

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Інтелектуальний торговий агент для автоматизованої торгівлі
на ринку Forex
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШ-21-1

Кіра Должикова
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)

Керівник ас. Максим Політ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____
(підпис)

Олег ЗОЛОТУХІН
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Должиковій Кірі Ігорівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інтелектуальний торговий агент для автоматизованої торгівлі на ринку
Forex _____

затверджена наказом університету від 19 травня 2025 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 червня 2025 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел, бібліотеки
Python, історичні дані валютних пар, фреймворк Backtrader MT5 й система MetaTrader5 Api

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Аналіз методів вирішення поставленої задачі _____

3) Опис проєктованої системи _____

4) Програмна реалізація торгового агента _____

РЕФЕРАТ

Пояснювальна записка: 96 с., 20 рис., 2 табл., 4 дод., 24 джерела.

АВТОМАТИЗОВАНА ТОРГІВЛЯ, К-НАЙБЛИЖЧИХ СУСІДІВ, МАШИННЕ НАВЧАННЯ, ТОРГОВИЙ АГЕНТ, ТРЕЙДИНГ, ФІЛЬТР ГАУСА, ФОРЕКС.

Об'єкт дослідження – автоматизований процес прийняття торгових рішень на валютному ринку Forex із застосуванням методів машинного навчання.

Предмет дослідження – інтелектуальний торговий агент, який аналізує ринкові дані й приймає торгові рішення в режимі реального часу з використанням методів машинного навчання.

Мета роботи – створити інтелектуальний торговий агент для автоматизованої торгівлі на ринку Forex для надання можливості пасивного заробітку користувачам.

Методи дослідження – аналіз предметної галузі, поставленої задачі й її актуальності у наш час, провести порівняльну характеристику аналогів й методів досягнення мети.

В ході виконання роботи було створено інтелектуальний торговий агент, який аналізує ринок й приймає торгові рішення про виконання угод на ринку Forex в режимі реального часу, що надає можливість користувачам отримувати пасивний дохід без наявності особливих знань у сфері трейдингу.

ABSTRACT

Bachelor's thesis contains: 96 pp., 20 fig., 2 tabl., 4 ann., 24 references.

AUTOMATED TRADING, GAUSSIAN FILTER, K-NEAREST NEIGHBORS, MACHINE LEARNING, TRADING, TRADING AGENT, FOREX.

The object of the research is the automated process on the Forex market using machine learning methods.

The subject of the research is an intelligent trading agent that analyzes market data and makes trading decisions in real time using machine learning methods.

The purpose of the work is to create an intelligent trading agent for automated trading on the Forex market to provide users with the opportunity to earn passive income.

Research methods are analysis of the subject area, the task and its relevance in our time, conduct a comparative characteristic of analogues and methods for achieving the goal.

In the course of the work, an intelligent trading agent was created which can analyze the Forex market and makes trading decisions on the execution of transactions on the market in real time, that allows users to receive passive income without having special knowledge in the field of trading.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ	9
1 Аналіз предметної галузі.....	12
1.1 Загальна характеристика ринку Forex	12
1.2 Валютні пари.....	14
1.3 Фактори руху ринку Forex	16
1.4 Стилi торгiвлi на ринку	17
1.5 Постановка задачі	20
2 Методи вирішення задачі.....	22
2.1 Види аналізу ринку.....	22
2.2 Методи машинного навчання.....	25
2.3 Порівняння інструментів для реалізації агента	30
3 Проєктування системи торгового агента	34
3.1 Опис проєктованої системи	34
3.1.1 Визначення типу користувачів.....	35
3.1.2 Функціональні вимоги	37
3.1.3 Нефункціональні вимоги	37
3.2 Технічний стек	38
3.3 Використані методи, їх параметри й метрики оцінювання	40
3.3.1 Фільтр Гауса й налаштування параметрів	40
3.3.2 kNN й налаштування параметрів	41
3.3.3 Метрики оцінювання ефективності роботи методів	43
3.4 Вхідні дані	44
3.5 Архітектура системи	47
4 Програмна реалізація торгового агента.....	51
4.1 Збір даних	51
4.1.1 Збір даних для класу MT5Connector.....	51
4.1.2 Збір даних для класу TradingEngine.....	52

4.2 Обробка даних.....	54
4.3 Генерація сигналів.....	57
4.4 Виконання угод.....	60
4.5 Інтерфейси користувача.....	64
Висновки.....	74
Перелік джерел посилання.....	75
Додаток А UML-схема для користувачів.....	78
Додаток Б Програмний код торгового агента.....	79
Додаток В Реалізовані інтерфейси торгового агента.....	92
Додаток Г Відомість кваліфікаційної роботи.....	96

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

MH – машинне навчання;

ШІ – штучний інтелект;

ATR – Average True Range – середній істинний діапазон;

kNN – k-Nearest Neighbors – k-найближчих сусідів;

LSTM – Long Short-Term Memory – довга короткочасна пам'ять;

MSE – Mean Squared Error – середньоквадратична похибка;

MT5 – MetaTrader5 – інформаційно-торгова платформа, яка обслуговує ринок Forex;

OHLC – Open High Low Close – ціна відкриття максимуму мінімуму закриття свічок;

OTC – Over-the-Counter – позабіржовий;

SL – Stop Loss – обмежений збиток;

SMA – Simple Moving Average – проста ковзаюча середня;

SNR – Signal-to-Noise Ratio – показник співвідношення сигналу до шуму;

SVM – Support Vector Machine – метод опорних векторів;

TP – Take Profit – фіксований прибуток.

