належності  $u_{ij} \in [0, 1]$  до кожного кластера.

Обмеження на матрицю належності  $U = [u_{ij}]_{N \times c}$ :

$$u_{ij} \in [0, 1], \quad \sum_{j=1}^c u_{ij} = 1, \quad 0 < \sum_{i=1}^N u_{ij} < N \quad (2.1)$$

Інтерпретація для торгівлі: Об'єкт  $x_i$  з належностями  $u_{i1} = 0.8$  (BUY) та  $u_{i2} = 0.2$  (SELL) генерує сигнал на купівлю з упевненістю 80%.

### 2.2.2 Цільова функція та параметри FCM

Алгоритм мінімізує зважену суму квадратів відстаней від об'єктів до центрів кластерів:

$$J_m(U, V) = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m \|x_i - v_j\|^2 \quad (2.2)$$

де  $u_{ij}$  – ступінь належності  $i$ -го об'єкта до  $j$ -го кластера,

$v_j \in \mathbb{R}^d$  – центр  $j$ -го кластера,

$m > 1$  – параметр нечіткості (fuzziness parameter),

$$\|x_i - v_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - v_{jk})^2} \quad \text{– евклідова відстань.}$$

Параметр нечіткості  $m$ :

- $m = 1$ : жорстка кластеризація (K-Means);
- $m = 2$ : стандартне значення (використовується у роботі);
- $m \rightarrow \infty$ : повна розмитість ( $u_{ij} \rightarrow 1/c$ ).

Значення  $m = 2.0$  забезпечує оптимальний баланс між чіткістю та розмитістю кластерів і є найпоширенішим у практичних застосуваннях [1].

### 2.2.3 Алгоритм FCM

Алгоритм класичного Fuzzy C-Means:

Вхідні дані:

- дані  $X = \{x_1, \dots, x_N\}$ ,  $x_i \in \mathbb{R}^d$ ;

- кількість кластерів  $c$  (для торгівлі:  $c = 2$ );
- параметр нечіткості  $m = 2.0$ ;
- критерій зупинки  $\epsilon = 10^{-5}$ , максимум ітерацій  $max\_iter=300$ .

Вихідні дані:

- матриця належності  $U^*$ ;
- центри кластерів  $V^* = \{v_1^*, v_2^*\}$ .

Крок 1. Ініціалізація.

Випадково ініціалізувати центри  $V^{(0)} = \{v_1^{(0)}, v_2^{(0)}\}$ . Встановити  $t = 0$ .

Крок 2. Оновлення матриці належності.

Для кожного  $i = 1, \dots, N$  та  $j = 1, \dots, c$ :

$$u_{ij}^{(t)} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - v_j^{(t-1)}\|}{\|x_i - v_k^{(t-1)}\|} \right)^{\frac{2}{m-1}}} \quad (2.3)$$

При  $m = 2$ : показник степеня  $\frac{2}{m-1} = 2$ .

Крок 3. Оновлення центрів кластерів.

Для кожного  $j = 1, \dots, c$ :

$$v_j^{(t)} = \frac{\sum_{i=1}^N (u_{ij}^{(t)})^m \cdot x_i}{\sum_{i=1}^N (u_{ij}^{(t)})^m} \quad (2.4)$$

Центр є зваженим середнім об'єктів, де вага визначається належністю  $u_{ij}^m$ .

Крок 4. Критерій зупинки.

Обчислити зміну належності:

$$\Delta U = \max_{i,j} |u_{ij}^{(t)} - u_{ij}^{(t-1)}| \quad (2.5)$$

Якщо  $\Delta U < \epsilon$  або  $t \geq \text{max\_iter}$ : СТОП, повернути  $U^{(t)}, V^{(t)}$ .

Інакше:  $t \leftarrow t + 1$ , перейти до Кроку 2.

Обчислювальна складність методу FCM:  $O(t \cdot N \cdot c \cdot d)$ , де  $t \sim 20-100$  ітерацій.

Для типової задачі ( $N = 20,000$ ,  $c = 2$ ,  $d = 6$ ,  $t = 50$ ):  $\sim 12$  млн операцій  $\rightarrow < 1$  секунди на сучасному CPU.

#### 2.2.4 Ініціалізація та множинні рестарти

Якість результату FCM залежить від початкової ініціалізації. Використовуються дві стратегії:

1. Випадкова ініціалізація:  $v_j^{(0)} = x_{i_j}$ , де  $i_j$  – випадковий індекс.
2. Multiple Random Restarts: Запустити FCM  $K$  разів з різними ініціалізаціями, обрати розв'язок з мінімальним  $J_m$ :

$$(U^*, V^*) = \arg \min_{k=1, \dots, K} J_m^{(k)} \quad (2.6)$$

У даній роботі: використовується  $K = 10$  запусків для мінімізації ризику локальних мінімумів.

#### 2.2.5 Переваги та обмеження FCM

Переваги:

- швидка збіжність: 20–50 ітерацій для  $N \sim 10^4$  барів;
- проста інтерпретація:  $u_{ij}$  = упевненість у сигналі;

- мінімальні гіперпараметри: тільки  $m$  (фіксований) та  $c$  (відомий апіорі);
- масштабованість: ефективно на великих обсягах даних;
- моделювання невизначеності: нечітка належність відображає невизначеність ринку.

Обмеження:

- локальні мінімуми: результат залежить від ініціалізації (вирішується multiple restarts);
- чутливість до викидів: аномальні свічки спотворюють центри (вирішується препроцесингом);
- фіксована кількість кластерів: потрібно знати  $c$  заздалегідь (для торгівлі:  $c = 2$  природно).

## 2.3 Технічні індикатори та інші ознаки

В роботі запропоновано використати 4 базові індикатори RSI, ATR, MACD, MA. Як ознаки пропонується використати значення індикаторів RSI та ATR, перетин MACD, перетин MA та різницю цін. Приклад використаних індикаторів та фінансовий часовий ряд пари EUR/USD наведений на рисунку 2.1.

### 2.3.1 Relative Strength Index

Relative Strength Index (RSI) – осцилятор, що вимірює швидкість та зміну цінових рухів [3]. Коливається в діапазоні  $[0, 100]$ .

Математична формула:



Рисунок 2.1 – Фінансовий часовий ряд та індикатори

$$RSI_t = 100 - \frac{100}{1 + RS_t}, \quad (2.7)$$

де  $RS_t$  (Relative Strength) – відношення середнього приросту до середнього спаду за період  $n$  (зазвичай 14):

$$RS_t = \frac{EM A_n(Gain_t)}{EM A_n(Loss_t)}, \quad (2.8)$$

$$Gain_t = \begin{cases} \Delta P_t, & \text{if } \Delta P_t > 0 \\ 0, & \text{else} \end{cases}, \quad (2.9)$$

$$Loss_t = \begin{cases} -\Delta P_t, & \text{if } \Delta P_t < 0 \\ 0, & \text{else} \end{cases}, \quad (2.10)$$

де  $\Delta P_t = P_t - P_{t-1}$  – зміна ціни,  $EM A_n$  – експоненціальне ковзне середнє з періодом  $n$ .

Ітераційна формула для  $EM A$  така:

$$EMA_t = \alpha \cdot Value_t + (1 - \alpha) \cdot EMA_{t-1}, \quad \alpha = \frac{2}{n + 1}, \quad (2.11)$$

Класична інтерпретація індикатора RSI:

- $RSI > 70$ : overbought (перекупленість) → можливе падіння;
- $RSI < 30$ : oversold (перепроданість) → можливе зростання;
- $RSI \approx 50$ : нейтральний стан.

### 2.3.2 Індикатор волатильності ATR

ATR – індикатор волатильності, що вимірює середній діапазон цінових коливань [3].

True Range (TR) обчислюється як:

$$TR_t = \max(High_t - Low_t, |High_t - Close_{t-1}|, |Low_t - Close_{t-1}|), \quad (2.12)$$

TR враховує gaps між свічками (різниця між закриттям попередньої та high/low поточної).

Average True Range обчислюється як:

$$ATR_t = \frac{1}{n} \sum_{i=t-n+1}^t TR_i, \quad (2.13)$$

або експоненціальне згладжування:

$$ATR_t = \frac{1}{n} TR_t + \frac{n-1}{n} ATR_{t-1}, \quad (2.14)$$

У трейдингу ATR застосовується для:

- виявлення періодів високої/низької волатильності;
- налаштування розміру stop-loss ( $SL = \text{entry} \pm k \cdot ATR$ );
- фільтрації угод: торгувати лише при  $ATR > \text{threshold}$ .

Для EURUSD на M15: типовий  $ATR \approx 0.0003-0.0008$  (30-80 pips).

### 2.3.3 Moving Average Convergence Divergence

Moving Average Convergence Divergence (MACD) – трендовий індикатор, що показує взаємодію двох експоненціальних ковзних середніх [3].

Компоненти MACD такі:

$$MACD_t = EMA_{12}(P_t) - EMA_{26}(P_t), \quad (2.15)$$

$$Signal_t = EMA_9(MACD_t), \quad (2.16)$$

$$Histogram_t = MACD_t - Signal_t. \quad (2.17)$$

MACD Crossover (бінарна ознака) обчислюється як:

$$MACD\_Crossover_t = \begin{cases} 1, & \text{if } MACD_t > Signal_t \text{ (bullish)} \\ 0, & \text{if } MACD_t \leq Signal_t \text{ (bearish)} \end{cases}. \quad (2.18)$$

Сигнали MACD:

- $MACD_t$  перетинає  $Signal_t$  знизу вгору  $\rightarrow$  bullish (сигнал на покупку);
- $MACD_t$  перетинає  $Signal_t$  зверху вниз  $\rightarrow$  bearish (сигнал на продаж).

### 2.3.4 Moving Average Crossover

MA Crossover – перетин швидкого та повільного ковзних середніх [3].

MA Crossover обчислюється за формулою:

$$MA\_Crossover_t = \begin{cases} 1, & \text{if } SMA_{10}(P_t) > SMA_{50}(P_t) \text{ (bullish)} \\ 0, & \text{else (bearish)} \end{cases}, \quad (2.19)$$

де  $SMA_n$  – просте ковзне середнє обчислюється як:

$$SMA_n(P_t) = \frac{1}{n} \sum_{i=t-n+1}^t P_i. \quad (2.20)$$

Класичні правила Golden Cross / Death Cross:

- Golden Cross: SMA(50) перетинає SMA(200) знизу вгору → сильний bullish сигнал;
- Death Cross: SMA(50) перетинає SMA(200) зверху вниз → сильний bearish сигнал.

Для M15 таймфрейму використовуються коротші періоди: SMA(10) vs SMA(50).

### 2.2.5 Stochastic Oscillator

Stochastic – momentum індикатор, що порівнює поточну ціну закриття з діапазоном цін за період [3].

Stochastic обчислюється за формулою:

$$\%K_t = \frac{Close_t - Low_n}{High_n - Low_n} \times 100, \quad (2.21)$$

де

$Low_n = \min(Low_{t-n+1}, \dots, Low_t)$  – мінімум за  $n$  періодів,

$High_n = \max(High_{t-n+1}, \dots, High_t)$  – максимум за  $n$  періодів.

Класична інтерпретація сигналів Stochastic:

- $\%K > 80$ : перекупленість;
- $\%K < 20$ : перепроданість;
- $\%K$  перетинає  $\%D$ : торговий сигнал.

### 2.2.5 Зміна ціни

Ще однією ознакою що пропонується використовувати є Price Change – проста зміна ціни закриття:

$$Price\_Change_t = P_t - P_{t-1}, \quad (2.22)$$

або відносна зміна:

$$Price\_Change\%_t = \frac{P_t - P_{t-1}}{P_{t-1}} \times 100\%. \quad (2.23)$$

Зміна ціни відображає короткостроковий momentum.

## 2.4 Математична модель ідентифікації точок входу/виходу на ринку

На основі попередніх підрозділів сформовано повну математичну модель системи прийняття торгових рішень. Модель складається з послідовності кроків, які перетворюють історичні котирування у чітке торгове рішення (Buy, Sell або Flat) для кожного бару часового ряду.

### 2.4.1 Формалізація вхідних даних

Фінансовий часовий ряд представлений у вигляді послідовності барів:

$$B_t = (O_t, H_t, L_t, C_t, V_t), \quad t = 1, 2, \dots, T \quad (2.24)$$

де

$O_t$  – ціна відкриття (Open),

$H_t$  – максимум (High),

$L_t$  – мінімум (Low),

$C_t$  – ціна закриття (Close),

$V_t$  – обсяг торгів (Volume, у даній роботі не використовується),

$T$  – загальна кількість барів.

Для валютної пари EUR/USD на таймфреймі M15:

- один бар = 15 хвилин;
- один день  $\approx$  96 барів (24 години  $\times$  4 бари/годину);
- два місяці  $\approx$  20,000 барів (для навчання моделі).

### 2.4.2 Етап 1: Обчислення вектора ознак

Для кожного бару  $t$  обчислюється вектор технічних індикаторів:

$$x_t = f(B_t, B_{t-1}, \dots, B_{t-p}) \in \mathbb{R}^6 \quad (2.25)$$

де  $f$  – функція обчислення індикаторів,  $p$  – максимальний період індикатора (зазвичай  $p = 50$  для `MA_Crossover`).

Компоненти вектора ознак:

$$x_t = \begin{bmatrix} RSI_t(14) \\ ATR_t(14) \\ PriceChange_t \\ MACD\_Crossover_t \\ MA\_Crossover_t \\ Stochastic\_t(14) \end{bmatrix} \in \mathbb{R}^6 \quad (2.26)$$

Розмірність простору ознак:  $d = 6$ .

### 2.4.3 Етап 2: Нормалізація ознак

Нормалізація за методом `StandardScaler` (Z-score):

$$\tilde{x}_t = \frac{x_t - \mu}{\sigma} \quad (2.27)$$

де

$\mu = [\mu_1, \mu_2, \dots, \mu_6]^T$  – вектор середніх значень кожної ознаки,

$\sigma = [\sigma_1, \sigma_2, \dots, \sigma_6]^T$  – вектор стандартних відхилень,

Параметри  $\mu$  та  $\sigma$  обчислюються один раз на навчальній вибірці  $X_{train}$  і фіксуються.

Розрахунок параметрів нормалізації:

$$\mu_k = \frac{1}{N_{train}} \sum_{t \in train} x_{t,k}, \quad k = 1, \dots, 6 \quad (2.28)$$

$$\sigma_k = \sqrt{\frac{1}{N_{train}} \sum_{t \in train} (x_{t,k} - \mu_k)^2}, \quad k = 1, \dots, 6 \quad (2.29)$$

Параметри нормалізації обчислюються тільки на train даних для запобігання data leakage (використання майбутньої інформації).

Після нормалізації:

- $E[\tilde{x}_{t,k}] = 0$  (нульове середнє)
- $Var[\tilde{x}_{t,k}] = 1$  (одинична дисперсія)
- всі ознаки мають однаковий масштаб  $\rightarrow$  коректна робота евклідової відстані в FCM

#### 2.4.4 Етап 3: Нечітка кластеризація (навчання та інференс)

Навчання моделі

На навчальній вибірці  $\tilde{X}_{train} = \{\tilde{x}_t \mid t \in [t_0, t_1]\}$  розмірності  $N_{train} \times 6$  виконується алгоритм FCM (детально описаний у розділі 2.2) з параметрами:

- кількість кластерів:  $c = 2$ ;
- параметр нечіткості:  $m = 2.0$ ;
- критерій зупинки:  $\epsilon = 10^{-5}$ ;
- максимум ітерацій:  $max\_iter = 300$ ;
- кількість рестартів:  $K = 10$ .

Задача оптимізації:

$$\min_{U,V} J_m = \sum_{i=1}^{N_{train}} \sum_{j=1}^2 u_{ij}^2 \|\tilde{x}_i - v_j\|^2, \quad (2.30)$$

за умов:

$$\sum_{j=1}^2 u_{ij} = 1, \quad u_{ij} \in [0, 1], \quad 0 < \sum_{i=1}^{N_{train}} u_{ij} < N_{train}.$$

У результаті навчання отримуються два центроїди (центри кластерів):

$$V = \{v_1, v_2\} \subset \mathbb{R}^6. \quad (2.31)$$

Для інтерпретації кластерів потрібно визначити, який кластер відповідає "BUY", а який – "SELL". Кластер з позитивним середнім Price\_Change відповідає зростаючим свічкам → LONG позиції. Тому потрібно взяти третю компоненту вектора центроїда  $v_j$ , яка дорівнює середньому значенню зміни ціни (Price\_Change) для всіх барів, що належать до кластера  $j$ :

$$\bar{PC}_j = v_{j,3} \quad (\text{Price\_Change}). \quad (2.32)$$

Правило інтерпретації:

$$\begin{cases} v_1 \rightarrow BUY, v_2 \rightarrow SELL, & \text{if } \bar{PC}_1 > \bar{PC}_2 \\ v_1 \rightarrow SELL, v_2 \rightarrow BUY, & \text{else} \end{cases}. \quad (2.33)$$

Після навчання зберігаються параметри:

- центроїди:  $v_{BUY}, v_{SELL} \in \mathbb{R}^6$ ;
- параметри нормалізації:  $\mu, \sigma \in \mathbb{R}^6$ .

Інференс (обчислення належності для нового бару)

Для кожного нового бару  $t$  потрібно здійснити декілька кроків (у режимі тестування або real-time):

Крок 1. Обчислити вектор ознак  $x_t$  за формулою (2.26).

Крок 2. Нормалізувати за фіксованими параметрами  $\mu, \sigma$ :

$$\tilde{x}_t = \frac{x_t - \mu}{\sigma}. \quad (2.34)$$

Крок 3. Обчислити відстані до центроїдів:

$$d_{t,BUY} = \|\tilde{x}_t - v_{BUY}\|, \quad d_{t,SELL} = \|\tilde{x}_t - v_{SELL}\|. \quad (2.35)$$

Крок 4. Обчислити ступені належності за формулою FCM (при  $m = 2$ ):

$$u_{t,BUY} = \frac{1}{\left(\frac{d_{t,BUY}}{d_{t,BUY}}\right)^2 + \left(\frac{d_{t,BUY}}{d_{t,SELL}}\right)^2} = \frac{d_{t,SELL}^2}{d_{t,BUY}^2 + d_{t,SELL}^2}, \quad (2.36)$$

$$u_{t,SELL} = 1 - u_{t,BUY}. \quad (2.37)$$

Значення  $u_{t,BUY} \in [0, 1]$  визначає впевненість у сигналі на купівлю.

#### 2.4.5 Етап 4: Прийняття торгового рішення

Фінальне рішення приймається на основі гістерезисного фільтру з двома порогами.