ВСТУП

Світовий фондовий ринок Forex є найліквіднішим й найдинамічнішим типом ринку у світі. Кожен день на ринку здійснюються транзакції на суму більше за 7 трильйонів доларів [1], що у сотні разів перевищує обсяги торгів фондового, товарного чи акційного ринків. Хоча деякі з цих операцій виконуються за практичними цілями (обміну суми грошей з однієї валюти на іншу), то сильно більша частина операцій виконуються з метою отримання прибутку.

Через надзвичайні масштаби об'єму грошей, які щохвилини проходять через ринок Forex, виникають коливання ціни. Такі коливання цін виникають через політичні й світові новини, вислови високопоставлених людей, підйоми або крахи великих фірм, але найбільший внесок в такі коливання привносять звичайні люди, які обмінюють гроші, й приватні-трейдери, які намагаються заробити на цих рухах ціни.

Ринок Forex з першого погляду є абсолютно непередбачуваним. Але саме ця невизначеність робить його привабливим для заробітку приватним-трейдерам й невеликим бізнесам.

Торгівлю на ринку Forex точно не можна назвати повністю безпечною й очікуваною. Наприклад, її можна порівняти зі стихією природи. В один день світить сонечко, гуляють хмарки по небу, дує легкий вітерець – чудова погода й тому ви вирішуєте поїхати на пікнік. А вже через годину небо повністю затягується громовими хмарами, все навкруги стає темний, пориви вітру стають швидше й швидше, наближається ураган, й це аж ніяк не вписувались у ваші плани пікніка й чудово проведеного дня. Звичайно в наші часи вже давно створити додатки за допомогою, яких прямо у вашому телефоні ви можете подивитись приблизний прогноз погоди на сьогоднішній день, й якщо б ви дізнались його заздалегідь, то мало ймовірно зібрались піти далеко від дому. Погода як і ринок Forex складне явище для передбачення, але маючи допоміжну систему, яка буде прогнозувати рухи

ринку або зміну погоди, явно стає легше зрозуміти ситуацію, особливо для людей, які зовсім не розуміються в цих сферах.

Інтенсивні рухи й коливання, які відбуваються на ринку Forex, створюють як і величезні можливості для отримання прибутку, так і високі ризики значних фінансових втрат. Розуміючи цю істину, логічно збагнути що традиційний підхід до торгівлі, який заснований виключно на інтуїції й баченні трейдера, виявляє низку проблем. Наприклад, уповільнена реакція на ринкові зміни. Активний рух цін відбувається на ринку щохвилини, звичайно людина не можна всю добу бути активною й сидіти біля комп'ютера в очікуванні сприятливих умов для угод. Додатково у людини є емоції, а стратегії засновані на інтуїції повністю залежить від настрою людини. Ми можемо самі зрозуміти як це хвилюватись перед прийняттям угоди на великі гроші з можливістю її за секунду втратити, а угода повинна прийматись миттєво, часу на емоції й подумати не має. Тому частота прийняття помилкових рішень зростає, а тому й зменшується кількість зароблених грошей, а той і зовсім трейдер уходить в мінус, бо помилкових угод більше, ніж сприятливих.

У відповідь на ці виклики, з якими стикаються приватні-трейдери, набирають популярність торгові агенти, тобто роботи, які аналізують ринок, оброблюють тисячі рядків інформації щосекунди й миттєво можуть приймати рішення ґрунтуючись на способі аналізу, який був заданий при його створенні.

Зараз з активним розвитком штучного інтелекту, який впроваджується в різні сфери, й трейдинг серед них не є виключенням. Впровадження алгоритмів машинного навчання як способу аналізу даних й прийняття рішень для торгового агента є чудовим варіантом. Такий підхід може використовуватись як для створення системи й допомоги для трейдерів-початківців, так й для людей, які нічого не розуміються в цій сфері, але хочуть пасивно заробляти гроші, інвестуючи власні.

Отже, для успішної роботи на ринку Forex треба вміти миттєво аналізувати дані, прогнозувати подальші рухи ціни й приймати рішення щодо входу в угоди миттєво, при цьому не піддаючись жодним емоціям; з цими завданнями занадто складно впоратись людині самотужки, тому на допомогу приходять нові технології – інтелектуальні агенти, які впораються з цими викликами й стануть в пригоді людям, які навчаються трейдингу й тим, хто хоче пасивно заробити.

Таким чином, в ході виконання практичної частини постає завдання створити інтелектуальний торговий агент для автоматичної торгівлі на ринку Forex з використанням методів машинного навчання. Такий вид агенту буде за допомогою машинно-орієнтованого підходу аналізувати ринок, навчатись на історичних даних й приймати рішення на графіку даних у реальному часі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальна характеристика ринку Forex

Forex (скорочення від Foreign Exchange) – це валютний ринок, в якому великі банки й приватні особи мають можливість виконувати грошові операції: покупати, продавати чи обмінювати валюти [2]. Це найбільший фінансовий ринок в світі. Його глобальна щоденна торгівля склала 7,7 трильйонів доларів за даними «Triennial Central Bank Survey 2022» [1].