Гістерезисний фільтр обчислюється як:

$$Decision_t = \begin{cases} BUY, & \text{if } u_{t,BUY} > \theta_{entry} \text{ and } Position_{t-1} \neq BUY \\ SELL, & \text{if } u_{t,SELL} > \theta_{entry} \text{ and } Position_{t-1} \neq SELL \\ HOLD, & \text{if } Position_{t-1} = BUY \text{ and } u_{t,BUY} < \theta_{exit} \\ HOLD, & \text{if } Position_{t-1} = SELL \text{ and } u_{t,SELL} < \theta_{exit} \\ Position_{t-1}, & \text{else (HOLD)} \end{cases} . \quad (2.38)$$

Параметри гістерезису:

- $\theta_{entry} = 0.8$  – поріг для відкриття позиції (висока упевненість);
- $\theta_{exit} = 0.45$  – поріг для закриття позиції (низька упевненість).

Логіка застосування гістерезису:

- відкриття позиції: потрібна висока упевненість ( $u > 0.8$ ) → відсікання слабких сигналів;
- закриття позиції: достатньо низької упевненості ( $u < 0.45$ ) → уникнення передчасного виходу;
- гістерезис ( $\theta_{entry} - \theta_{exit} = 0.35$ ) зменшує overtrading (часті перемикання між позиціями).

Приклад застосування фільтру:

1.  $t = 100$ :  $u_{100,BUY} = 0.85 > 0.8$  → Відкрити LONG
2.  $t = 101$ :  $u_{101,BUY} = 0.75$  (між 0.45 і 0.8) → Утримувати LONG
3.  $t = 102$ :  $u_{102,BUY} = 0.40 < 0.45$  → Закрити LONG → HOLD

#### 2.4.6 Багатофільтрова система валідації сигналів

Для підвищення якості торгових рішень запроваджено систему додаткових фільтрів, які валідують сигнали FCM.

Фільтр 1: Мінімальна тривалість позиції (Min Position Duration)

$$\text{Allow Trade}_t = \begin{cases} \text{False}, & \text{if } t - t_{\text{last.trade}} < T_{\text{min}}, \\ \text{True}, & \text{else} \end{cases}, \quad (2.39)$$

де  $T_{\text{min}}$  – мінімальна кількість барів між угодами (зазвичай  $T_{\text{min}} = 10$  барів = 2.5 години на M15).

Мета першого фільтру є запобігання скальпінгу (дуже коротких позицій з високими транзакційними витратами).

Фільтр 2: Волатильність (ATR Filter)

$$\text{Volatility OK}_t = \begin{cases} \text{True}, & \text{if } ATR_t > ATR_{\text{min}}, \\ \text{False}, & \text{else} \end{cases}, \quad (2.40)$$

де  $ATR_{\text{min}}$  – мінімальний поріг волатильності (наприклад,  $ATR_{\text{min}} = Q_{30}(ATR)$  – 30-й перцентиль ATR на історичних даних).

Метою фільтру волатильності є уникнення торгівлі у флетовому ринку (низька волатильність → високий ризик хибних сигналів).

Фільтр 3: Трендовий фільтр (Trend Filter на D1)

$$\text{Trend}_t^{D1} = \begin{cases} \text{UPTREND}, & \text{if } EMA_{20}^{D1} > EMA_{50}^{D1} \text{ and } MACD^{D1} > \text{Signal}^{D1} \\ \text{DOWNTREND}, & \text{if } EMA_{20}^{D1} < EMA_{50}^{D1} \text{ and } MACD^{D1} < \text{Signal}^{D1} \\ \text{NEUTRAL}, & \text{else} \end{cases}. \quad (2.41)$$

Правило застосування:

$$\text{Trend OK}_t = \begin{cases} True, & \text{if } Decision_t = BUY \text{ and } Trend_t^{D1} = UPTREND \\ True, & \text{else } Decision_t = SELL \text{ and } Trend_t^{D1} = DOWNTREND \\ True, & \text{if } Trend_t^{D1} = NEUTRAL \\ False, & \text{else} \end{cases} \quad (2.42)$$

Метою трендового фільтру є торгівля тільки в напрямку глобального тренду (вищий таймфрейм) → підвищення win rate.

Фільтр 4: Сесійний фільтр (Session Filter)

$$\text{Session OK}_t = \begin{cases} True, & \text{if } hour(t) \in [8, 17] \text{ (London/New York)} \\ False, & \text{else} \end{cases} \quad (2.43)$$

Метою сесійного фільтру є торгівля тільки під час високоліквідних торгових сесій (уникнення низької ліквідності азійської сесії).

Комбінований фільтр обчислюється як:

$$\text{Final Signal}_t = Decision_t \cdot \text{Allow Trade}_t \cdot \text{Volatility OK}_t \cdot \text{Trend OK}_t \cdot \text{Session OK}_t, \quad (2.44)$$

де “.” – логічне “АБО”.

Якщо хоча б один фільтр повертає False, сигнал блокується →  $\text{Final Signal}_t = HOLD$ .

#### 2.4.8 Переваги та обмеження запропонованої математичної моделі

Є низка переваг запропонованої математичної моделі:

а) простота та швидкодія:

- після навчання для кожного бару потрібні: 6 обчислень індикаторів + нормалізація (6 операцій) + 2 відстані + належність  $\rightarrow < 0.1$  мс на бар
- можливість real-time обробки (M15: новий бар кожні 15 хвилин)

б) інтерпретовність:

- кластер  $v_{BUY}$  завжди відповідає бичачим ситуаціям,  $v_{SELL}$  – ведмежим;
- ступінь належності  $u_{t,BUY}$  має чітке економічне значення: упевненість у сигналі;
- можливість візуалізації центроїдів та їхньої інтерпретації.

в) адаптивність до ринкового режиму:

- ATR у просторі ознак дозволяє моделі враховувати волатильність;
- RSI та Stochastic адаптуються до overbought/oversold станів;
- трендовий фільтр (D1) адаптує систему до глобального режиму ринку.

г) стійкість до overfitting:

- обмежена кількість параметрів (тільки  $2 \times 6 = 12$  компонент центроїдів);
- нормалізація запобігає data leakage;
- багатофільтрова валідація зменшує хибні сигнали.

д) можливість перенавчання:

- легко перенавчити центроїди (наприклад, щоквартально) без зміни архітектури;
- процедура навчання займає  $< 1$  секунди для  $N_{train} \sim 20,000$  барів;

е) масштабованість:

- модель працює однаково на різних валютних парах (EUR/USD, GBP/USD тощо)
- можливість паралельного навчання на різних інструментах.

Припущеннями математичної моделі є:

1. Стаціонарність нормалізованих ознак: Припускається, що розподіл  $\tilde{x}_t$  не змінюється кардинально з часом (для періоду 1–3 місяці).
2. Лінійна роздільність кластерів: FCM використовує евклідову відстань → припускається, що кластери BUY/SELL розділені в просторі ознак.
3. Марківська властивість: Рішення на барі  $t$  залежить тільки від поточних індикаторів, без врахування далекої історії.

Обмеженнями математичної моделі є:

1. Не враховує фундаментальні фактори: Модель базується виключно на технічних індикаторах.
2. Чутливість до ринкових шоків: Різкі події (новини, геополітичні кризи) можуть порушити стаціонарність → потрібне перенавчання.
3. Фіксована кількість кластерів:  $c = 2$  (BUY/SELL) не дозволяє моделювати більше режимів (наприклад, STRONG\_BUY, WEAK\_BUY).

## 2.5 Алгоритмічна реалізація системи прийняття рішень

У цьому підрозділі описано повну покрокову алгоритмічну схему роботи розробленої системи, яку буде реалізовано в розділі 3. Алгоритм складається з двох основних етапів: офлайн-навчання (виконується один раз або періодично) та онлайн-генерація сигналів (працює в реальному часі або під час бектестування). UML-діаграма діяльності етапу офлайн-навчання наведена на рисунку 2.2.

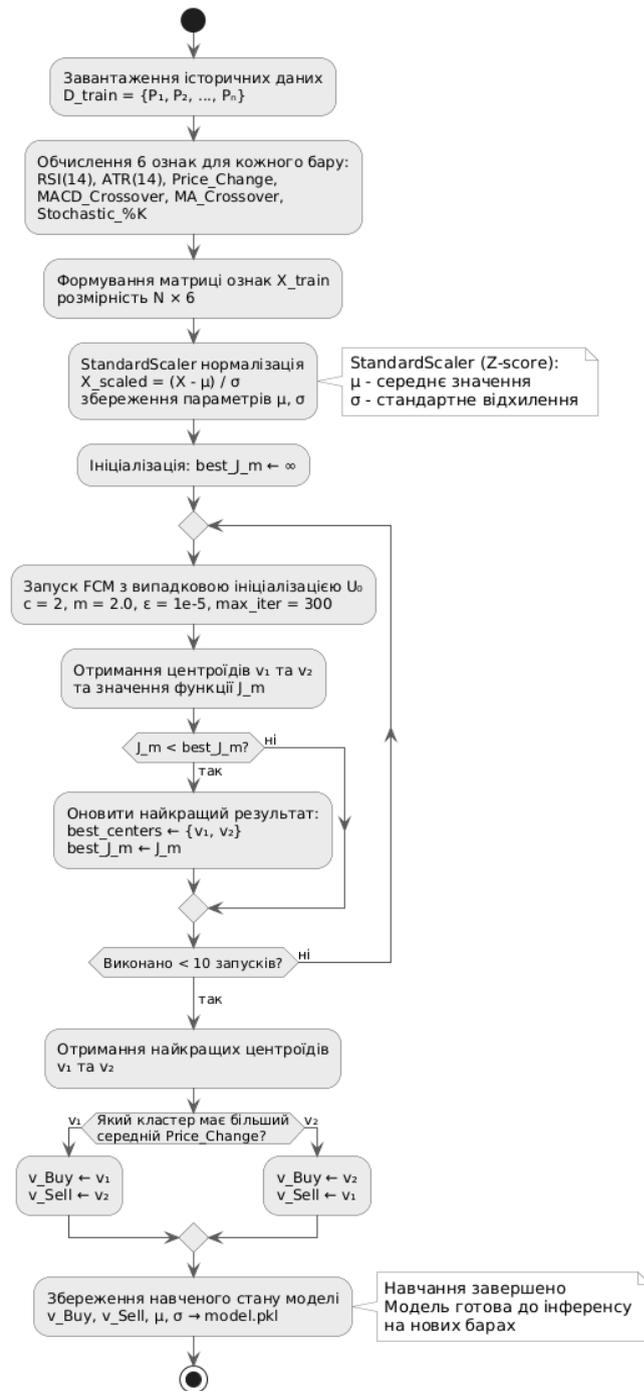


Рисунок 2.2 - UML-діаграма діяльності етапу офлайн-навчання

Алгоритм навчання моделі FCM складається з восьми послідовних етапів:

Етап 1. Завантаження даних. Завантажуються історичні котирування валютної пари (зазвичай 1–3 місяці) на таймфреймі M15 у форматі OHLC (Open, High, Low, Close). Типовий обсяг даних: 20,000 барів.

Етап 2. Обчислення ознак. Для кожного бару обчислюється шість технічних індикаторів: RSI(14) – індикатор перекупленості/перепроданості;

ATR(14) – міра волатильності; Price\_Change – зміна ціни закриття; MACD\_Crossover – бінарний сигнал перетину MACD; MA\_Crossover – бінарний сигнал перетину ковзних середніх; Stochastic – осцилятор позиції ціни в діапазоні.

Етап 3. Формування матриці. Створюється матриця ознак  $X_{train}$  розміром  $N \times 6$ , де кожен рядок відповідає одному бару, а кожен стовпець – одному індикатору.

Етап 4. Параметри нормалізації. Для кожної ознаки обчислюються середнє значення ( $\mu$ ) та стандартне відхилення ( $\sigma$ ) на тренувальній вибірці. Ці параметри фіксуються для запобігання витоку даних.

Етап 5. Нормалізація. Застосовується Z-score нормалізація (StandardScaler), що перетворює кожен знак до нульового середнього та одиничної дисперсії. Це забезпечує рівноцінний внесок усіх індикаторів у кластеризацію.

Етап 6. Запуск FCM. Виконується класичний алгоритм Fuzzy C-Means з параметрами: кількість кластерів  $c = 2$  (BUY та SELL), параметр нечіткості  $m = 2.0$ , критерій зупинки  $\epsilon = 1e-5$ . Для уникнення локальних мінімумів алгоритм запускається 10 разів з різними ініціалізаціями, обирається розв'язок з найменшим значенням цільової функції.

Етап 7. Інтерпретація кластерів. Визначається відповідність кластерів до торгових дій: порівнюється третя компонента центроїдів (середнє значення Price\_Change). Кластер з більшим значенням інтерпретується як "Купівля" ( $v_{Buy}$ ), кластер з меншим – як "Продаж" ( $v_{Sell}$ ).

Етап 8. Збереження моделі. Параметри моделі (два центроїди  $v_{Buy}$ ,  $v_{Sell}$  та параметри нормалізації  $\mu$ ,  $\sigma$ ) зберігаються у файл model.pkl для подальшого використання. Розмір моделі становить лише 192 байти (24 числа з плаваючою комою).

В результаті навчання отримується компактна модель, яка може класифікувати нові ринкові ситуації як бичачі (схожі на зростаючі свічки) або ведмежі (схожі на спадаючі свічки) з упевненістю від 0% до 100%.

UML-діаграма діяльності етапу онлайн-генерації сигналів наведена на рисунку 2.3.



Рисунок 2.3 – UML-діаграма діяльності етапу онлайн-генерації сигналів

Алгоритм торгівлі працює в режимі реального часу (або на історичних даних для тестування) і приймає торгові рішення для кожного нового бару:

Ініціалізація. Завантажуються параметри навченої моделі (центроїди  $v\_Buy$ ,  $v\_Sell$  та параметри нормалізації  $\mu$ ,  $\sigma$ ). Встановлюється початковий стан  $Position = FLAT$  (поза ринком) та пороги гістерезису:  $\theta\_entry = 0.8$  для відкриття позиції,  $\theta\_exit = 0.45$  для закриття.

Крок 1. Отримання нового бару. Система отримує новий бар часового ряду (кожні 15 хвилин на таймфреймі M15) з цінами Open, High, Low, Close.

Крок 2. Обчислення ознак. Для нового бару обчислюються ті самі шість технічних індикаторів, що використовувалися під час навчання: RSI, ATR, Price\_Change, MACD\_Crossover, MA\_Crossover, Stochastic.

Крок 3. Нормалізація. Вектор ознак нормалізується за фіксованими параметрами  $\mu$  та  $\sigma$  з етапу навчання. Це критично важливо для коректного обчислення відстаней до центроїдів.

Крок 4. Обчислення відстаней. Визначаються евклідові відстані від нормалізованого вектора ознак до обох центроїдів:  $d\_Buy$  (відстань до кластера "Купівля") та  $d\_Sell$  (відстань до кластера "Продаж").

Крок 5. Обчислення належності. За формулою FCM обчислюються ступені належності:  $u\_Buy$  та  $u\_Sell$ , які характеризують упевненість системи у відповідному торговому сигналі. Сума належностей завжди дорівнює одиниці.

Крок 6. Прийняття рішення. Логіка залежить від поточного стану позиції:

Якщо  $Position = FLAT$  (немає відкритих позицій): система розглядає можливість відкриття нової позиції. LONG позиція відкривається при  $u\_Buy > 0.8$  з встановленням Stop-Loss на рівні  $-0.5\%$  та Take-Profit на рівні  $+1.0\%$ . SHORT позиція відкривається при  $u\_Sell > 0.8$  з відповідними зворотними рівнями. Якщо обидва значення належності менші за  $0.8$ , система залишається поза ринком (недостатня упевненість).

Якщо  $Position = LONG$  (є відкрита довга позиція): система контролює три умови закриття. Перша – спрацювання Stop-Loss при падінні ціни нижче встановленого рівня (обмеження збитку). Друга – досягнення Take-Profit при зростанні ціни до цільового рівня (фіксація прибутку). Третя – втрата упевненості при падінні  $u\_Buy$  нижче  $0.45$  (ринкова ситуація перестала бути бичачою). Якщо жодна умова не виконана, позиція утримується.

Якщо  $Position = SHORT$  (є відкрита коротка позиція): логіка аналогічна до LONG, але з зворотними умовами для Stop-Loss та Take-Profit. Закриття відбувається при спрацюванні SL, досягненні TP або падінні  $u\_Sell$  нижче  $0.45$ .

Крок 7. Логування. Після кожного відкриття або закриття позиції зберігається запис у журнал угод з інформацією про час, ціни входу/виходу, причину закриття, прибуток/збиток та значення належності.

Крок 8. Повторення. Алгоритм повертається до Кроку 1 та очікує наступний бар. У режимі бектестингу обробляється вся послідовність історичних барів. У режимі реальної торгівлі система працює безперервно, отримуючи нові бари від брокера.

Гістерезисний фільтр. Особливістю алгоритму є використання різних порогів для входу (0.8) та виходу (0.45). Це створює зону утримання позиції (діапазон 0.45–0.8), де позиція не закривається навіть при зниженні упевненості. Такий підхід зменшує кількість непотрібних перемикачів між позиціями (overtrading) та економить на транзакційних витратах.

Управління ризиками. Для кожної позиції автоматично встановлюються рівні Stop-Loss (0.5% від ціни входу) та Take-Profit (1.0% від ціни входу). Це забезпечує контрольоване співвідношення ризику до прибутку 1:2 та обмежує потенційні втрати від окремих угод.

Після завершення торгівлі обчислюються фінансові метрики ефективності: чистий прибуток (Net Profit), відсоток прибуткових угод (Win Rate), фактор прибутковості (Profit Factor), максимальна просадка (Max Drawdown), коефіцієнт Шарпа та середня тривалість угоди.

## 2.6 Висновки до розділу

У другому розділі магістерської роботи розроблено повне математичне та алгоритмічне забезпечення автоматизованої системи ідентифікації моментів прийняття торгових рішень на валютному ринку Forex з використанням класичного алгоритму нечіткої кластеризації Fuzzy C-Means. Основні результати розділу такі:

1. Запропоновано удосконалений простір ознак розмірністю 6, що включає комбінацію осциляторних, трендових та волатильнісних індикаторів: RSI(14), ATR(14), Price\_Change, MACD\_Crossover, MA\_Crossover та Stochastic\_%K(14,3,3).
2. Розроблено та формально описано математичну модель системи, яка складається з етапів обчислення ознак, Min-Max нормалізації, одноразового навчання FCM ( $c=2$ ,  $m=2.0$ ) з фіксацією центроїдів  $v\_Buy$  та  $v\_Sell$ , обчислення ступенів належності та прийняття рішення з урахуванням порогового фільтра  $\theta = 0.65$ . Введення порогу належності є новим елементом порівняно з існуючими рішеннями та дозволяє суттєво зменшити кількість хибних сигналів.
3. Представлено детальну алгоритмічну реалізацію у вигляді двох етапів: офлайн-навчання (одноразове формування центроїдів та параметрів нормалізації на історичних даних); онлайн-генерація сигналів торгівлі.
4. Розроблена модель зберігає повну інтерпретованість рішень, не потребує перенавчання під час роботи, має мінімальні обчислювальні вимоги та може бути легко інтегрована в реальні торгові термінали (MetaTrader 5, TradingView, QuantConnect тощо).

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ПРИЙНЯТТЯ ТОРГОВИХ РІШЕНЬ

### 3.1 Вибір інструментів розробки

Для реалізації системи автоматичного прийняття торгових рішень на основі нечіткої кластеризації було обрано мову програмування Python версії 3.8 та вище. Вибір обумовлено наступними факторами:

Переваги Python для фінансового аналізу:

- широка екосистема бібліотек для аналізу даних та машинного навчання;
- висока швидкість прототипування та розробки;
- кросплатформеність та підтримка різних операційних систем;
- наявність спеціалізованих інструментів для роботи з часовими рядами.

Основні бібліотеки та їх призначення:

- 1) `scikit-fuzzy` (0.4.2) – реалізація алгоритму Fuzzy C-Means кластеризації. Бібліотека надає функцію `fuzz.cluster.cmeans()` для навчання моделі та `fuzz.cluster.cmeans_predict()` для класифікації нових спостережень. Підтримує налаштування параметра розмитості `m` (`fuzziness`) та кількості кластерів.
- 2) `pandas` (2.0+) – обробка та аналіз часових рядів фінансових даних.

Використовується для:

- a) завантаження історичних даних у форматі CSV/Excel;
- b) ресемплювання даних (M15 → H4 → D1) для багатотаймфреймового аналізу;
- c) розрахунку технічних індикаторів (rolling window операції);
- d) фільтрації даних за часовими діапазонами.

3) NumPy (1.24+) – математичні обчислення та операції з масивами.

Забезпечує:

- a) швидкі векторизовані операції над масивами даних;
- b) реалізацію математичних функцій для індикаторів;
- c) ефективне використання пам'яті для великих датасетів.

4) scikit-learn (1.3+) – препроцесинг даних та оцінка моделі:

- a) StandardScaler для нормалізації ознак (z-score standardization);
- b) SimpleImputer для обробки пропущених значень;
- c) train\_test\_split для розділення даних на навчальну/тестову вибірки.

5) joblib (1.3+) – серіалізація та збереження натренованих моделей:

- a) збереження центрів кластерів (fuzzy\_centers\_filtered.pkl);
- b) збереження параметрів StandardScaler (fuzzy\_scaler\_filtered.pkl);
- c) ефективне зберігання великих NumPy масивів.

6) matplotlib / seaborn – візуалізація результатів:

7) графіки equity curve для оцінки динаміки капіталу;

8) розподіл прибутків/збитків за угодами;

9) кореляційні матриці між індикаторами.

Додаткові інструменти:

1) MetaTrader 5 Terminal – торгова платформа для тестування стратегії на реальних даних через Strategy Tester та можливість виконання угод на демо/live рахунку.

2) MQL5 – мова розробки Expert Advisors для MT5. Створено інтеграційний EA (FuzzyFilteredEA\_Enhanced.mq5) який збирає дані M15/H4/D1 та відправляє на Python сервер через TCP/IP з'єднання.

3) Git – система контролю версій для відстеження змін у коді та співпраці.

Середовище розробки:

1) IDE: Visual Studio Code / PyCharm

- 2) Jupyter Notebook – для інтерактивного аналізу та дослідження даних
- 3) віртуальне середовище `venv` для ізоляції залежностей

Апаратні вимоги:

- 1) CPU: мінімум 2 ядра, рекомендовано 4+ (для паралельної оптимізації)
- 2) RAM: мінімум 4 GB, рекомендовано 8+ GB (для обробки великих датасетів)
- 3) HDD: 2 GB вільного місця для даних та моделей

Обрана архітектура інструментів забезпечує баланс між продуктивністю, зручністю розробки та можливістю масштабування системи для роботи з множиною торгових інструментів.

### 3.2 Архітектура програмного забезпечення

Програмна система побудована за модульним принципом з використанням об'єктно-орієнтованого підходу. Загальна архітектура складається з п'яти основних шарів (рис. 3.1).

Структура класів та взаємозв'язків між ними представлена на UML діаграмі класів (рис. 3.2).

Система складається з дев'яти основних модулів, кожен з яких реалізує конкретну функціональність. Модулі організовані у трьох директоріях:

- `src/data/` – завантаження та обробка даних;
- `src/strategy/` – торгові стратегії;
- `src/engine/` – механізм бектестингу та управління портфелем.

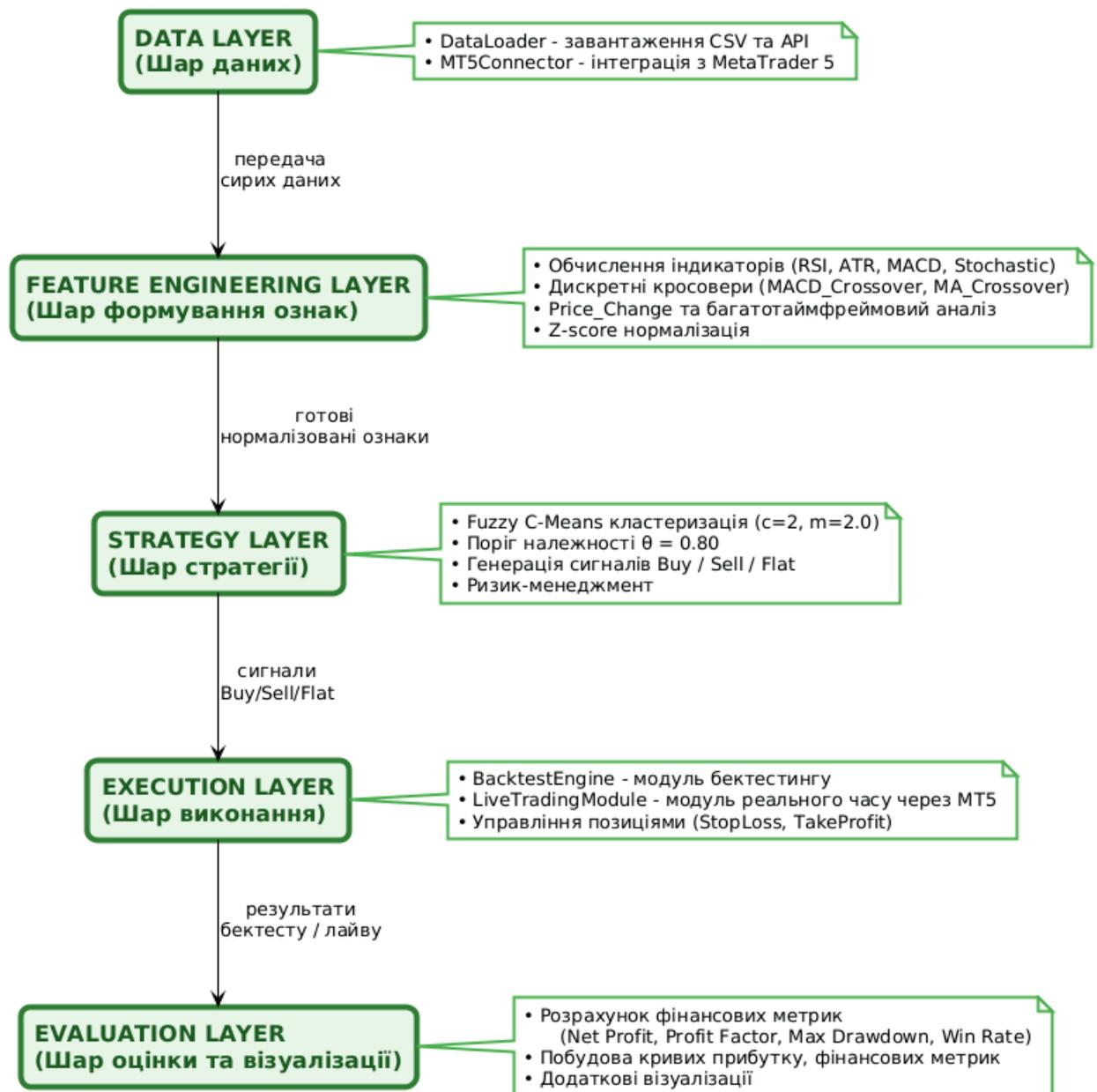


Рисунок 3.1 – Загальна архітектура системи

Основними модулями системи є:

- модуль завантаження даних (DataLoader). Відповідає за завантаження історичних котировань з різних джерел (CSV-файли, бази даних) та валідацію цілісності даних;
- модуль нечіткої кластеризації (FuzzyClusteringModule). Реалізує логіку алгоритму Fuzzy C-Means. Відповідає за навчання моделі, обчислення належності до кластерів та збереження/завантаження параметрів моделі;

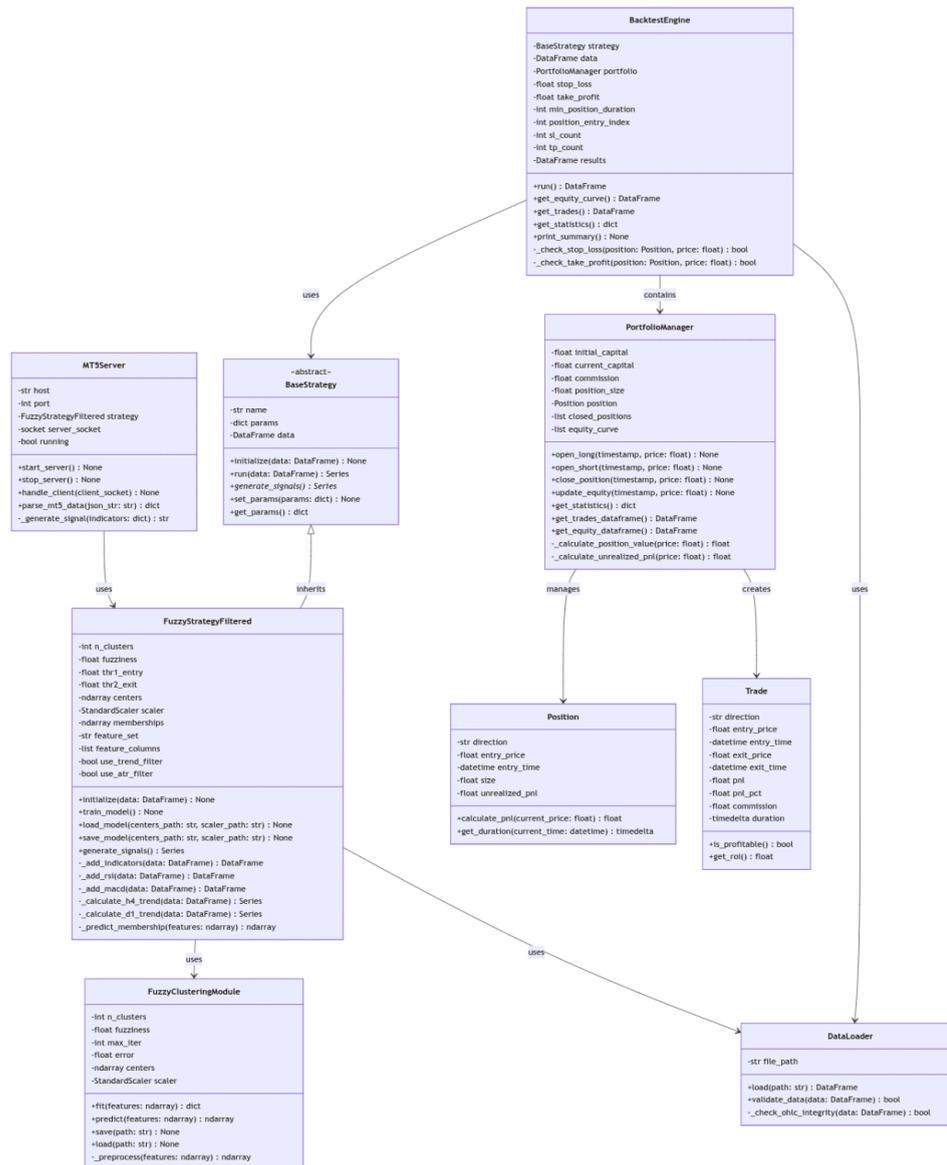


Рисунок 3.2 – UML-діаграма класів

- модуль торгової стратегії (FuzzyStrategyFiltered). Модуль реалізує повну логіку торгової стратегії на основі нечіткої кластеризації з багатофільтровою системою валідації сигналів;
- модуль бектестингу (BacktestEngine). Модуль симулює історичну торгівлю на основі сигналів стратегії. Враховує реалістичні умови: Stop-Loss, Take-Profit, комісії, slippage;

- модуль управління портфелем (PortfolioManager). Управляє капіталом, відкриває та закриває позиції, розраховує фінансові результати, веде історію угод;
- модуль інтеграції з MetaTrader 5 (MT5Server). Забезпечує роботу системи в режимі реального часу через інтеграцію з торговою платформою MetaTrader 5.

### 3.2.1 Шар даних

Відповідає за завантаження, валідацію та первинну обробку ринкових даних. Реалізується у класі DataLoader:

Основні функції:

- завантаження історичних даних у форматі OHLCV (Open, High, Low, Close, Volume);
- валідація цілісності даних (перевірка на gaps, дублікати, некоректні значення);
- конвертація часових міток у pandas DatetimeIndex;
- підтримка множини джерел даних (CSV, API, MT5 Terminal).

Формат даних: Кожен рядок датасету містить:

- datetime – мітка часу (індекс)
- open – ціна відкриття свічки
- high – максимальна ціна
- low – мінімальна ціна
- close – ціна закриття
- volume – обсяг торгів (опціонально)

### 3.2.2 Шар формування ознак

Реалізує розрахунок технічних індикаторів та підготовку вхідних даних для моделі кластеризації.

Набори індикаторів (Feature Sets):

1) default (6 індикаторів):

- RSI (14) – Relative Strength Index;
- ATR (14) – Average True Range;
- Price\_Change – зміна ціни;
- MACD\_Crossover – перетин MACD з сигнальною лінією;
- MA\_Crossover – перетин MA(10) з MA(50);
- Stochastic\_Oscillator – стохастичний осцилятор.

2) extended (12 індикаторів):

- default набір +;
- S%K, S%D – компоненти стохастика;
- Intersection – перетин %K з %D;
- macd1, macd2 – компоненти MACD;
- VWAP – Volume Weighted Average Price.

3) minimal (3 індикатори): RSI, MACD\_Crossover, ATR

Код для розрахунку індикатору RSI наведений на рисунку 3.3.

Для реалізації багатотаймфреймового аналізу з метою визначення глобального тренду використовується ресемплювання даних з M15 на старші таймфрейми (рис. 3.4).

На H4/D1 даних розраховуються:

- EMA(20), EMA(50) – експоненціальні ковзні середні;
- MACD(12, 26, 9) – індикатор конвергенції-дивергенції.

Умови тренду (для D1):

- UPTREND:  $EMA(20) > EMA(50)$  AND  $price > EMA(20)$  AND  $MACD > Signal$

- DOWNTREND:  $EMA(20) < EMA(50)$  AND  $price < EMA(20)$  AND  $MACD < Signal$
- NEUTRAL: інші випадки

```
def rsi(data: pd.Series, period: int = 14) -> pd.Series:
    """
    Relative Strength Index (RSI)

    Args:
        data: Series з цінами
        period: Період для розрахунку (за замовчуванням 14)

    Returns:
        Series з RSI значеннями (0-100)
    """
    if TA_AVAILABLE:
        indicator = RSIIndicator(close=data, window=period)
        return indicator.rsi()
    else:
        # Ручний розрахунок RSI
        delta = data.diff()
        gain = (delta.where(delta > 0, other=0)).rolling(window=period).mean()
        loss = (-delta.where(delta < 0, other=0)).rolling(window=period).mean()

        rs = gain / loss
        rsi = 100 - (100 / (1 + rs))
        return rsi
```

Рисунок 3.3 – Код для розрахунку індикатора RSI

```
def _resample_to_h4(self, data: pd.DataFrame) -> pd.DataFrame:
    """
    Resample M15 data to H4 timeframe

    M15 → H4: 16 свічок M15 = 1 свічка H4 (4 години / 15 хвилин = 16)
    """
    # Resample to 4H
    h4_data = data.resample('4H').agg({
        'open': 'first',
        'high': 'max',
        'low': 'min',
        'close': 'last',
    })

    # Видаляємо NaN (якщо є)
    h4_data = h4_data.dropna()

    return h4_data
```

Рисунок 3.4 – Код для ресемплювання з M15 в H4

Перед подачею до моделі FCM всі ознаки нормалізуються методом z-score:

```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(features)
```

Нормалізація забезпечує:

- рівний вплив всіх індикаторів на кластеризацію;
- стабільність чисельних обчислень;
- можливість порівняння моделей.

### 3.2.3 Шар стратегії

Шар стратегії є центральним компонентом системи, що реалізує логіку прийняття торгових рішень на основі нечіткої кластеризації. Головним класом є клас `FuzzyStrategyFiltered`, який успадковується від абстрактного класу `BaseStrategy`.

Головними методами класу є:

- `_get_trading_session(self, timestamp: pd.Timestamp) -> str` визначає торгову сесію для заданого часу (UTC);
- `_is_allowed_session(self, timestamp: pd.Timestamp) -> bool` перевіряє чи дозволена торгівля в поточній сесії;
- `_calculate_h4_trend(self, data: pd.DataFrame) -> pd.Series` розраховує тренд на H4 таймфреймі;
- `_calculate_d1_trend(self, data: pd.DataFrame) -> pd.Series` Розраховує тренд на D1 (Daily) таймфреймі;
- `_calculate_indicators(self, data: pd.DataFrame) -> pd.DataFrame:`  
Розраховує технічні індикатори;
- `_select_features(self, data: pd.DataFrame) -> pd.DataFrame:`  
Вибирає features для кластеризації в залежності від `feature_set` (“default”, “minimal”, “extended”);

- `train(self, train_data: pd.DataFrame, verbose: bool = True) -> dict`  
Навчає Fuzzy C-Means модель;
- `_predict_membership(self, single_data: np.ndarray) -> np.ndarray`  
прогнозує membership для однієї точки даних;
- `initialize(self, data: pd.DataFrame) -> None`: Ініціалізація для prediction;
- `generate_signals(self) -> pd.Series` генерує торгові сигнали з фільтрами;
- `save_model(self, centers_path: str = "fuzzy_centers_filtered.pkl",  
scaler_path: str = "fuzzy_scaler_filtered.pkl")` зберігає модель;
- `load_model(self, centers_path: str = "fuzzy_centers_filtered.pkl",  
scaler_path: str = "fuzzy_scaler_filtered.pkl")` завантажує модель.