Торгівлю на ринку Forex ведуть банки, комерційні компанії, фірми з управління інвестиціями, роздрібні форекс-брокери й інвестори, операції яких у наш час більше ніж у 7 разів перевищують щоденну вартість валюти, порівнюючи з початок тисячоліття [2].

Також важливими фактами про роботу ринку Forex є те, що він відкритий 24 години на добу й 5 днів на тиждень, захоплюючи при цьому усі часові пояси. А так як у вихідні й святкові дні банки, які ведуть основні валютні операції, закриті, то й фінансовий ринок закритий, це означає що показник значення відношення однієї валюти до іншої зупиняється на якомусь значенні, й не змінюється до відкриття ринку. Зазвичай в таких ситуаціях, після закриття ринку на вихідні, формується прогалина в значеннях ціни.

Цілодобовість роботи у будні дні відрізняє валютний ринок від ринку акцій, облігацій й фондових, всі зазвичай відкриті тільки в години активної роботи їх бірж у відповідних країнах торгівлі. Але є виключення й для ринку Forex деякі не надто популярні валютні пари закриваються на час перерви під час торгового дня чи після його закінчення.

Глобальна мережа з тисячі комп'ютерів й брокерів по всьому світу складає з себе ринок Forex. На ньому не переважає єдина біржа, а всі учасники тримають й будують валютний ринок. Брокери також можуть

творити ринок, висуваючи свої ставки й ціни за валютними парами, які відрізняються від більш конкурентоспроможних ставок по ринку.

Ринок Forex є позабіржовим (Over-the-Counter (OTC)) ринком. Він працює за допомогою глобальної децентралізованої мережі, в якій валюти торгуються безпосередньо між продавцями й покупцями, а не за участі центральної біржи. В той час роздрібні трейдери отримують доступ до цього ринку через брокерів. Одною з ключових особливостей позабіржового ринку є його гнучкість. Так як без участі центральної біржи й її прописаних валютних вимог, учасники можуть самі налаштувати розмір й умови угоди. Але оскільки при такому виключенні центральної біржи відсутня стандартизація цін валют, то ринок стає менш прозорим. Ціни валют постійно варіюються, тому покупці часто стикаються з великою відстанню в значеннях (спредом) між ціною попиту й пропозиції в моменти більш низькою ліквідності, особливо для менш популярних пар валют.

Ринок Forex ділиться на 3 види:

- спотовий (Spot Forex Market);
- форвардний (Forward Forex Market);
- ф'ючерсний (Futures Forex Market);
- опціонний (Options Market);
- своповий (Swap Market).

На спотовому ринку відбуваються найшвидші валютні операції. Це фактичний обмін валютними парами. Усі платежі здійснюються негайно після їх підтвердження за обмінний курсом на момент обміну. Приблизно третина всіх операцій на ринку Forex є спотовими [2].

На форвардному ринку дві сторони, учасники угоди, домовляються здійснити угоду в певний час у майбутньому, але за встановленою в прийнятому контракті ціною й кількістю валюти. Учасники мають самостійно оцінити надійність контрагента, щоб знизити кредитний ризик.

Ф'ючерсний ринок це вдосконалений варіант форвардного. Учасники вкладають контракт на покупку чи продаж валют, за яким вирішують певну

ціну й кількість валюти на угоду, яка буде виконана в майбутню дату. Ф'ючерсний ринок відрізняється від форвардного тим, що він публічно торгується на біржі й є юридично зобов'язуючим. Ціни й об'єми торгів є в загальному доступі, тому це знижує ризик контрагенту.

На опціонному валютному ринку учасники торгують опціонами й маю можливість закласти фінансовий контракт, який називається опціоном. Опціонний контракт дає його власнику право (але не зобов'язання) купити чи продати певну кількість валютної пари протягом встановленого періоду часу. Ця покупка чи продаж виконуються по заздалегідь вирішеній ціні, яка називається ціна виконання опціону.

На ринку свопів учасники можуть укласти валютний своп, який є угодою між двома сторонами, який включає в себе одночасне займання однієї валюти на іншу. Ця угода має свою силу на час певного обговореного періоду, після чого обидві сторони обмінюються сумами зворотно. Можуть виконуватись й подвійні угоди, які включають в себе спотовому угоду, за якою йде форвардна торгівля.

Отже, ринок Forex це найбільшим валютним ринок, на якому щодня виконуються тисячі торгових операцій на трильйони доларів з використанням різних валютних пар.

1.2 Валютні пари

На фінансовому ринку Forex всі торгові операції здійснюють парами, кожна з яких складається з двох валют. Валютна пара – це обмінний курс між двома валютами, який вказує, яка частина валютного котирування необхідна для покупки одиниці базовою валюти.

Наприклад, в парі EUR/USD (євро до долара) євро є базовою валютою, а долар є валютним котируванням. Якщо припустимо обмінний курс EUR/USD дорівнює значенню 1,2, це можна описати як те, що один євро можна обміняти на 1,2 долару.