Для зменшення overtrading та покращення якості угод у роботі реалізовано 5 типів фільтрів:

1. Hysteresis Filter (Гістерезис): Використовує різні пороги для входу та виходу з позиції:

`thr1_entry = 0.8` – вхід у LONG (`level1 > 0.8`)

`thr2_exit = 0.45` – вихід з LONG (`level1 ≤ 0.45`)

`thr3_entry = 0.8` – вхід у SHORT (`level2 > 0.8`)

`thr4_exit = 0.45` – вихід з SHORT (`level2 ≤ 0.45`)

Різниця між порогами ( $0.8 - 0.45 = 0.35$ ) створює "мертву зону", що знижує частоту переключень між позиціями.

2. Min Position Duration Filter: Мінімальна тривалість утримання позиції (за замовчуванням 50 свічок = 12.5 годин для M15):

`if position_bars_count < min_position_duration:`

`signal = 0 # утримувати позицію`

Запобігає передчасному закриттю позицій при короткострокових флуктуаціях membership values.

3. ATR Filter (Average True Range): Дозволяє відкривати позиції тільки при достатній волатильності ринку:

```
if current_atr > min_atr_threshold: # наприклад, 0.0004 для EURUSD
    allow_entry = True
```

Знижує ризик торгівлі у флетових умовах з низьким потенціалом прибутку.

4. Trend Filter: Дозволяє торгувати тільки в напрямку глобального тренду (H4 або D1):

```
if trend_d1 == 'UPTREND':
    allow_long = True
    allow_short = False
elif trend_d1 == 'DOWNTREND':
    allow_long = False
    allow_short = True
```

Значно підвищує win rate за рахунок виключення контртрендових угод.

5. Session Filter: Обмежує торгівлю активними торговими сесіями (London, New York):

```
current_hour = datetime.now().hour
if 8 <= current_hour < 17: # London session
    is_session_allowed = True
```

Уникає періодів низької ліквідності (азіатська сесія, вихідні).

### 3.2.4 Шар виконання

Для тестування на історичних даних розроблений клас BacktestEngine (рис. 3.5).

Клас BacktestEngine здійснює симуляцію торгівлі на історичних даних. Особливості класу такі:

1. Вхідні дані: часовий ряд у форматі OHLCV (Open, High, Low, Close, Volume).

```

10 class BacktestEngine:
11     """Движок для запуску backtesting на історичних даних"""
12
13     def __init__(self, strategy: BaseStrategy,
14                 data: pd.DataFrame,
15                 initial_capital: float = 10000.0,
16                 commission: float = 0.0001,
17                 position_size: float = 0.95,
18                 stop_loss: Optional[float] = None,
19                 take_profit: Optional[float] = None,
20                 min_position_duration: int = 0):
21
22         """
23         Args:
24             strategy: Торгова стратегія
25             data: DataFrame з OHLCV даними
26             initial_capital: Початковий капітал
27             commission: Комісія за угоду
28             position_size: Частка капіталу для позиції (0.0-1.0, за замовчуванням 0.95 = 95%)
29             stop_loss: Stop-loss в % (наприклад 0.02 = 2% збитку)
30             take_profit: Take-profit в % (наприклад 0.05 = 5% прибутку)
31             min_position_duration: Мінімальна тривалість позиції в свічках (0 = без обмеження)
32         """

```

Рисунок 3.5 – Клас BacktestEngine

## 2. Інтеграція з торговою стратегією:

- отримання сигналів: викликає методи стратегії для генерації торгових рішень (BUY/SELL/HOLD).
- виконання сигналів: автоматично відкриває/закриває позиції згідно з сигналами.
- гнучкість: працює з будь-якою стратегією, що успадковується від BaseStrategy.

## 3. Управління ризиками.

### Stop-Loss (автоматичний захист від збитків):

- приклад, stop\_loss=0.005 # 0.5% максимальний збиток на угоду;
- автоматично закриває позицію при досягненні заданого рівня збитку;
- працює незалежно від сигналів стратегії;
- захищає від катастрофічних втрат.

### Take-Profit (фіксація прибутку):

- приклад, take\_profit=0.01 # 1% цільовий прибуток;
- автоматично закриває позицію при досягненні цільового прибутку;
- запобігає "віддачі" вже заробленого profit.

Мінімальна тривалість позиції:

- приклад, `min_position_duration=50` # мінімум 50 свічок (12.5 годин для M15);
- запобігає передчасному закриттю позицій;
- знижує кількість "хибних" виходів;
- зменшує `overtrading` (надмірну частоту угод).

Розмір позиції:

- `position_size=0.95` # використовувати 95% капіталу на угоду
- Контролює `exposure` (наскільки агресивно торгувати)
- 0.95 = ризикуємо 95% капіталу
- 0.50 = консервативніше, тільки 50% капіталу

5. Відстеження еволюції капіталу (крива капіталу):

- відстежує зміну капіталу на кожній свічці;
- враховує:
  - а) закриті угоди (`realized P&L`);
  - б) відкриті позиції (`unrealized P&L`);
  - в) комісії.

6. Управління позиціями.

Відкриття позиції:

- LONG: купити актив (очікування зростання ціни);
- SHORT: продати актив (очікування падіння ціни);
- зберігає `entry price`, `entry time`, `position size`.

Закриття позиції, позиція закривається при:

- зворотному сигналі (`SELL` при відкритому LONG);
- сигналі HOLD;
- спрацюванні `Stop-Loss`;
- спрацюванні `Take-Profit`;
- закінченні тестового періоду.

7. Збір статистики та метрик.

Фінансові метрики обчислюються за допомогою функції `get_statistics()`:

```
stats = engine.get_statistics()
```

Ця функція повертає:

- `initial_capital` – початковий капітал (\$);
- `final_equity` – фінальний капітал (\$);
- `total_return_%` – загальна прибутковість (%);
- `total_trades` – кількість угод;
- `closed_positions` – кількість закритих позицій;
- `win_rate_%` – відсоток прибуткових угод;
- `profit_factor` – співвідношення прибутку до збитку;
- `max_drawdown_%` – максимальна просадка капіталу;
- `sharpe_ratio` – коефіцієнт Шарпа (risk-adjusted return);
- `total_commission` – загальна комісія (\$);
- `sl_exits` – кількість виходів за Stop-Loss;
- `tp_exits` – кількість виходів за Take-Profit.

8. Візуалізація результатів здійснюється за допомогою функції `get_equity_curve()`:

```
equity_curve = engine.get_equity_curve()
```

Ця функція повертає DataFrame з колонками: `timestamp`, `equity`.

Підсумок друкується за допомогою функції `print_summary()`. Приклад виводу наведено на рисунку 3.6.

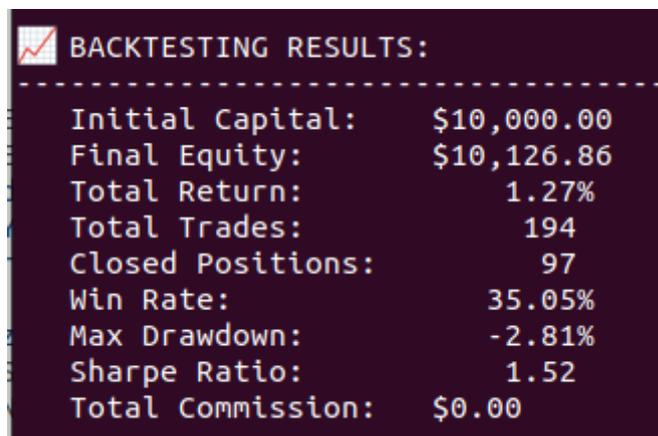


Рисунок 3.6 – Вивід результату шару виконання

### 3.2.5 Шар оцінки

Модуль `src/metrics/` реалізує розрахунок фінансових та статистичних метрик для оцінки ефективності стратегії.

Основні метрики:

- Total Return (%) – відображає загальну прибутковість стратегії за весь період;
- Sharpe Ratio – характеризує ефективність стратегії з урахуванням прийнятого ризику. Усі значення вище 1.9 свідчать про високу якість стратегії та стабільність прибутку;
- Maximum Drawdown (%) – максимальна просадка, найважливіший показник ризику;
- Win Rate (%) – відсоток прибуткових угод до всієї кількості угод;
- Profit Factor – співвідношення сумарного прибутку до сумарного збитку. Значення вище 2.0 свідчить про стійку прибутковість навіть при серії програшних угод.

### 3.3 Тестування системи на історичних даних

Для валідації ефективності розробленої системи було здійснено тестування історичних даних. Для цього були підготовлені два датасети у вигляді csv файлів, для валютних пар EUR/USD та GBD/USD і таймфрейму 15М (15-ти хвилинного). Датасети підготовлені при використанні засобів MetaTrader 5 (рисунок 3.7)

Датасети містять 35137 барів, в діапазоні дат від 2024.01.01 до 2025.06.01 та зберігаються у форматі CSV. Датасети мають такі стовпці як “Дата”, “Відкриття”, “Максимум”, “Мінімум”, “Закриття”, “Тікові обсяги”, “Обсяг” та “Спред”.

Дата	Відкриття	Максимум	Мінімум	Закриття	Тікові обсяги	Обсяг	Спред
2025.03.14 22:45	1.08835	1.08839	1.08746	1.08751	260	0	4
2025.03.14 22:30	1.08829	1.08853	1.08823	1.08835	146	0	4
2025.03.14 22:15	1.08827	1.08843	1.08819	1.08828	178	0	4
2025.03.14 22:00	1.08824	1.08836	1.08809	1.08826	248	0	4
2025.03.14 21:45	1.08814	1.08846	1.08811	1.08831	417	0	2
2025.03.14 21:30	1.08816	1.08838	1.08810	1.08815	428	0	2
2025.03.14 21:15	1.08807	1.08818	1.08782	1.08814	374	0	2
2025.03.14 21:00	1.08799	1.08817	1.08786	1.08807	419	0	2
2025.03.14 20:45	1.08767	1.08811	1.08752	1.08803	497	0	2
2025.03.14 20:30	1.08753	1.08811	1.08732	1.08768	507	0	2
2025.03.14 20:15	1.08789	1.08821	1.08753	1.08756	452	0	2
2025.03.14 20:00	1.08822	1.08863	1.08781	1.08790	553	0	2
2025.03.14 19:45	1.08828	1.08841	1.08803	1.08822	484	0	2
2025.03.14 19:30	1.08819	1.08880	1.08804	1.08830	623	0	2
2025.03.14 19:15	1.08797	1.08832	1.08797	1.08818	433	0	2
2025.03.14 19:00	1.08829	1.08829	1.08789	1.08795	572	0	2

Рисунок 3.7 - Завантаження датасету з історичними даними EUR/USD

Для навчання кожен датасет був поділений на тренувальну та тестову вибірку у співвідношенні 80% на 20%. Причому розділення відбувалося послідовно у часі. Модель навчається тільки на тренувальних даних. Оцінка ефективності здійснюється виключно на тестових даних. Це дозволяє симулювати реальні умови торгівлі.

Результати досліджень при наборі параметрів  $THR1\_ENTRY = 0.8$ ,  $THR2\_EXIT = 0.45$ ,  $THR3\_ENTRY = 0.8$ ,  $THR4\_EXIT = 0.45$ ,  $USE\_ATR\_FILTER = False$ ,  $N\_CLUSTERS = 2$ ,  $FUZZINESS = 2.0$ ,  $STOP\_LOSS = 0.005$ ,  $TAKE\_PROFIT = 0.01$ ,  $INITIAL\_CAPITAL = 10000.0$ ,  $POSITION\_SIZE = 0.95$ ,  $TRAIN\_RATIO = 0.8$ .

В подальших експериментах змінними будуть параметри:  $USE\_TREND\_FILTER = True/False$ ,  $TREND\_TIMEFRAME = 'D1'/'H1'$ ,  $USE\_SESSION\_FILTER = True/False$ ,  $ALLOWED\_SESSIONS = ['london', 'newyork', 'asian']/['london', 'newyork']$ ,  $MIN\_POSITION\_DURATION = 1/10/25/50$ .

Для дослідження впливу використання трендового фільтру на результат було проведено два експерименти на парі EUR/USD при значеннях  $USE\_TREND\_FILTER = True/False$ . Інші змінні параметри мали значення:

TREND\_TIMEFRAME = 'D1', USE\_SESSION\_FILTER = False,  
MIN\_POSITION\_DURATION = 1. Результати занесені у таблицю 3.1.

Таблиця 3.1 – Дослідження впливу фільтрації за D1 для пари EUR/USD

USE_TREN D_FILTER	Кількість угод	Коефіцієнт Шарпа	WinRate (%)	Максимальн а просадка (%)	Загальний прибуток (%)
ні	494	1.53	36.44	-1.89	+2.45
так	<b>360</b>	<b>2.13</b>	36.11	<b>-1.87</b>	<b>+3.15</b>

Як видно з таблиці 3.1 використання трендового фільтру покращило результати тому в подальших дослідженнях він буде використовуватися. Далі буде досліджено вплив тривалості тримання позиції на результат при USE\_SESSION\_FILTER = False, USE\_TREND\_FILTER = True, TREND\_TIMEFRAME = 'D1' (табл. 3.2).

Таблиця 3.2 – Дослідження тривалості тримання позиції для EUR/USD

MIN_POSIT ION_DURA TION	Кількість угод	Коефіцієнт Шарпа	WinRate (%)	Максимальн а просадка (%)	Загальний прибуток (%)
1	360	2.13	36.11	<b>-1.87</b>	+3.15
10	340	2.05	40.59	-1.93	+3.44
25	250	1.95	44.0	-2.26	+3.67
50	<b>178</b>	<b>2.79</b>	<b>51.69</b>	-2.81	<b>+6.13</b>
100	124	1.19	45.16	-3.73	+2.77

Найкращий результат згідно таблиці 3.2 показав MIN\_POSITION\_DURATION = 50, тому він буде використовуватися в подальших експериментах.

У наступному експерименті досліджувався вплив фільтрації за сесіями, з можливими варіантами “лише Лондонська сесія”, “Лондонська та Нью-йоркська сесії”, “Лондонська, Нью-йоркська та Азіатська сесії” та “всі періоди” при фіксованих значеннях  $USE\_SESSION\_FILTER = True$ ,  $USE\_TREND\_FILTER = True$ ,  $TREND\_TIMEFRAME = 'D1'$ . Результати для пари EUR/USD наведені у таблиці 3.3, для пари GBD/USD у таблиці 3.4.

Таблиця 3.3 – Дослідження впливу фільтрації за сесіями для EUR/USD

Дозволені сесії	Кількість угод	Коефіцієнт Шарпа	WinRate (%)	Максимальна просадка (%)	Загальний прибуток (%)
'london'	<b>114</b>	<b>3.85</b>	<b>63.16</b>	<b>-2.35</b>	+7.47
'london', 'newyork'	126	3.57	58.73	-2.53	+7.00
'london', 'newyork', 'asian'	168	3.61	58.33	-2.49	<b>+7.88</b>
всі	178	2.79	51.69	-2.81	+6.13

Таблиця 3.4 – Дослідження впливу фільтрації за сесіями для GBD/USD

ALLOWED_SESSIONS	Кількість угод	Коефіцієнт Шарпа	WinRate (%)	Максимальна просадка (%)	Загальний прибуток (%)
'london'	<b>130</b>	0.8	52.31	-4.49	+1.31
'london', 'newyork'	138	0.67	50.72	-4.49	+1.09
'london', 'newyork', 'asian'	184	2.77	54.35	-3.38	+5.23
всі	190	<b>4.11</b>	<b>55.79</b>	<b>-1.67</b>	<b>+7.96</b>

В наступному експерименті результати порівнювалися з базовими стратегіями:

- Buy and Hold – купити і утримувати;
- Simple MA Crossover – перетин ковзних середніх.

Для експерименту були використані параметри USE\_SESSION\_FILTER = False, USE\_TREND\_FILTER = True, TREND\_TIMEFRAME = 'D1', MIN\_POSITION\_DURATION = 50 на валютній парі GBD/USD (табл. 3.5).

Таблиця 3.5 – Порівняння з базовими стратегіями для пари GBD/USD

Стратегія	Кількість угод	Коефіцієнт Шарпа	WinRate (%)	Максимальна просадка (%)	Загальний прибуток (%)
Купити і утримувати	<b>2</b>	2.84	<b>100.0</b>	-3.47	+4.27
Перетин ковзних середніх	382	1.92	50.72	-2.62	+6.47
Запропонована	190	<b>4.11</b>	55.79	<b>-1.67</b>	<b>+7.96</b>

Як видно з таблиці 3.5 найбільш збалансовані результати показує запропонована система, як з точки зору загального прибутку так і по параметрам максимальна просадка та коефіцієнт Шарпа. Хоча на цих тестових даних стратегія купити і тримати показує 100% WinRate але це пояснюється лише особливістю тестових даних які були використані коли курс валют за декілька місяців лише підіймався угору, якби було б навпаки результат би був зворотнім. Крім того лиш дві угоди на відкриття позиції та закриття не дозволяють отримати максимальний прибуток з фінансового часового ряду. Стратегія перетину ковзних середніх хоча й показала гарний результат але все ж була гіршою за запропоновану за всіма критеріями, і особливо за великою, двічі більшою, кількістю угод, що при врахуванні комісії за відкриття позиції значно погіршить її результат.

Візуалізація цін за тестовий період, угод та їх тривалості (червоні - збиткові, зелені прибуткові), періодів фільтрації, кривої прибутку, кривої просадки депозиту наведені на рисунку 3.8. Візуалізація додаткових статистичних показників торгівлі за тестовий період запропонованої стратегії, зокрема розподіл по величині прибутків та втрат (P&L - Profit and Loss),

розподіл по тривалості тримання позиції (Position Duration), візуалізація відсотку WinRate, помісячний прибуток та Win/Loss аналіз вигравшів/програшів підряд наведено на рисунку 3.9.

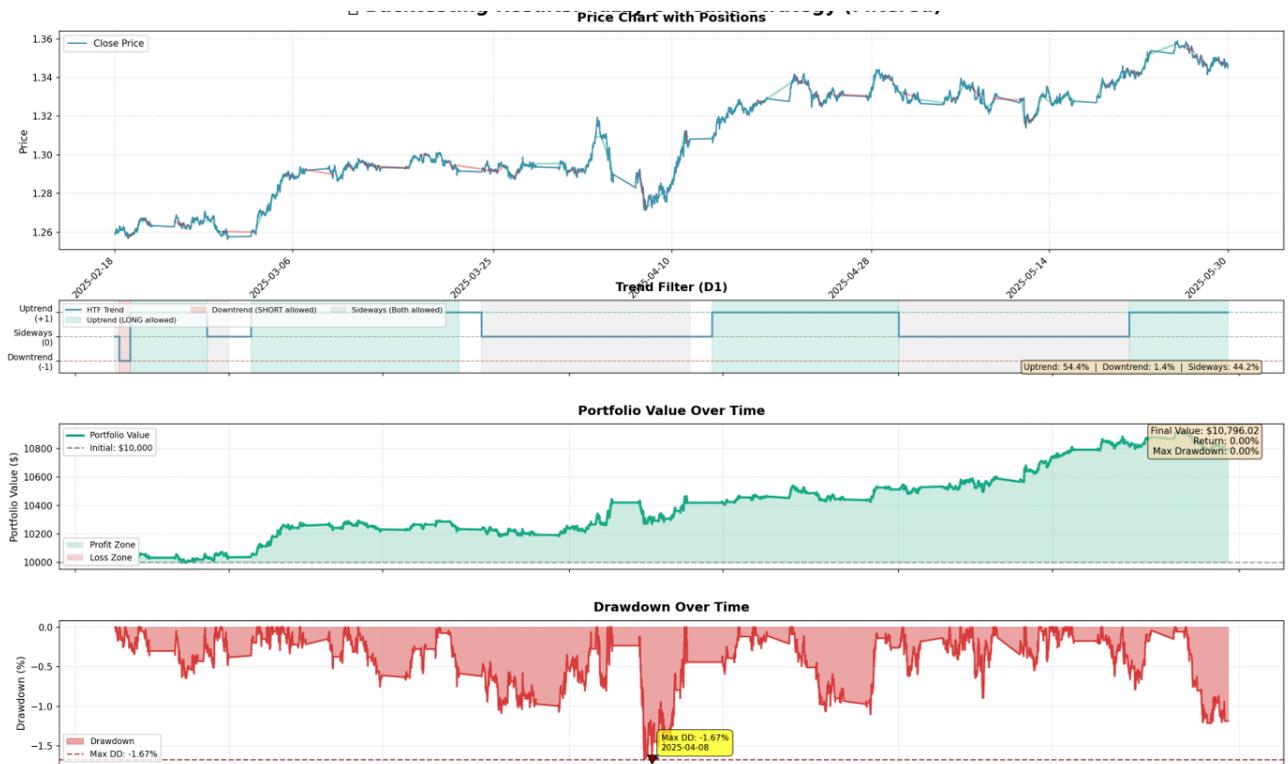


Рисунок 3.8 – Візуалізація результату за тестовий період запропонованої стратегії на валютній парі GBD/USD

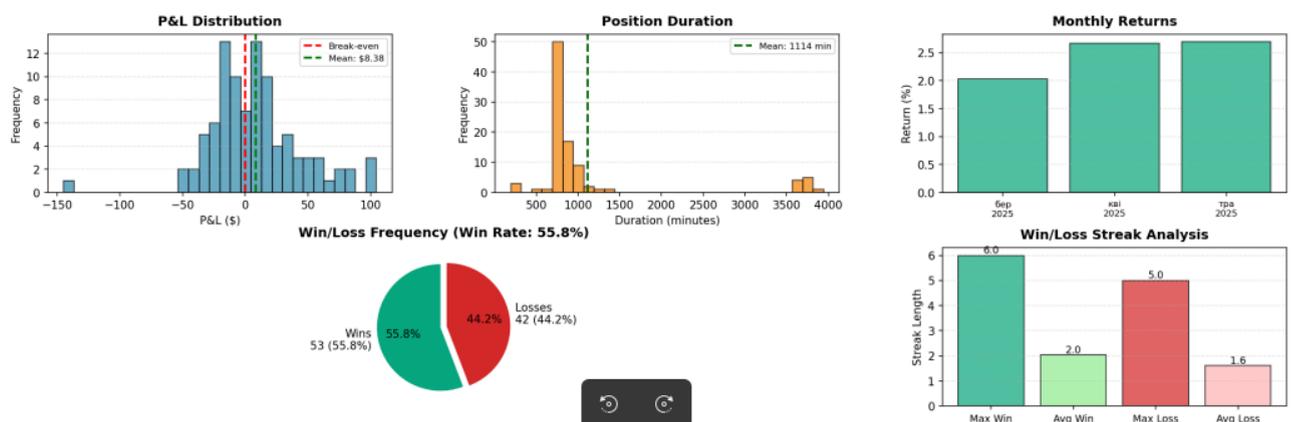


Рисунок 3.9 – Візуалізація додаткових статистичних показників торгівлі за тестовий період запропонованої стратегії на валютній парі GBD/USD

Приклад частини звіту у форматі csv по позиціям що були відкриті наведено на рисунку 3.10.

1	entry_time	exit_time	direction	entry_price	exit_price	size	pnl	pnl_percent	pips	exit_reason
2	2025-02-18 15:30:00	2025-02-19 04:00:00	LONG	1.25966	1.26134	7541.7176063382	12.670085578647482	0.13336932188049982	16.799999999999038	SIGNAL
3	2025-02-19 05:00:00	2025-02-19 17:30:00	SHORT	1.26118	1.2597	7542.1720779743	11.162414675401507	0.1173504178626307	14.799999999999258	SIGNAL
4	2025-02-19 18:45:00	2025-02-20 07:15:00	SHORT	1.25714	1.25961	7574.8451844992	-18.70986760571286	-0.1964777192675416	24.699999999999722	SIGNAL
5	2025-02-20 07:30:00	2025-02-21 00:00:00	LONG	1.25929	1.26628	7547.7979663269	52.759107784626075	0.5550746849415187	69.900000000000052	SIGNAL
6	2025-02-21 05:15:00	2025-02-21 17:45:00	LONG	1.26713	1.26384	7540.6530138275	-24.808748415492783	-0.25964186784307963	32.900000000000015	SIGNAL
7	2025-02-24 01:30:00	2025-02-24 14:15:00	LONG	1.26557	1.2637	7531.3252861687	-14.083578285135868	-0.14775950757366546	18.700000000000383	SIGNAL
8	2025-02-24 15:30:00	2025-02-25 08:30:00	LONG	1.26441	1.2632	7527.6531687077	-9.10846033413583	-0.09569680720651792	12.099999999999334	SIGNAL
9	2025-02-25 14:00:00	2025-02-26 03:30:00	LONG	1.26523	1.26657	7515.9353680582	10.071353393197226	0.10590959746448446	13.399999999999898	SIGNAL
10	2025-02-26 15:45:00	2025-02-27 04:15:00	LONG	1.2654	1.26629	7522.4867168103	6.6950131779599795	0.07033349138610995	8.8999999999998354	SIGNAL
11	2025-02-27 04:30:00	2025-02-27 20:45:00	SHORT	1.26645	1.26284	7521.2720233494	27.1517920042923	0.28504875834025134	36.100000000000113	SIGNAL
12	2025-02-27 22:15:00	2025-02-28 10:45:00	SHORT	1.26049	1.25998	7577.2986349554	3.8644223038273466	0.04046045585447012	5.1000000000001044	SIGNAL
13	2025-02-28 11:15:00	2025-03-03 05:00:00	LONG	1.26032	1.25997	7581.2336212736	-2.6534317674463255	-0.02777072489527044	3.5000000000007248	SIGNAL
14	2025-03-03 11:30:00	2025-03-04 00:00:00	LONG	1.26163	1.2696	7571.3637099502	60.343768768303704	0.6317224542853318	79.700000000000033	SIGNAL
15	2025-03-04 09:00:00	2025-03-05 00:45:00	LONG	1.2703	1.2789	7564.8163250526	65.05742039545258	0.6770054317877621	85.99999999999994	SIGNAL
16	2025-03-05 06:15:00	2025-03-06 00:00:00	LONG	1.27874	1.28875	7563.2190492907	75.7078226834013	0.7828018205421019	100.100000000000075	SIGNAL
17	2025-03-06 04:00:00	2025-03-06 19:00:00	LONG	1.29023	1.28853	7551.6095259289	-12.8377361940795	-0.1317594537408086	17.000000000000348	SIGNAL
18	2025-03-07 06:00:00	2025-03-07 18:30:00	LONG	1.2888	1.29131	7550.5255348036	18.951819092357265	0.19475481067659933	25.100000000000122	SIGNAL
19	2025-03-07 22:45:00	2025-03-10 11:15:00	LONG	1.29235	1.2898	7543.7161275139	-19.236476125161037	-0.1973149688551903	25.500000000000522	SIGNAL
20	2025-03-11 02:15:00	2025-03-11 15:00:00	LONG	1.28859	1.29339	7551.5461745580	36.24742163787979	0.37250017460946755	48.000000000000138	SIGNAL
21	2025-03-11 17:30:00	2025-03-12 06:00:00	LONG	1.29492	1.29213	7541.2241185785	-21.040015290834763	-0.2154573255490741	27.900000000000702	SIGNAL
22	2025-03-12 10:30:00	2025-03-12 23:00:00	LONG	1.29322	1.29599	7535.6814162350	20.873837522970604	0.21419402731166692	27.699999999999939	SIGNAL
23	2025-03-13 03:00:00	2025-03-13 15:30:00	LONG	1.29678	1.29364	7530.2858362639	-23.64509752586972	-0.24213821928161622	31.4000000000001427	SIGNAL
24	2025-03-13 22:30:00	2025-03-14 11:00:00	LONG	1.29503	1.29253	7523.1162398560	-18.807790599639617	-0.19304572094854536	24.999999999999467	SIGNAL
25	2025-03-14 14:30:00	2025-03-17 06:00:00	LONG	1.29408	1.2932	7514.8320220010	-6.613052179360828	-0.06800197823936635	8.799999999999919	SIGNAL
26	2025-03-17 10:00:00	2025-03-17 22:30:00	LONG	1.29439	1.29919	7508.1786968847	36.03925774504791	0.3708310478294901	48.000000000000138	SIGNAL

Рисунок 3.10 – Торговий звіт у форматі csv

Як видно з рисунку у торговому звіті детально відображаються елементи угод, зокрема такі як:

- час відкриття угоди;
- час закриття угоди;
- напрямок - SHORT чи LONG;
- початкова ціна;
- кінцева ціна;
- прибуток чи втрата - pnl;
- відсоток pnl;
- прибуток в піпсах;
- причина закриття - SIGNAL, TAKE PROFIT чи STOP LOSS.

### 3.4 Висновки до розділу

У третьому розділі представлено розробку та тестування програмного забезпечення системи автоматичного прийняття торгових рішень на основі

нечіткої кластеризації. Для реалізації обрано мову програмування Python 3.8+ з потужним технологічним стеком, що включає scikit-fuzzy для алгоритму Fuzzy C-Means, pandas для роботи з часовими рядами, scikit-learn для препроцесингу даних та MetaTrader 5 для інтеграції з торговою платформою. Система побудована за модульним принципом з п'ятьма основними шарами: шар даних, шар формування ознак, шар стратегії, шар виконання та шар оцінки. Ключовою особливістю розробленої системи є багатофільтрова архітектура, що включає фільтри гістерезису, мінімальної тривалості позиції, ATR-фільтр волатильності, трендовий фільтр на основі старших таймфреймів (H4/D1) та сесійний фільтр для торгівлі в періоди максимальної ліквідності.

Тестування системи проведено на історичних даних валютних пар EUR/USD та GBP/USD за період з 01.01.2024 до 01.06.2025 (35,137 барів таймфрейму M15) з розділенням на тренувальну (80%) та тестову (20%) вибірки.

Експериментальні дослідження виявили оптимальні параметри системи: використання трендового фільтру D1 підвищило прибутковість (з +2.45% до +3.15% для EUR/USD), мінімальна тривалість утримання позиції 50 свічок забезпечила найкращу стабільність результатів, а комбінація фільтрації за торговими сесіями показала різні оптимуми для різних валютних пар. Порівняння з базовими стратегіями (Buy and Hold та MA Crossover) на парі GBP/USD продемонструвало, що запропонована система показує найбільш збалансовані результати: Total Return 7.96%, Win Rate 55.79%, Sharpe Ratio 4.11 при максимальній просадці -1.67%, що значно перевершує базові стратегії за комплексом показників ризик-прибутковість.

Розроблена система забезпечує повний цикл роботи від навчання моделі нечіткої кластеризації до генерації торгових сигналів з можливістю детального бектестингу, включаючи реалістичне моделювання виконання ордерів з урахуванням Stop-Loss, Take-Profit, комісій та slippage. Система генерує детальні звіти по кожній угоді з візуалізацією кривої капіталу, розподілу прибутків/збитків, аналізу серій вигравів/програвів та інших статистичних

показників. Досягнуті результати підтверджують ефективність використання нечіткої кластеризації для автоматизації прийняття торгових рішень та готовність системи до практичного застосування в реальних умовах торгівлі на валютному ринку.

## 4 ЕКОНОМІЧНИЙ РОЗДІЛ

Для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи, для визначення відповідності вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки.

### 4.1 Проведення комерційного та технологічного аудиту

Метою проведення комерційного і технологічного аудиту дослідження за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями [38] (таблиця 4.1).

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	36	39	38
Середньоарифметична сума балів $CB_c$	37,7		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [38].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» становить 37,7 бала, що, відповідно до [38], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

#### 4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розраховуємо за формулою [39]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом і при

цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$  ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість одночасних користувачів	шт	100	250	2,5	0,3
Середній час обробки запитів	сек	2	1,5	1,33	0,2
Середній час завантаження сторінки	сек	1,5	1	1,5	0,15
Обсяг резервної копії	Гігабайти	2	6	3	0,25
Кількість підтримуваних типів звітів	шт	2	3	1,5	0,1

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 2,5 \cdot 0,3 + 1,33 \cdot 0,2 + 1,5 \cdot 0,15 + 3 \cdot 0,25 + 1,5 \cdot 0,1 = 2,14.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,14 рази.

#### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

### 4.3.1 Витрати на оплату праці

#### Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [38]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=21$  дні.

$$Z_o = 18000,00 \cdot 21 / 21 = 18000,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Проектний менеджер (координатор)	18000,00	857,14	21	18000,00
Інженер-програміст	19200,00	914,29	15	13714,29
Консультант (HR-фахівець)	18000,00	857,14	4	3428,57
Технік 1-ї категорії	8100,00	385,71	18	6942,86
Всього				42085,71

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Система

прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де  $M_M$  – розмір мінімальної місячної заробітної плати, прийmemo  $M_M=8000,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення [18, таблиця Б.2];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 21$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_i = 8000,00 \cdot 1,10 \cdot 1,15 / (21 \cdot 8) = 60,24 \text{ (грн)}.$$

$$Z_{pl} = 60,24 \cdot 8,20 = 493,95 \text{ (грн)}.$$

Величина витрат на основну заробітну плату робітників наведено в таблиці 4.4.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1	2	3	4	5	6
Встановлення обладнання для моделювання та проектування системи прийняття торгових рішень	8,20	2	1,10	60,24	493,95
Підготовка робочого місця розробника програмного забезпечення	4,70	3	1,35	73,93	347,46
Підготовка серверного обладнання	3,20	4	1,50	82,14	262,86
Інсталяція програмного забезпечення для моделювання та розробки системи прийняття торгових рішень	6,22	4	1,50	82,14	510,93
Компіляція програмних блоків	8,00	5	1,70	93,10	744,76
Налагодження програмних блоків	4,60	4	1,50	82,14	377,86
Тестування системи	10,00	2	1,10	60,24	602,38
Всього					3340,20

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.5)$$

де  $H_{\text{доо}}$  – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{доо}} = (42085,71 + 3340,20) \cdot 10 / 100\% = 4542,59 \text{ грн.}$$

#### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{доо}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.6)$$

де  $H_{\text{зн}}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (42085,71 + 3340,20 + 4542,59) \cdot 22 / 100\% = 10993,07 \text{ грн.}$$

#### 4.3.3 Сировина та матеріали

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_{\text{п},j} \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C \quad (4.7)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 3,0 \cdot 215,00 \cdot 1,02 - 0 \cdot 0 = 657,90 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
1	2	3	4	5	6
Офісний папір Cristal A4 500	215,00	3,0	0	0	657,90
Папір для записів Cristal LightPapers 65 A5	100,00	5,0	0	0	510,00
Органайзер офісний Cristal	175,00	2,0	0	0	357,00
Набір офісний Cristal Base	190,00	3,0	0	0	581,40
Картридж для принтера Canon LBP5000	780,00	1,0	0	0	795,60
Диск оптичний Vybir CD-R	23,25	3,0	0	0	71,15
Flesh-пам'ять Kingston 32 GB	169,00	1,0	0	0	172,38
Тека для паперів	89,00	4,0	0	0	363,12
Інші матеріали	200,00	1,0	0	0	204,00
Всього					3712,55

#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні НДР на тему «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_e = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.8)$$

де  $H_j$  – кількість комплектуючих  $j$ -го виду, шт.;

$C_j$  – покупна ціна комплектуючих  $j$ -го виду, грн;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ ).

$$K_e = 1 \cdot 3820,00 \cdot 1,02 = 3896,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
RouterBOARD TP-Link230	1	3820,00	3896,40
База даних (тестувальна) DataBase 64-A10	1	1750,00	1785,00
Всього			5681,40

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i \quad (4.9)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  – кількість найменувань устаткування.

$$B_{спец} = 44599,00 \cdot 1 \cdot 1,02 = 45490,98 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання на основі ЕОМ ASUS AJ13-A71BC	1	44599,00	45490,98
Всього			45490,98

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прог} = \sum_{i=1}^k C_{инрг} \cdot C_{прог.i} \cdot K_i \quad (4.10)$$

де  $C_{инрг}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прог.i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{прог} = 6820,00 \cdot 1 \cdot 1,02 = 6956,40 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	К-сть, шт	Ціна за шт, грн	Вартість
Емулятор серверу для моделювання поведінки системи прийняття торгових рішень	1	6820,00	6956,40
Середовище розробки Pucharm Professional	1	9320,00	9506,40
Платформа .NET Framework 4.8	1	5280,00	5385,60
Абонентна плата доступу	1	352,00	359,04
Всього			22207,44

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{г}} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

де  $Ц_{б}$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{г}$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (32999,00 \cdot 1) / (2 \cdot 12) = 1374,96 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ	32999,00	2	1	1374,96
Персональний комп'ютер розробника аналітично-інформаційних систем	24999,00	2	1	1041,63
Робоче місце інженера-програміста	8700,00	5	1	145,00
Робоче місце аналітика	8700,00	5	1	145,00
Пристрої передачі даних	6800,00	4	1	141,67
Оргтехніка	8100,00	4	1	168,75
Приміщення лабораторії розробки ІС	415000,00	25	1	1383,33
OS Windows 11	6800,00	2	1	283,33
Прикладний пакет Microsoft Office	6700,00	2	1	279,17
Всього				4962,83

#### 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{ени}}{\eta_i}, \quad (4.12)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; прийmemo  $C_e = 10,98$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,42 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 737,86 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.10

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ	0,42	160,0	737,86
Персональний комп'ютер розробника аналітично-інформаційних систем	0,32	160,0	562,18
Робоче місце інженера-програміста	0,12	120,0	158,11
Робоче місце аналітика	0,10	20,0	21,96
Пристрої передачі даних	0,05	120,0	65,88
Оргтехніка	0,45	1,3	6,18
Серверне обладнання на основі EOM ASUS A13-A71BC	0,10	120,0	131,76
RouterBOARD TP-Link230	0,02	160,0	35,14
Всього			1719,06

#### 4.3.9 Службові відрядження

Витрати за статтею «Службові відрядження» відсутні.