Усі валютні пари представлені на ринку Forex діляться на чотири категорії:

- мажорні (major);
- мінорні (minor);
- екзотичні;
- регіональні.

До мажорних відносяться 7 валютних пар: EUR/USD, USD/JPY, GBP/USD, USD/CHF, AUD/USD, USD/CAD, NZD/USD. 85% усіх операцій на ринку Forex виконують ці пари, а з цього 22,7% операцій виконувались з EUR/USD (за статистикою 2022 року) [3]. Так як на цих валютних пар тримається весь фінансових ринок, тому вони й мають найбільшу ліквідність й нестабільність рухів й коливань.

Мінорні пари рідше використовуються для торгівлі, але також мають свою важливість й популярність на ринку форекс. Мінорні пари складають з двох основних валют, які обмінюються одна на одну, але не пов'язані з долларом. Наприклад, EUR/GBP, GBP/JPY, AUD/CHF, тощо.

До екзотичних валютних пар відносяться пари, які складаються з однієї з основних валют, яка обмінюється на валюти країн, які розвиваються [4]. Наприклад, USD/PLN, GBP/MXN, EUR/CZK, тощо. Екзотичні валютні пари складають найменший відсоток від усіх торгових операцій на ринку.

Регіональні пари – це пари, які класифіковані за розташування країни між валютою яких йде обмін. Наприклад, у EUR/НОК для скандинавського регіону.

Отже, щоб валютні пари це основа фінансового ринку Forex, тому щоб його успішно зрозуміти треба розрізняти, які пари є більш спокійними, а які навпаки створюють активні рухи і коливання та є ліквідними для торгівлі. Такий аналіз можливості руху валютної пари буде потрібним при виборі валютних пар, на яких буде виконуватись торгівля в торговому агенті.

1.3 Фактори руху ринку Forex

Основними учасниками фінансовими ринку Forex є центральні й комерційні банки, компанії з виконання зовнішньо-торгових операцій, інвестиційні фонди й організації, валютні біржі, брокерські компанії, дилінгові центри і приватні роздрібні трейдери.

Як бачимо на фінансовому ринку багато учасників, які виконують торгові операції при цьому рухаючи сам ринок. На коливання ринку впливають які й десятки видів валютних пар, які торгуються на Forex, так і багато різних факторів, які впливають на коливання цін.

Основний рух на ринку Forex створюють його учасники, бо відбувається зріст або спад попиту й пропозиції, які прямо впливають на зміни ціни й графіку валютних пар.

Звичайно великі компанії з інвестицій й банки хочуть вкладати свої гроші, маючи при сильні перспективи заробітку. При цьому позитивні новини проникають на ринки в певному регіоні, що буде стимулювати інвестиції й підвищить попит у людей на дану валюту в певному регіоні [5]. При навпаки потраплянні негативних новин про перспективи якоїсь валюти, то очікується зниження попиту.

Настрій ринку – один з основних аспектів руху ринку. Зазвичай на настрої ринку прямо впливають новини. І якщо інвесторами очікується певний рух валюти, то вони й самі будують її торгувати в тому ж напрямі й запевняти в цьому інших, тим самим власноруч збільшити або зменшити попит на очікувану валюту [5].

Звичайно економічні дані як основа руху всього фінансового ринку у світі й частіше всього за ними останнє рішення при певному руху цін на валюту. По-перше, економічні дані прямо означають як функціонує економіка в регіоні, де обрана валюта є популярною. По-друге, дають представлення, що далі з цією валютою будуть робити центральні банки.

Оприлюднення ставки кредитного рейтингу кожного місяця одна з основних новин для всього фінансового ринку, яка викликає великі коливання й зміни руху цін мажорних валют. Так відбувається через те, що інвестори прагнуть максимально збільшити свій прибуток від ринку, але при цьому мінімізувати ризик торгівлі. Отже, для цього продуктивно використовувати не тільки процентну ставки й економічну статистику, а й кредитні рейтинги для аналізу інформації куди краще вкласти гроші, щоб заробити як найбільше при цьому не маючи великого ризику втрати.

Щоб використовувати ринок Forex для отримання прибутку треба зрозуміти через що виникають основні рухи й коливання ринку й хто є його учасниками й створює ліквідність й попит на певні валюти.

1.4 Стилi торгiвлi на ринку

Якщо ви не є центральним банком, а звичайний початківець в трейдингу, то ваш спосіб заробітку на ринку Forex це прогнозування змін на валютних парах, проаналізувати графік цін й прийняти рішення чи буде йти укріплення чи навпаки ослаблення однієї валюти по відношенню до іншої. Коли ціна пари росте, це означає, що база валютної пари укріплюється відносно котирування, а коли вона падає, навпаки, база ослаблює проти котирування [5]. Так відбувається тому, що за одну базу потрібно більше заплатити котирування у випадку якщо ціна валютної пари зростає й, навпаки, менше котирування, якщо ціна падає. Тому трейдери обирають покупати (на мові трейдингу заходити в Long), якщо база укріплюється відносно до валюти котирування, або продавати (заходити в Short), якщо база послаблюється.

Тепер розглянемо стилі торгівлі на ринку Forex, які зазвичай використовують роздрібні трейдери.