#### 4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.13)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 30\%$ .

$$B_{cn} = (42085,71 + 3340,20) \cdot 30 / 100\% = 13627,78 \text{ грн.}$$

#### 4.3.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_g = (Z_o + Z_p) \cdot \frac{H_{ig}}{100\%}, \quad (4.14)$$

де  $H_{ig}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{ig} = 50\%$ .

$$I_g = (42085,71 + 3340,20) \cdot 50 / 100\% = 22712,96 \text{ грн.}$$

## 4.3.12 Накладні (загально виробничі) витрати

Витрати за статтею «Накладні (загально виробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.15)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загально виробничі) витрати», прийmemo  $H_{нзв} = 100\%$ .

$$B_{нзв} = (42085,71 + 3340,20) \cdot 100 / 100\% = 45425,92 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сн} + I_v + B_{нзв}. \quad (4.16)$$

$$\begin{aligned} B_{заг} = & 42085,71 + 3340,20 + 4542,59 + 10993,07 + 3712,55 + 5681,40 + 45490,98 + \\ & 22207,44 + 4962,83 + 1719,06 + 0,00 + 13627,78 + 22712,96 + 45425,92 = \\ & 226502,48 \text{ грн.} \end{aligned}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.17)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta = 0,9$ .

$$ЗВ = 226502,48 / 0,9 = 251669,43 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Результати дослідження проведені за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

$\Delta N$  – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються (таблиця 4.11).

Таблиця 4.11 – Показники збільшення кількості споживачів

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	550	900	1500	1300

$N$  – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 75000 осіб;

$Ц_б$  – вартість послуги у році до впровадження інформаційної системи, прийmemo 70,00 грн;

$\pm\Delta C_o$  – зміна вартості послуги від впровадження результатів, прийmemo 29,87 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [18]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.18)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 42\%$ ;

$\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році  $\vartheta = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (29,87 \cdot 75000,00 + 99,87 \cdot 550) \cdot 0,83 \cdot 0,42 \cdot (1 - 0,18/100\%) = 656081,36 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (29,87 \cdot 75000,00 + 99,87 \cdot 1450) \cdot 0,83 \cdot 0,42 \cdot (1 - 0,18/100\%) = 681774,60 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (29,87 \cdot 75000,00 + 99,87 \cdot 2950) \cdot 0,83 \cdot 0,42 \cdot (1 - 0,18/100\%) = 724596,66 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (29,87 \cdot 75000,00 + 99,87 \cdot 4250) \cdot 0,83 \cdot 0,42 \cdot (1 - 0,18/100\%) = 761709,11 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків  $ПП$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.19)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,25$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 656081,36/(1+0,25)^1 + 681774,60/(1+0,25)^2 + 724596,66/(1+0,25)^3 + \\ &+ 761709,11/(1+0,25)^4 = 524865,09 + 436335,74 + 370993,49 + 311996,05 = \\ &= 1644190,38 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$BB = k_{inv} \cdot ZB, \quad (4.20)$$

де  $k_{inv}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{inv} = 2$ ;

$ZB$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 251669,43 грн.

$$BB = k_{inv} \cdot ZB = 2 \cdot 251669,43 = 503338,86 \text{ грн.}$$

Абсолютний економічний ефект  $E_{abc}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV \quad (4.21)$$

де  $III$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1644190,38 грн;

$PV$  – теперішня вартість початкових інвестицій, 503338,86 грн.

$$E_{abc} = III - PV = 1644190,38 - 503338,86 = 1140851,52 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_e$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_{\varepsilon} = T_{жс} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.22)$$

де  $E_{абс}$  – абсолютний економічний ефект вкладених інвестицій, 1140851,52 грн;

$PV$  – теперішня вартість початкових інвестицій, 503338,86 грн;

$T_{жс}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_{\varepsilon} = T_{жс} \sqrt{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 1140851,52/503338,86)^{1/4} - 1 = 0,34.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{мін}$ :

$$\tau_{мін} = d + f, \quad (4.23)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні  $d = 0,11$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,15.

$\tau_{мін} = 0,11 + 0,15 = 0,26 < 0,34$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_{\varepsilon}$ , вища мінімальної внутрішньої дохідності. Тобто

інвестувати в науково-дослідну роботу за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_v}, \quad (4.24)$$

де  $E_v$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,34 = 2,90 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### 4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів» становить 37,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,14 рази.

Також термін окупності становить 2,90 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може

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

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів».

## ВИСНОВКИ

У магістерській кваліфікаційній роботі вирішено актуальну науково-прикладну задачу розробки автоматизованої системи прийняття торгових рішень на валютному ринку Форекс на основі методів нечіткої кластеризації. Створена система забезпечує ідентифікацію оптимальних точок входу та виходу з ринку через інтеграцію алгоритму Fuzzy C-Means з багатофільтровою архітектурою та технічними індикаторами, що дозволяє підвищити стабільність торгівлі та зменшити кількість хибних сигналів.

У першому розділі проведено аналіз предметної області автоматизованих торгових систем, досліджено характеристики фінансових часових рядів та існуючі методи ідентифікації торгових рішень. Встановлено, що валютний ринок Форекс з денним обсягом понад 7.5 трильйонів доларів характеризується високою волатильністю, нестационарністю та низьким відношенням сигнал/шум, що робить автоматизацію торгових рішень критично важливою для досягнення стабільної прибутковості. Розглянуто властивості фінансових часових рядів: нелінійність, кластеризація волатильності, важкі хвости розподілу, наявність короткострокових та довгострокових залежностей. Аналіз існуючих методів (технічний аналіз, фундаментальний аналіз, машинне навчання) виявив їхні обмеження: технічні індикатори генерують багато хибних сигналів у флетовому ринку (ефективність 55-60%), методи глибокого навчання схильні до перенавчання та потребують великих обсягів даних. Обґрунтовано вибір алгоритму Fuzzy C-Means як основи системи через його здатність працювати з невизначеністю, інтерпретовність результатів та меншу схильність до перенавчання порівняно зі складними ML-моделями.

У другому розділі розроблено математичне та алгоритмічне забезпечення системи прийняття торгових рішень на основі нечіткої кластеризації. Формалізовано задачу ідентифікації торгових сигналів як задачу нечіткої кластеризації в багатомірному просторі технічних індикаторів (RSI, ATR,

MACD, Stochastic). Адаптовано алгоритм Fuzzy C-Means для кластеризації ринкових станів на два кластери (умовно "BUY" та "SELL") з параметром розмитості  $m=2.0$ , що забезпечує оптимальний баланс між чіткістю класифікації та стійкістю до шуму. Розроблено комплексну систему валідації сигналів, що включає п'ять типів фільтрів: гістерезисний фільтр з різними порогами для входу (0.8) та виходу (0.45), фільтр мінімальної тривалості позиції (50 свічок = 12.5 годин для M15), ATR-фільтр для торгівлі при достатній волатильності, трендовий фільтр на основі старших таймфреймів (H4/D1) для торгівлі за глобальним трендом, сесійний фільтр для обмеження торгівлі активними сесіями (London, New York).

У третьому розділі реалізовано програмне забезпечення системи з використанням мови Python 3.8+ та сучасних бібліотек для наукових обчислень (scikit-fuzzy, pandas, NumPy, scikit-learn). Розроблено модульну архітектуру системи, та здійснено її опис. Забезпечено інтеграцію з торговою платформою MetaTrader 5 через мову MQL5 для можливості тестування на реальних ринкових даних. Проведено тестування на історичних даних валютних пар EUR/USD та GBP/USD за період з 01.01.2024 до 01.06.2025 (35,137 барів таймфрейму M15) з розділенням на тренувальну (80%) та тестову (20%) вибірки. Експериментальні дослідження виявили оптимальні параметри системи: використання трендового фільтру D1 підвищило прибутковість (з +2.45% до +3.15% для EUR/USD), мінімальна тривалість утримання позиції 50 свічок забезпечила найкращу стабільність результатів, а комбінація фільтрації за торговими сесіями показала різні оптимуми для різних валютних пар. Порівняння з базовими стратегіями (Buy and Hold та MA Crossover) на парі GBP/USD продемонструвало, що запропонована система показує найбільш збалансовані результати: Total Return 7.96%, Win Rate 55.79%, Sharpe Ratio 4.11 при максимальній просадці -1.67%, що значно перевершує базові стратегії за комплексом показників ризик-прибутковість.

У четвертому розділі проведено економічне обґрунтування доцільності розробки та впровадження системи. Виконано оцінку ринку автоматизованих

торгових систем, що демонструє стабільне зростання попиту на алгоритмічну торгівлю з боку як роздрібних, так і інституційних інвесторів. Розраховано економічні показники проекту, що підтверджують його комерційну привабливість та потенціал для практичного застосування.

Науково-технічний результат роботи полягає у розробці комбінованого підходу, що поєднує нечітку кластеризацію з багатофільтровою архітектурою для автоматизації прийняття торгових рішень на валютному ринку, що забезпечує вищу стабільність результатів порівняно з традиційними методами технічного аналізу та меншу схильність до перенавчання порівняно з методами глибокого навчання.

Практична цінність роботи підтверджується створенням працездатної системи з відкритим вихідним кодом, детальною документацією та можливістю інтеграції з MetaTrader 5, що дозволяє використовувати систему для реальної торгівлі на валютному ринку Форекс, навчання та подальшого дослідження методів автоматизації торгових рішень.

Напрямки подальших досліджень включають розширення системи для торгівлі на інших фінансових ринках (фондовий ринок, криптовалюти, товарні ринки), адаптацію алгоритму для більшої кількості кластерів для детальнішої класифікації ринкових станів, інтеграцію з методами глибокого навчання для автоматичного виявлення оптимальних комбінацій індикаторів, розробку адаптивних механізмів автоматичного налаштування параметрів фільтрів залежно від поточних ринкових умов, дослідження ефективності системи на інших валютних парах та таймфреймах.

Всі поставлені задачі магістерської дипломної роботи виконано в повному обсязі. Створена система забезпечує автоматизацію прийняття торгових рішень на валютному ринку Форекс, демонструє стабільні результати на тестових даних та готова до практичного використання.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Павленко В. В., Маслій Р. В., Кабачій В.В. Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів // Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)» URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2026/index> (дата звернення 05.12.2025).
2. Hu Z. Accuracy of Financial Time Series Forecasting Models / Z. Hu та ін. // Journal of Financial Modeling. – 2020. – Vol. 15, № 3. – P. 45–60.
3. Murphy J. J. Technical Analysis of the Financial Markets / J. J. Murphy. – New York : New York Institute of Finance, 1999. – 542 p.
4. Бакай Є. І. Модель прийняття рішень для фінансових часових рядів на основі пари середніх з використанням оцінки різних часових вимірів / Є. І. Бакай, В. В. Кабачий, Р. В. Маслій // Вісник Вінницького політехнічного інституту. – 2017. – № 1. – С. 70–77.
5. Форекс [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – Режим доступу: <https://uk.wikipedia.org/wiki/Форекс> (дата звернення: 25.11.2025). – Назва з екрана.
6. MetaTrader 4 [Електронний ресурс] // MetaTrader. – Режим доступу: <https://www.metatrader4.com/en> (дата звернення: 25.11.2025). – Назва з екрана.
7. cTrader [Електронний ресурс] // cTrader. – Режим доступу: <https://ctrader.com/> (дата звернення: 25.11.2025). – Назва з екрана.
8. Schmitt T. A. Non-stationarity in financial time series: Generic features and tail behavior / T. A. Schmitt, D. Chetalova, R. Schäfer, T. Guhr // Europhysics Letters. – 2013. – Vol. 103, № 5. – Art. 58003. – 6 p.
9. Rabhi F. A. State-of-the-art in applying machine learning to electronic trading / F. A. Rabhi, N. Mehandjiev, A. Baghdadi // Enterprise applications, markets

and services in the finance industry / eds. B. Clapham, J. A. Koch. – Cham : Springer, 2020. – P. 3–20.

10. Chen W. A CEP-driven framework for real-time news impact prediction on financial markets / W. Chen, A. El Majzoub, I. Al-Qudah, F. A. Rabhi // *Service Oriented Computing and Applications*. – 2023. – Vol. 17, № 2. – P. 129–144.

11. Aloui C. Modelling and forecasting value at risk and expected shortfall for GCC stock markets: Do long memory, structural breaks, asymmetry, and fat-tails matter? / C. Aloui, H. B. Hamida // *The North American Journal of Economics and Finance*. – 2014. – Vol. 29. – P. 349–380.

12. Takahashi S. Modeling financial time-series with generative adversarial networks / S. Takahashi, Y. Chen, K. Tanaka-Ishii // *Physica A: Statistical Mechanics and its Applications*. – 2019. – Vol. 527. – Art. 121261. – 14 p.

13. D'Urso P. Clustering of financial time series / P. D'Urso, C. Cappelli, D. di Lallo, R. Massari // *Physica A: Statistical Mechanics and its Applications*. – 2013. – Vol. 392, № 9. – P. 2114–2129.

14. Buonocore R. J. Measuring multiscaling in financial time-series / R. J. Buonocore, T. Aste, T. Di Matteo // *Chaos, Solitons & Fractals*. – 2016. – Vol. 88. – P. 38–47.

15. Zhang N. Detrended moving average partial cross-correlation analysis on financial time series / N. Zhang, A. Lin, P. Yang // *Physica A: Statistical Mechanics and its Applications*. – 2020. – Vol. 542. – Art. 122960. – 11 p.

16. Li Q. A multimodal event-driven LSTM model for stock prediction using online news / Q. Li, J. Tan, J. Wang, H. Chen // *IEEE Transactions on Knowledge and Data Engineering*. – 2021. – Vol. 33, № 10. – P. 3323–3337.

17. Ballings M. Evaluating Multiple Classifiers for Stock Price Direction Prediction / M. Ballings, D. Van den Poel, N. Hespeels, R. Gryp // *Expert Systems with Applications*. – 2015. – Vol. 42, № 20. – P. 7046–7056.

18. Breiman L. Random Forests / L. Breiman // *Machine Learning*. – 2001. – Vol. 45, № 1. – P. 5–32.

19. Huang W. Forecasting Stock Market Movement Direction with Support Vector Machine / W. Huang, Y. Nakamori, S.-Y. Wang // *Computers & Operations Research*. – 2005. – Vol. 32, № 10. – P. 2513–2522.
20. Fischer T. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions / T. Fischer, C. Krauss // *European Journal of Operational Research*. – 2018. – Vol. 270, № 2. – P. 654–669.
21. Sezer O. B. Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach / O. B. Sezer, A. M. Ozbayoglu // *Applied Soft Computing*. – 2018. – Vol. 70. – P. 525–538.
22. Deng Y. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading / Y. Deng, F. Bao, Y. Kong, Z. Ren, Q. Dai // *IEEE Transactions on Neural Networks and Learning Systems*. – 2017. – Vol. 28, № 3. – P. 653–664.
23. Sezer O. B. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019 / O. B. Sezer, M. U. Gudelek, A. M. Ozbayoglu // *Applied Soft Computing*. – 2020. – Vol. 90. – Article 106181.
24. Zadeh L. A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes / L. A. Zadeh // *IEEE Transactions on Systems, Man, and Cybernetics*. – 1973. – Vol. SMC-3, № 1. – P. 28–44.
25. Yager R. R. Essentials of Fuzzy Modeling and Control / R. R. Yager, D. P. Filev. – New York : John Wiley & Sons, 1994. – 408 p.
26. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms / J. C. Bezdek. – New York : Plenum Press, 1981. – 272 p.
27. Lai K. K. A Double-Stage Genetic Optimization Algorithm for Portfolio Selection / K. K. Lai, L. Yu, S. Wang, C. Zhou // *Lecture Notes in Computer Science*. – 2006. – Vol. 3994. – P. 928–937.
28. Boyacioglu M. A. An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the Prediction of Stock Market Return: The Case of the Istanbul Stock Exchange / M. A. Boyacioglu, D. Avci // *Expert Systems with Applications*. – 2010. – Vol. 37, № 12. – P. 7908–7912.

29. Esfahanipour A. Adapted Neuro-Fuzzy Inference System on Indirect Approach TSK Fuzzy Rule Base for Stock Market Analysis / A. Esfahanipour, W. Aghamiri // *Expert Systems with Applications*. – 2010. – Vol. 37, № 7. – P. 4742–4748.
30. Schumaker R. P. Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System / R. P. Schumaker, H. Chen // *ACM Transactions on Information Systems*. – 2009. – Vol. 27, № 2. – Article 12.
31. Cervelló-Royo R. Stock Market Trading Rule Based on Pattern Recognition and Technical Analysis: Forecasting the DJIA Index with Intraday Data / R. Cervelló-Royo, F. Guijarro, K. Michniuk // *Expert Systems with Applications*. – 2015. – Vol. 42, № 14. – P. 5963–5975.
32. Dourra H. Investment Using Technical Analysis and Fuzzy Logic / H. Dourra, P. Siy // *Fuzzy Sets and Systems*. – 2002. – Vol. 127, № 2. – P. 221–240.
33. Esfahanipour A. Adapted Neuro-Fuzzy Inference System on Indirect Approach TSK Fuzzy Rule Base for Stock Market Analysis / A. Esfahanipour, W. Aghamiri // *Expert Systems with Applications*. – 2010. – Vol. 37, № 7. – P. 4742–4748.
34. Krauss C. Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500 / C. Krauss, X. A. Do, N. Huck // *European Journal of Operational Research*. – 2017. – Vol. 259, № 2. – P. 689–702.
35. Fischer T. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions / T. Fischer, C. Krauss // *European Journal of Operational Research*. – 2018. – Vol. 270, № 2. – P. 654–669.
36. scikit-fuzzy: A Python Library for Fuzzy Logic [Электронный ресурс] / J. Warner, J. Sexauer, scikit-fuzzy Development Team. – 2019. – Режим доступа: <https://github.com/scikit-fuzzy/scikit-fuzzy> (дата звернення: 29.11.2025). – Назва з екрана.
37. Kabachii V. Identifying moments of decision making on trade in financial time series using fuzzy cluster analysis / V. Kabachii, R. Maslii, S.

Kozlovskiy, O. Dronchack // *Neuro-Fuzzy Modeling Techniques in Economics*. – 2023. – № 12. – P. 175–192. – DOI: 10.33111/nfmte.2023.175.

38. Козловський В. О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с.

39. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

**ДОДАТКИ**

Додаток А (обов'язковий)

Технічне завдання

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

«17» жовтня 2025 року

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

**«СИСТЕМА ПРИЙНЯТТЯ ТОРГОВИХ РІШЕНЬ НА ФІНАНСОВИХ РИНКАХ  
НА ОСНОВІ НЕЧІТКОГО КЛАСТЕРНОГО АНАЛІЗУ ЧАСОВИХ РЯДІВ»**

08-31.МКР.006.02.000 ТЗ

Керівник роботи:

к.т.н., доц. каф. АІТ

Роман МАСЛІЙ

«16» жовтня 2025 р.

Виконавець:

ст. гр. ІСТ-24м

Володимир ПАВЛЕНКО

«16» жовтня 2025 р.

## Вінниця ВНТУ – 2025

### 1. Назва та галузь застосування

Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів.

Інформаційні системи та технології. Алгоритмічна торгівля на фінансових ринках. Застосування методів машинного навчання для аналізу фінансових часових рядів.

### 2. Підстава для розробки

Розробку системи здійснювати на підставі наказу по університету № 313 від 24 вересня 2025 року та завдання до магістерської кваліфікаційної роботи, складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій»

### 3. Мета та призначення розробки

Метою роботи є розробка автоматизованої торговельної системи для валютного ринку Forex, яка забезпечить:

- Автоматичний аналіз ринкових даних з використанням алгоритму Fuzzy C-Means для виявлення латентних закономірностей у 6-вимірному просторі технічних індикаторів;
- Генерацію торгових сигналів (BUY/SELL) на основі розрахунку ступенів належності поточного ринкового стану до кластерів з використанням гістерезисного фільтра для зменшення шуму;
- Багатофільтрову валідацію сигналів (ATR Filter, Trend Filter, Session Filter, Min Position Duration) для зменшення overtrading та підвищення якості угод;
- Інтеграцію з MetaTrader 5 через TCP/IP протокол для автоматичного виконання угод у реальному часі з підтримкою stop-loss та take-profit механізмів;
- Комплексне backtesting на історичних даних з розрахунком ключових метрик ефективності (Total Return, Sharpe Ratio, Max Drawdown, Win Rate, Profit Factor) та порівняння з базовими стратегіями.

Призначення системи: забезпечення прибуткової автоматизованої торгівлі на валютному ринку Forex з мінімізацією емоційного впливу на прийняття

рішень та можливістю адаптації до змінних ринкових умов через періодичне перенавчання моделі.

#### 4. Джерела розробки

1. Murphy J. J. Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York: New York Institute of Finance, 1999. 576 p.
2. Kabachii V. Identifying moments of decision making on trade in financial time series using fuzzy cluster analysis / V. Kabachii, R. Maslii, S. Kozlovskiy, O. Dronchack // Neuro-Fuzzy Modeling Techniques in Economics. – 2023. – № 12. – P. 175–192. – DOI: 10.33111/nfnte.2023.175.
3. MetaQuotes Software Corp. MQL5 Reference. URL: <https://www.mql5.com/en/docs> (дата звернення: 01.10.2025)
4. Scikit-learn Developers. Scikit-learn: Machine Learning in Python. URL: <https://scikit-learn.org/stable/documentation.html> (дата звернення: 01.10.2025)
5. Scikit-fuzzy Developers. Scikit-fuzzy Documentation. URL: <https://pythonhosted.org/scikit-fuzzy/> (дата звернення: 01.10.2025)

#### 5. Показники призначення

##### 5.1. Основні технічні характеристики системи

Функціональні можливості:

##### 5.1.1 Модуль навчання FCM моделі:

- Завантаження історичних OHLCV даних з CSV файлів;
- Розрахунок 6 технічних індикаторів (RSI, ATR, Price Change, MACD Crossover, MA Crossover, Stochastic %K);
- Нормалізація даних методом StandardScaler (Z-score);
- Навчання Fuzzy C-Means з параметрами:  $c=2$  кластери,  $m=2.0$  (fuzziness), 10 рестартів для уникнення локальних мінімумів;
- Інтерпретація центроїдів для визначення кластерів BUY та SELL;
- Збереження навченої моделі (центроїди та scaler) у форматі pickle.

##### 5.1.2 Модуль генерації сигналів:

- Розрахунок ступенів належності для нових даних:  $u\{BUY\}$ ,  $u\{SELL\}$ ;

- Гістерезисний фільтр з подвійними порогами ( $\theta_{entry}=0.8$ ,  $\theta_{exit}=0.45$ );
- Виведення торгових сигналів: +1 (BUY), -1 (SELL), 0 (HOLD).

#### 5.1.3 Багатофільтрова система валідації:

- Hysteresis Filter: різні пороги для входу та виходу з позиції;
- Min Position Duration: мінімальна тривалість утримання позиції (50 свічок = 12.5 годин для M15);
- ATR Filter: торгувати тільки при достатній волатильності ( $ATR > 0.0004$ );
- Trend Filter: торгувати тільки в напрямку тренду вищого таймфрейму (D1);
- Session Filter: торгувати тільки під час активних торгових сесій (London, New York).

#### 5.1.4 Модуль backtesting:

Імітація виконання угод на історичних даних з урахуванням:

- Комісій ( $0.0001\% = 1 \text{ pip}$ );
- Stop-loss та Take-profit рівнів;
- Position sizing (95% від доступного капіталу);

Розрахунок метрик ефективності:

- Total Return (%);
- Max Drawdown (%), Drawdown Duration (днів);
- Sharpe Ratio, Sortino Ratio;
- Win Rate (%), Profit Factor, Average Win/Loss;
- Total Trades, Long/Short розподіл.

#### 5.1.5 Модуль візуалізації:

- Графік equity curve (зміна капіталу в часі);
- Графік drawdown (просідання від максимального рівня);
- Розподіл PnL (Profit and Loss) для закритих позицій;
- Розподіл тривалості угод (trade duration);
- Підсумкова таблиця метрик.

### 5.2. Мінімальні системні вимоги

#### 5.2.1 Апаратне забезпечення:

- Процесор: тактова частота не менше 2.5 GHz, рекомендовано 4+ ядра для швидкого навчання FCM;
- Оперативна пам'ять: 8 GB RAM (мінімум 4 GB для невеликих датасетів);
- Місце на диску: 2 GB для програмного забезпечення та бібліотек; 5-10 GB для історичних даних (залежить від кількості валютних пар та періоду);

- Мережа: стабільне Інтернет-з'єднання для завантаження даних та підключення до MT5 (мінімум 10 Mbps).

#### 5.2.2 Програмне забезпечення:

- Операційна система: Windows 10/11, Linux (Ubuntu 20.04+), macOS 10.15+;
- Python: версія 3.8 або вище;
- MetaTrader 5 (опціонально): для торгівлі в реальному часі.

#### 5.3. Вхідні дані

##### Історичні OHLCV дані:

- Формат: CSV файли з колонками: timestamp, open, high, low, close, volume;
- Валютні пари: EURUSD, GBPUSD (можливість додавання інших пар);
- Таймфрейм: M15 (15 хвилин) – основний, з можливістю використання H1, H4, D1;
- Період: мінімум 1 рік для навчання, рекомендовано 2-3 роки;
- Джерела: MetaTrader 5, Dukascopy, FXCM, Yahoo Finance.

#### 5.4 Результати роботи програми

##### 5.4.1 Навчена FCM модель:

- Файли: fuzzycentersfiltered.pkl (центроїди), fuzzyscalerfiltered.pkl (StandardScaler параметри);
- Інтерпретація центроїдів: визначення кластерів BUY та SELL на основі компоненти Price\_Change.

##### 5.4.2 Backtest звіт:

Текстовий звіт у консолі з детальною статистикою:

- Initial Capital, Final Equity, Total Return (%);
- Total Trades, Closed Positions, Win Rate (%);
- Max Drawdown (%), Drawdown Duration (днів);
- Sharpe Ratio, Sortino Ratio, Calmar Ratio;
- Profit Factor, Average Win, Average Loss;
- Longest Winning/Losing Streak.

Графічний звіт (PNG файли):

- 6 графіків (Price + Signals, Equity, Drawdown, PnL Distribution, Trade Duration, Frequency Analysis).

## 6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 250 тис. грн.

- узагальнений коефіцієнт якості розробки – більше 2-х
- термін окупності – до 3х років

#### 7. Стадії розробки

1. Розділ 1 «Дослідження предметної області та існуючих методів» має бути виконаний до 05.10.2025 р.
2. Розділ 2 «Розробка математичного та алгоритмічного забезпечення системи прийняття торгових рішень» має бути виконаний до 25.10.2025 р.
3. Розділ 3 «Розробка програмного забезпечення системи прийняття торгових рішень» має бути виконаний до 20.11.2025 р.
4. Розділ 4 «Економічний розділ» має бути виконаний до 01.12.2025 р.

#### 8. Порядок контролю та приймання

1. Рубіжний контроль провести до 14.11.2025.
2. Попередній захист магістерської кваліфікаційної роботи провести до 02.12.2025.
3. Захист магістерської кваліфікаційної роботи провести в період з 15.12.2025 р. до 19.12.2025 р.