За часовими характеристиками стилі поділяються на:

– короткострокові (угоди, які приймаються у межах одного дня);

- середньострокові (угоди, які зазвичай тривають декілька днів);
- довгострокові (угоди, які тривають тижнями, місяцями, а тої роками).

Далі стилі поділяються на 4 основні типи [6] в залежності від кількості й тривалості здійснення угод:

- скальп-торгівля (Scalp trading);
- внутрішньоденна торгівля (Intraday trading);
- свінг-торгівля (Swing trading);
- позиційна торгівля (Position trading).

Скальпінг – це найкоротша за часом торгівля, угоди тривають від декількох секунд до хвилин, але потрібно виконувати сотні торгових угод за робочий день. Ідея такого стилю торгівлю полягає в тому, що відкрити угоду й вийти з неї, як тільки ринок піде на вашу користь, тим самим заробити мало в мірі однієї угоди, але часто.

Також скальпінг вважається дуже швидким й інтенсивний видом торгівлі, тому потребує високої концентрації, стабільної психіки й швидкого аналізу ринку.

Внутрішньоденна торгівля передбачає відкриття позицій протягом робочого дня й обов'язково їх закриття до завершення торгового дня (без переносу на наступний день). Такі трейдери виконують декілька угод на день (зазвичай до 10) використовуючи декілька валютних пар. Інколи вигідно відкривати декілька угод на одній парі за день, щоб використати короткострокові ринкові рухи. При цьому вони виключають деякі ризики й додаткові витрати, які пов'язані з переносом угод на наступний день.

Головне завдання трейдера з таким стилем торгівлі це повністю дотримуватись прописаної стратегії торгівлі й не забувати про ризик менеджмент, щоб одна невдала угода не перекреслила декілька прибуткових. Для цього трейдери використовують стопи й ліміти. Встановлення стоп-лосса (Stop Loss (SL)) до позиції дозволяє зберегти зафіксований ризик й закрити позицію при його досягненні, щоб витрата

грошей на цю угоду не перевищила прибуток інших. В той час як при досягненні ліміту (Take Profit (TP)) зафіксується прибуток угоди, це дасть можливість отримати визначений відсоток прибуток, застерігаючи від ситуацію, що ринок може розвернутись.

Свінг-торгівля – це стиль, який фокусується на тому, щоб зайняти позицію в рамках більш великих рухів на ринку [6]. Свінг-торгівля використовує середньостроковий стиль, угоди зазвичай тривають декілька днів або тижнів.

Не існує точного показника тривалості виконання угод, все залежить від ринку й тривалості його рухів й змін. Тому угоди можуть тривати від години до декілька тижнів. Угоди так само закриваються при досягненні допустимого прибутку чи збитку за допомогою стопів й лімітів.

Позиційна торгівля полягає у виконанні угод, тривалість яких є від декількох місяців до років. Такому стилю торгівлі не цікаві короткі рухи й коливання на графіку, а потрібні всеохоплюючі рухи, які призводять до великих змін на ринку.

Позиційна торгівля найбільше всього схоже на інвестування, коли ви покупаете який інструмент й чекаєте поки його ціна збільшиться, щоб продати, й ваш заробіток це різниця між ціною покупки й продажу. Але на відміну від інвестицій, позиційна торгівля може працювати й в іншу сторону, для продажу.

Позиційна торгівля включає в себе відкриття мінімальної кількості угод, але при цьому їх цінність буде набагато більшою в порівнянні з іншими стилями торгівлі. Хоча це й збільшує прибуток трейдера й зменшує час проведення біля комп'ютеру для аналізу графіків, але водночас збільшується схильність до ризику через малу кількість угод й їх довгу тривалість. Тому позиційні трейдери повинні повністю притримуватись торгового плану, щоб знати коли угоду ще можна почекати до її досягнення максимуму, а коли краще вийти з угоду, зафіксувавши вже отриманий прибуток, запобігаючи втратити все.

1.5 Постановка задачі

Ознайомившись з основною інформацією про ринок Forex можна прийти до чітких висновків, що це реальний спосіб заробітку, але він складний і не кожен зможе досягти успіхів в торгівлі на фінансовому ринку. Також ринок є сильно змінним з великими коливання, що ускладнює роботу. Й для отримання хороших результатів на торгівлю потрібно витратити увесь робочий день.

Щоб людина досягла хороших результатів у трейдингу на ринку Forex підуть роки на відточування майстерності, а можливо це буде й недосяжним завданням. Бо за статистикою банку Compare Forex Brokers [3] проведеного дослідження оцінили 35 брокерів Forex, було виявлено, що в середньому 71% приватних трейдерів зазнають збитків від торгівлі. Інші 29% – отримують прибуток, але 99% з них виходять в мінус протягом чотирьох кварталів поспіль [7]. Оцінивши описану статистику, з'являється проблема мінімізації ризиків й збільшення прибутку торгівлі.

Торгівля на ринку Forex зазвичай складний й нестабільний процес для людини-трейдера, особливо початківця, який намагається сам здолати систему ринку й не те, щоб заробити гроші, а хоча б їх не втратити. Людина повільно реагує й не може цілодобово знаходитись за графіками, коли машина має можливість здійснювати угоди в будь-який час й за лічені секунди. Але найбільша проблема людини в торгівлі це психологія та її емоції, що призводять до помилкових рішень, тим паче коли угоду треба прийняти за декілька секунд, й відхилення від прописаних правил стратегії. В той час коли машина не піддається жодним емоціям й працює повністю за правилами, внесеними у систему.

Отже, метою кваліфікаційної роботи є створенні інтелектуального торгового агента для торгівлі на ринку Forex з використанням методів машинного навчання. Торговий агент повинен навчатись на історичних

даних й виконувати угоди в режимі реального часу. Він повинен мати інтуїтивно-зрозумілу структуру для користувачів.

Для досягнення поставленої мети необхідно виконати низку завдань, які деталізують порядок виконання роботи:

- проаналізувати предметну галузь;
- обрати метод машинного навчання як основу торгового агента;
- обрати інструменти для реалізації програми торгового агента;
- визначити вимоги до агента;
- реалізувати торговий агент з використанням методів машинного навчання;
- під'єднати створений агент до платформи для доступу до брокерів й демо-рахунку;
- розробити візуальний інтерфейс для торгового агента;
- провести тестування системи.

Таким чином, для подолання ринку Forex треба створити систему, яка буде власноруч аналізувати ринок й приймати рішення щодо відкриття угод. Тим самим облегшивши життя початківцям у трейдингу й надасть можливість людям не знайомими з торгівлею заробляти гроші пасивно, не прикладаючи до цього жодних зусиль.

2 МЕТОДИ ВИРІШЕННЯ ЗАДАЧІ

2.1 Види аналізу ринку

В попередньому розділі було з'ясовано, що ринок Forex це складний механізм. Тому неважливо людина чи машина, хто б не намагався заробити гроші на ринку, повинен спочатку проаналізувати графік зміни цін, а потім починати торгівлю.

Існують сотні різних стратегій аналізу ринку, якісь прибуткові, якісь ні. Але кожна людина-трейдер підбирає стратегію максимальну близьку по душі, яку б їй було легше дотримуватись. Незважаючи на таку кількість різноманітних стратегій, усі стилі аналізу можна поділити на 3 види:

- фундаментальний аналіз (Fundamental Analysis);
- технічний аналіз (Technical Analysis);
- аналіз настроїв (Sentiment Analysis).

Фундаментальний аналіз – це процес дослідження економічних, політичних й соціальних факторів, які можуть вплинути на ціну валюти [8]. Також до фундаментального аналізу відноситься дослідження основних компаній, які виготовляють торгові інструменти чи надають якісь послуги. Основною ціллю такого виду аналізу є виявлення того, чи вигідна на даний момент інвестиційна й торгова можливість для даних валютних пар. Фундаментальні аналітики, люди чи машини, вивчають економічну фактори, які могли вплинути на те, як актив рухається, що може показати, чи відбувається торгівля вище чи нижче його реальної стандартної ціни.

Існує два типи фундаментальних факторів: кількісний й якісний аналіз.

Кількісний фундаментальний аналіз розглядає фактори пов'язані напряму з грошима, наприклад, такі як фінансові коефіцієнти, ринкову капіталізацію, політику відсоткових ставок й прибуток на акцію [8]. Типи кількісних даних будуть залежать від того, який актив досліджується. Далі

отриманні результати порівнюються з іншими активами або середніми показниками по регіону валюти, щоб визначити, наскільки вони є прибутковими.

Якісний фундаментальний аналіз визначає фактори, які демонструють якість активу. Наприклад, якість управління, політичну обстановку, популярність валюти у світі, настрої населення країни, тощо. Цей підвид аналізу більше зосереджується на аналітичних показниках, які не можна точно описати.

Фундаментальний аналіз наразі має багато переваг, який є одним з популярних видів для прийняття рішення на довгий термін. Такий вид аналізу дає більш чітку картину того, що відбувається на ринку, але новини й фактори не є швидкими носіями інформації.

Технічний аналіз досі залишається найпопулярнішим методом аналізу, який здійснюється на пряму з графіком ціни. Технічний аналіз використовується для прогнозування можливого майбутнього руху ціни на валютну пару, на основі ринкових даних.

Технічні трейдери вважають, що теперішні й минулі цінові рухи на ринку є найбільш надійними показниками майбутньої цінової дії [9]. Все що зараз відбувається на графіку, колись вже відбувалось.

Технічні аналітики використовують широкий спектр інструментів й різних індикаторів для аналізу цінового графіку й виявлення відповідних закономірностей змінення ціни. До найпопулярніших інструментів відноситься ковзаючі середні (технічний індикатор, який розраховується шляхом додавання ціни закриття валютної пари протягом певного періоду часу й ділення загальної ціни на відповідну кількість торгових днів [9]), лінії тренду (являють собою рівні підтримки й супротиву), осцилятори, такі як індекс відносної сили (вимірює швидкість й зміну руху ціни).

Основною різницею між технічним й фундаментальним аналізом є їх підходи до дослідження валют. Технічний аналіз фокусується на аналізі історичних даних, рухів й закономірностей, які вже відбувались на графіку,

у той час коли фундаментальний аналіз вивчає основні фактори й характеристики активів, які можуть вплинути на його рух. Технічний аналіз використовується для коротко- й середньострокових угод, а фундаментальний – для довгострокових.