## Додаток Б (обов'язковий)

## Ілюстративна частина

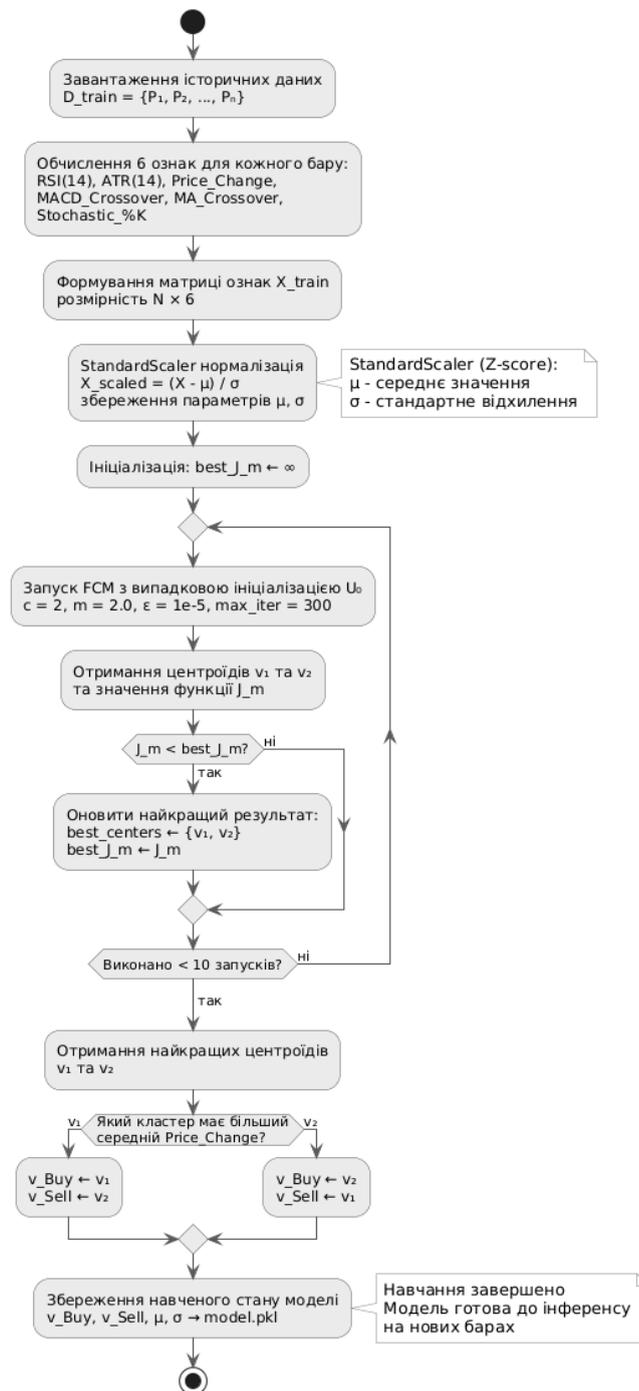


Рисунок Б.1 – UML-діаграма діяльності етапу офлайн-навчання



Рисунок Б.2 – UML-діаграма діяльності етапу онлайн-генерації сигналів

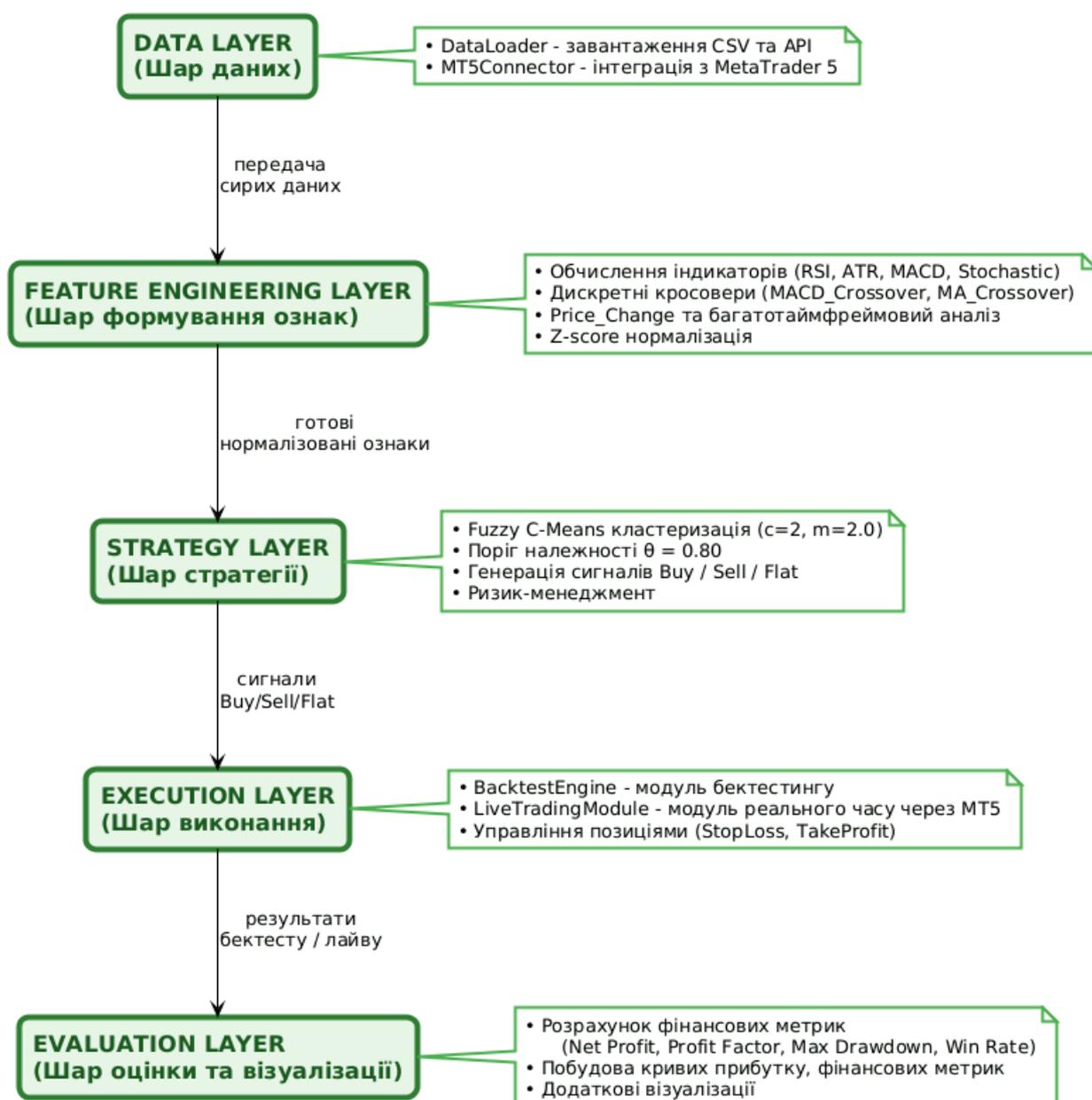


Рисунок Б.3 – Загальна архітектура системи

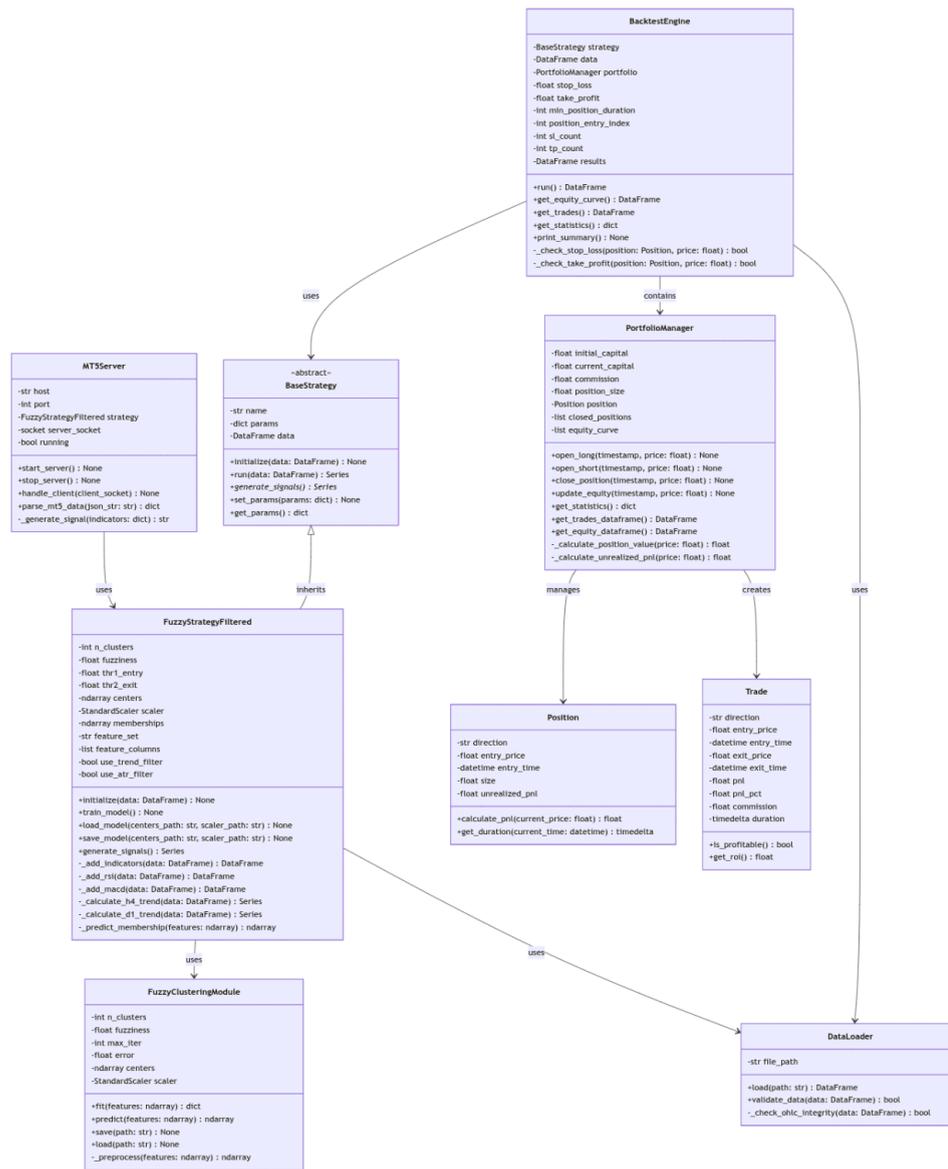


Рисунок Б.4 – UML-діаграма класів

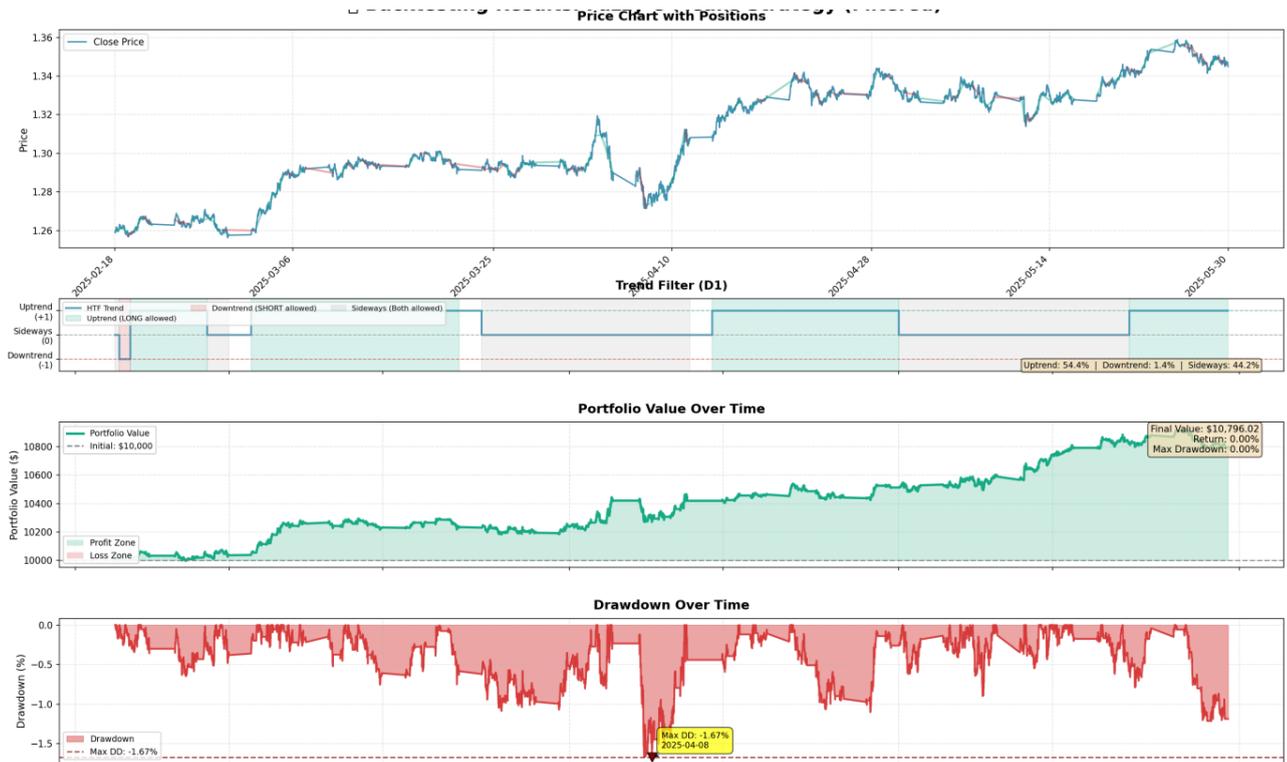


Рисунок Б.5 – Візуалізація результату за тестовий період запропонованої стратегії на валютній парі GBD/USD

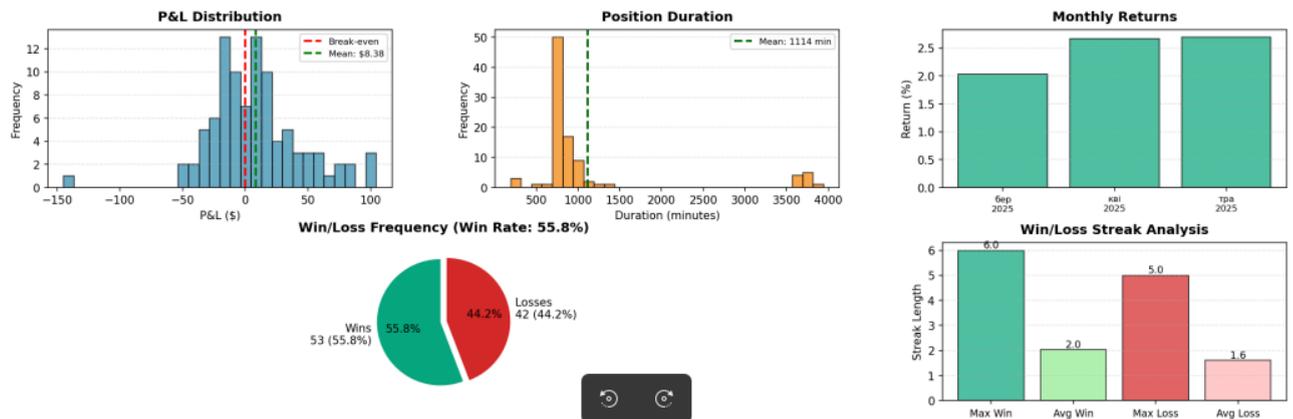


Рисунок Б.6 – Візуалізація додаткових статистичних показників торгівлі за тестовий період запропонованої стратегії на валютній парі GBD/USD

Додаток В (обов'язковий)  
Лістинг програмного забезпечення

*Фрагмент коду движка для бектестингу (BacktestEngine)*

```
import pandas as pd
from typing import Optional
from ..strategy.base_strategy import BaseStrategy, Signal
from ..portfolio.portfolio_manager import PortfolioManager

class BacktestEngine:
    """Движок для запуску backtesting на історичних даних"""

    def __init__(self, strategy: BaseStrategy,
                 data: pd.DataFrame,
                 initial_capital: float = 10000.0,
                 commission: float = 0.0001,
                 position_size: float = 0.95,
                 stop_loss: Optional[float] = None,
                 take_profit: Optional[float] = None,
                 min_position_duration: int = 0):
        """
        Args:
            strategy: Торгова стратегія
            data: DataFrame з OHLCV даними
            initial_capital: Початковий капітал
            commission: Комісія за угоду
            position_size: Частка капіталу для позиції (0.0-1.0, за замовчуванням 0.95 = 95%)
            stop_loss: Stop-loss в % (наприклад 0.02 = 2% збитку)
            take_profit: Take-profit в % (наприклад 0.05 = 5% прибутку)
            min_position_duration: Мінімальна тривалість позиції в свічках (0 = без обмеження)
        """
        self.strategy = strategy
```



```

for indicator in indicators_to_copy:
    if indicator in self.strategy.data.columns:
        data_with_signals[indicator] = self.strategy.data[indicator]

# Проходимо по кожній свічці
previous_signal = Signal.HOLD
current_candle_index = 0 # Лічильник свічок (не timestamp!)

for idx, row in data_with_signals.iterrows():
    current_price = row['close']
    current_signal = row['signal']

    # Оновлюємо equity curve
    self.portfolio.update_equity(idx, current_price)

    # Перевірка мінімальної тривалості позиції
    position_duration = 0
    if self.portfolio.position is not None and self.position_entry_index is not None:
        position_duration = current_candle_index - self.position_entry_index

    # Чи можна закрити позицію (якщо пройшла мінімальна тривалість)
    can_close_position = (
        self.portfolio.position is None or
        self.min_position_duration == 0 or
        position_duration >= self.min_position_duration
    )

    # Перевіряємо Stop-Loss та Take-Profit (ЗАВЖДИ, незалежно від мінімальної
    тривалості!)
    # SL/TP - це захисні механізми, вони мають працювати завжди
    if self.portfolio.position is not None:
        entry_price = self.portfolio.position.entry_price

        if self.portfolio.position.direction == 'LONG':
            # Для LONG позиції

```

```

pnl_pct = (current_price - entry_price) / entry_price

# Stop-Loss (збиток)
if self.stop_loss and pnl_pct <= -self.stop_loss:
    self.portfolio.close_position(idx, current_price)
    self.position_entry_index = None
    self.sl_count += 1 # Лічильник SL
    previous_signal = Signal.HOLD
    current_candle_index += 1
    continue

# Take-Profit (прибуток)
if self.take_profit and pnl_pct >= self.take_profit:
    self.portfolio.close_position(idx, current_price)
    self.position_entry_index = None
    self.tp_count += 1 # Лічильник TP
    previous_signal = Signal.HOLD
    current_candle_index += 1
    continue

elif self.portfolio.position.direction == 'SHORT':
    # Для SHORT позиції
    pnl_pct = (entry_price - current_price) / entry_price

    # Stop-Loss (збиток)
    if self.stop_loss and pnl_pct <= -self.stop_loss:
        self.portfolio.close_position(idx, current_price)
        self.position_entry_index = None
        self.sl_count += 1 # Лічильник SL
        previous_signal = Signal.HOLD
        current_candle_index += 1
        continue

    # Take-Profit (прибуток)
    if self.take_profit and pnl_pct >= self.take_profit:

```

```

self.portfolio.close_position(idx, current_price)
self.position_entry_index = None
self.tp_count += 1 # Лічильник TP
previous_signal = Signal.HOLD
current_candle_index += 1
continue

# Логіка торгівлі на основі сигналів
if current_signal == Signal.BUY and previous_signal != Signal.BUY:
    # Закриваємо short позицію якщо є (і якщо можна)
    if self.portfolio.position and self.portfolio.position.direction == 'SHORT':
        if can_close_position:
            self.portfolio.close_position(idx, current_price)
            self.position_entry_index = None
        else:
            # Не можемо закрити - пропускаємо сигнал
            current_candle_index += 1
            continue
    # Відкриваємо long позицію
    if self.portfolio.position is None:
        self.portfolio.open_long(idx, current_price)
        self.position_entry_index = current_candle_index

elif current_signal == Signal.SELL and previous_signal != Signal.SELL:
    # Закриваємо long позицію якщо є (і якщо можна)
    if self.portfolio.position and self.portfolio.position.direction == 'LONG':
        if can_close_position:
            self.portfolio.close_position(idx, current_price)
            self.position_entry_index = None
        else:
            # Не можемо закрити - пропускаємо сигнал
            current_candle_index += 1
            continue
    # Відкриваємо short позицію
    if self.portfolio.position is None:

```

```

self.portfolio.open_short(idx, current_price)
self.position_entry_index = current_candle_index

elif current_signal == Signal.HOLD:
    # При HOLD закриваємо будь-яку відкриту позицію (ТІЛЬКИ якщо можна)
    if self.portfolio.position is not None and can_close_position:
        self.portfolio.close_position(idx, current_price)
        self.position_entry_index = None

    previous_signal = current_signal
    current_candle_index += 1

# Закриваємо позицію в кінці якщо вона ще відкрита
if self.portfolio.position is not None:
    last_price = data_with_signals.iloc[-1]['close']
    last_time = data_with_signals.index[-1]
    self.portfolio.close_position(last_time, last_price)

# Збираємо результати
self.results = data_with_signals
return self.results

def get_equity_curve(self) -> pd.DataFrame:
    """
    Повертає криву equity

    Returns:
        DataFrame з equity curve
    """
    return self.portfolio.get_equity_dataframe()

def get_trades(self) -> pd.DataFrame:
    """
    Повертає всі угоди

```

Returns:

DataFrame з угодами

"""

```
return self.portfolio.get_trades_dataframe()
```

def get\_statistics(self) -> dict:

"""

Повертає статистику backtesting

Returns:

Словник зі статистикою

"""

```
stats = self.portfolio.get_statistics()
```

```
# Додаємо лічильники SL/TP
```

```
stats['sl_exits'] = self.sl_count
```

```
stats['tp_exits'] = self.tp_count
```

```
return stats
```

def print\_summary(self) -> None:

"""Виводить підсумкову інформацію про backtesting"""

```
stats = self.get_statistics()
```

```
print("=" * 60)
```

```
print(f'Backtesting Results: {self.strategy.name}')
```

```
print("=" * 60)
```

```
print(f'Початковий капітал:   ${stats['initial_capital']:,.2f}')
```

```
print(f'Фінальний капітал:   ${stats['final_equity']:,.2f}')
```

```
print(f'Загальна прибутковість: {stats['total_return_%']:,.2f}%')
```

```
print(f'Кількість угод:       {stats['total_trades']}')
```

```
print(f'Загальна комісія:      ${stats['total_commission']:,.4f}')
```

```
print(f'Відкрита позиція:     {'Так' if stats['has_open_position'] else 'Hi'})')
```

```
print("=" * 60)
```

*Фрагмент коду модуля завантаження та обробки історичних даних  
(DataLoader)*

```
import pandas as pd
from pathlib import Path
from typing import Optional
from datetime import datetime

class DataLoader:
    """Клас для завантаження даних з CSV файлів MetaTrader"""

    def __init__(self, filepath: str):
        """
        Args:
            filepath: Шлях до CSV файлу з історичними даними
        """
        self.filepath = Path(filepath)
        self.data: Optional[pd.DataFrame] = None

    def load(self) -> pd.DataFrame:
        """
        Завантажує дані з CSV файлу та обробляє їх

        Returns:
            DataFrame з OHLCV даними та індексом datetime
        """
        # Читаємо CSV з правильними колонками
        df = pd.read_csv(
            self.filepath,
            sep='\t',
            names=['date', 'time', 'open', 'high', 'low', 'close', 'tick_volume', 'volume', 'spread'],
            skiprows=1 # Пропускаємо заголовок
        )
```

```

# Комбінуємо дату та час в один datetime
df['datetime'] = pd.to_datetime(df['date'] + ' ' + df['time'])

# Встановлюємо datetime як індекс
df.set_index('datetime', inplace=True)

# Видаляємо оригінальні колонки date та time
df.drop(['date', 'time'], axis=1, inplace=True)

# Конвертуємо всі числові колонки до float
for col in ['open', 'high', 'low', 'close', 'tick_volume', 'volume', 'spread']:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Перевіряємо на наявність пропущених значень
if df.isnull().any().any():
    print(f'Попередження: знайдено {df.isnull().sum().sum()} пропущених значень")
    df.dropna(inplace=True)

self.data = df
return df

def get_ohlc(self) -> pd.DataFrame:
    """
    Повертає тільки OHLC дані

    Returns:
        DataFrame з колонками open, high, low, close
    """
    if self.data is None:
        raise ValueError("Дані не завантажені. Спочатку викличте load()")

    return self.data[['open', 'high', 'low', 'close']]

def get_data_range(self, start: Optional[datetime] = None,
                    end: Optional[datetime] = None) -> pd.DataFrame:

```

```
"""
```

Повертає дані за вказаний період

Args:

start: Початкова дата (якщо None - з початку)

end: Кінцева дата (якщо None - до кінця)

Returns:

Відфільтрований DataFrame

```
"""
```

```
if self.data is None:
```

```
    raise ValueError("Дані не завантажені. Спочатку викличте load()")
```

```
df = self.data.copy()
```

```
if start is not None:
```

```
    df = df[df.index >= start]
```

```
if end is not None:
```

```
    df = df[df.index <= end]
```

```
return df
```

```
def get_info(self) -> dict:
```

```
"""
```

Повертає інформацію про завантажені дані

Returns:

Словник з метаданими

```
"""
```

```
if self.data is None:
```

```
    raise ValueError("Дані не завантажені. Спочатку викличте load()")
```

```
return {
```

```
    'start_date': self.data.index.min(),
```

```
    'end_date': self.data.index.max(),
```

```
'total_bars': len(self.data),  
'columns': list(self.data.columns),  
'missing_values': self.data.isnull().sum().to_dict()  
}
```

## ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Система прийняття торгових рішень на фінансових ринках на основі нечіткого кластерного аналізу часових рядів»

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

Підрозділ кафедра АІТ  
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КПІ) 1.0 %

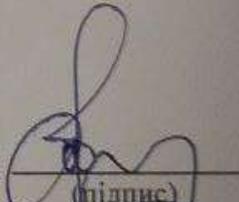
Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

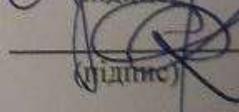
- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють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

Експертна комісія:

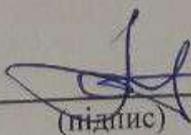
Бісікало О.В., зав. каф. АІТ  
(прізвище, ініціали, посада)

Овчинников К.В., доц. каф. АІТ  
(прізвище, ініціали, посада)

  
(підпис)

  
(підпис)

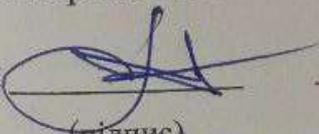
Особа, відповідальна за перевірку

  
(підпис)

Маслій Р.В.  
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

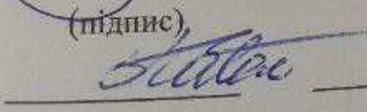
Керівник

  
(підпис)

Маслій Р.В., доц. каф. АІТ

(прізвище, ініціали, посада)

Здобувач

  
(підпис)

Павленко В.В.

(прізвище, ініціали)