Аналіз настроїв – це підхід до вивчення колективного настрою психології учасників ринку [10]. Основна ціль виявлення загального настрою ринку чи він є оптимістичним, чи, навпаки, песимістичним.

Аналіз настрою ринку допомагає оцінити переважне відношення до валюти більшості, це дозволяє їм передбачити потенційні зміни чи продовження тренду ринку. Але це доволі не легка задача, розуміючи факт, що на настрої людей може вплинути економічні й політичні новини, що відбуваються у певному регіону.

Найпопулярнішими показниками для ефективного вимірювання ринкових настроїв є звіт з обов'язків трейдерів, який надає повну інформацію про положення великих учасників ринку, індекс страху й жадібності, який вимірює рівень показників й надає знімок теперішнього емоційного стану трейдерів, й індекс волатильності, який вимірює очікуваний рівень волатильності на графіку.

Спори й роздуми про те, який вид торгівлі кращий будуть завжди актуальними. На це впливають й особисті людські фактори, й цілі, які переслідує трейдер на ринку. На сьогоднішній день найпопулярнішим залишається технічний аналіз, бо він використовує багато різних інструментів, де кожен може обрати інструмент для аналізу по душі й ефективності, й для самого аналізу потрібен лише актуальний графік зміни ціни валютних пар, не потрібно використовувати багато джерел для аналізу ситуації поза межами графіку, це спрощує процес роботи трейдеру, бо його фокус направлений тільки на ціновий ряд, не потребуючи цілі аналізувати настрої валютної пари, регіону й фундаментальні чинники росту й падіння графіку.

2.2 Методи машинного навчання

Один з підвидів технічного аналізу є використання алгоритмічного аналізу в торгівлі на ринку Forex [11]. Алгоритмічна торгівля включає в себе використання систем для автоматизації торгового процесу. Зараз автоматизація досягла рівня використання машинного навчання, за допомогою якого не просто автоматизується процес виконання прописаних у системі правил стратегії торгівлі, а система використовуючи широкі набори даних самонавчається й ефективність стратегії з часом зростає [12].

Основними перевагами використання саме машинного навчання для автоматизації систем є швидкість й ефективність. Такі моделі можуть оброблювати й аналізувати тисячі рядків історичних даних за секунди, що дозволяє пришвидшити процес прийняття рішень й виконання торговельних угод. Також важливим фактором є точність моделі. Так само моделі навчаються на історичних даних й з часом вдосконалюються, що призводить до виконання більш надійних й чітких прогнозів. Людина завжди знаходиться під впливом емоцій, особливо при прийнятті складних рішень. Одним з основних переваг є виключення емоцій, які можуть призвести до ірраціональних рішень.

Існують десятки методів машинного навчання. Щоб вирішити який саме метод обрати порівнюємо такі 5 алгоритмів: k-найближчих сусідів, фільтр Гауса, SVM, Random Forest й LSTM.

Метод k-найближчих сусідів (kNN) є одним з базових алгоритмів машинного навчання. kNN використовується для визначення актуальної ситуації, яка відбувається на ринку в даний момент, це відбувається на основі аналізу історичних аналогій, й далі приймається рішення про вхід в угоду.

Принцип роботи алгоритму kNN є надзвичайно простим для розуміння. Етап навчання відбувається швидко, бо алгоритм лише зберігає всі вхідні приклади історичних даних. А вже на етапі прогнозу

вираховується відстань від нового вектору ознак до кожного з цих оброблених прикладів [13]. Після сортування точок за відстанню обираються перші k сусідів й прогноз формується як клас, який домінує серед цих сусідів.

Отже, kNN не потребує реалізації складної внутрішньої моделі, бо прогноз базується безпосередньо на самих даних. Але метод kNN занадто чутливий до шумів в даних й при дуже великих масивах історичних даних потребує додаткового й часу, й вартості через потрібність прогнозу для кожної точки у даних.

Алгоритм Support Vector Machine (SVM) знаходить гіперплощину, яка максимально віддалена від найближчих прикладів класів [14]. Якщо дані трохи переплітаються з площиною, тоді декільком точкам дозволяють такий підхід, але застосовуючи м'який штраф (soft-margin). Якщо ж між класами наявна нелінійна залежність, то використовується такий інструмент як kernel-trick. Тобто він спершу переносить всі дані у вищий вимір, де їх вже буде легше розділити, а потім вже у цьому новому просторі знаходить гіперплощину, яка намагається бути максимально віддалена від точок й не перетинає їх.

У підсумку модель запам'ятовує тільки декілька опорних точок, які знаходяться найближче до межі розмежування між класами. SVM має високу точність на обмеженому наборі даних, але при великій кількості даних обчислення стають дуже складними. Також алгоритм погано працює, якщо дані суттєво перетинаються й складно визначити окремі класи, бо в такому випадку буде багато точок який будуть отримувати штрафи.

Метод Long Short-Term Memory (LSTM) це різновид рекурентних неймереж, який є спеціально адаптований для роботи з часовими рядами, наприклад, [15] такими як зміна ціни валютної пари в залежності від часу. На ринку Forex LSTM використовується для прогнозування майбутніх значень цін й показників індикаторів.

Навчання LSTM відбувається на послідовних історичних даних й метод запам'ятовує довго- й короткострокові закономірності між ними. Аналіз відбувається через спеціальний механізм, який вирішує яку саме інформацію відкинути чи зберегти, що дозволяє відстежити закономірності між трендами й коливаннями, які вже відбувались на графіку. Після проходження етапу навчання алгоритм може видавати сигнали на купівлю чи продаж.

Основними перевагами LSTM є здатність виявляти складні нелінійні залежності й враховувати вплив подій на графік. Але щоб уникнути перенавчання й забезпечити швидке реагування, потрібно ретельно налаштувати гіперпараметри роботи алгоритму й мати достатній об'єм пам'яті для їх зберігання.

Фільтр Гауса це лінійний фільтр, який використовується для зменшення випадкового шуму в даних, й робить лінійний графік більш чітким й якісно згладженим для подальшого використання.

Фільтрація виконується наступним чином: кожне значення нового ряду обчислюється як зважена сума сусідніх цін, причому значення ваги задається гаусівською кривою, що плавно спадає з відстанню від поточної точки [16]. У результаті відсікають викиди, які є ринковим шумом, а залишаються лише довгі чіткі тренди.

На ринку Forex використання методів фільтрації дозволяє знизити кількість хибних сигналів й підвищити точність алгоритмів. Але треба бути обережним з налаштуванням гіперпараметрів, щоб не відсікти важливі частини тренду зміни ціни.

Random Forest – це ансамблевий метод, який використовує комбінацію з багатьох випадкових дерев для більш точного прогнозу [17]. Спочатку з набору історичних даних про зміну ціни формуються багато підвибірок з повторенням, на кожній з яких буде навчатись окремо дерево рішень. Для зниження перенавчання й кореляції між деревами, при побудові кожного вузла дерева розглядається випадкових набір ознак.

На ринку Forex алгоритм Random Forest використовується для класифікації напрямку руху ціни.

Однією з головних переваг є стабільність й невразливість до перенавчання за допомогою спеціальною техніки навчання, а також автоматичного відбору найважливіших ознак, який полегшує процес написання самого механізму. При великій кількості даних створюється дуже велика кількість дерев рішень через прогнозування буде відбуватись повільно. Якщо в даних присутній дисбаланс, то результати можуть бути упередженими. Через доволі специфічний процес прогнозування й роботи самого методу складно інтерпретувати й відділити один прогноз від іншого, тому можуть бути складнощі при усунуванні особливих проблем чи спробах покращення дій алгоритму.

2.2.6 Порівняння алгоритмів

Ознайомившись зі п'ятьма алгоритмами машинного навчання можемо порівняти їх за такими характеристиками: застосування на ринку Forex, час навчання, швидкість виконання прогнозу, вимоги до обсягу даних, стійкість до шуму, адаптивність в онлайн режимі. Всі показники є особливо важливими адже майбутня система буде виконувати всі прогнозування й угоди в реальному часі, тому система повинна бути налагоджена й готова працювати за будь-яких обставин й на різних ситуаціях на ринку Forex.

Аналізуючи таблицю 2.1 «Порівняння методів машинного навчання» можемо прийти до висновку, що використовуючи тільки один з цих алгоритмів не вдасться повністю вирішити поставлену задачу у вигляді реалізації торгового автоматичного агента. Всі використанні характеристики є дуже важливими для створення стратегії, але саме швидкість роботи й стійкість до шуму є надзвичайно потрібними в умовах роботи на неспокійному ринку й з високою швидкістю прогнозувати й виконувати угоди в реальному часі.

Таблиця 2.1 – Порівняння характеристик методів машинного навчання

Метод	Застосування	Стійкість до шуму	Швидкість	Вимоги до даних	Адаптивність
kNN	Класифікація напряму руху ціни	Низька	Висока	Середні	Так
LSTM	Прогноз часових рядів	Середня	Низька	Високі	Ні
SVM	Класифікація руху ціни	Середня	Середня	Середні	Ні
Random Forest	Ансамблева класифікація руху ціни	Висока	Середня	Середні	Ні
Фільтр Гауса	Згладжування цінового ряду	Висока	-	Низькі	Так

Порівнюючи описанні в таблиці 2.1 методи найкраще буде використати тандем з методів k-найближчих сусідів й фільтру Гауса. Метод kNN легко створюється й є високо ефективним й швидким для роботи на графіку зміни ціни, але має високі потреби до даних й низьку стійкість до шуму. Тому додатковим методом пропонується обрати фільтр Гауса, який буде згладжувати ціновий ряд, усуваючи відхили й залишаючи довгі тренди. Таке поєднання повинно мати найбільшу ефективність й продуктивність для торгівлі в умовах реального часу, бо один алгоритм компенсує недоліки іншого, тим самим покращуючи створену стратегію торгівлі. Поєднання kNN й фільтру Гауса забезпечить спочатку ефективне згладжування цінового ряду, а потім використання методів для прогнозування майбутнього руху валютної пари й створення сигналів.

2.3 Порівняння інструментів для реалізації агенту

Для виконання поставленої задачі, а саме реалізації автоматизованого торгового агенту з можливістю виконання угод в режимі реального часу й проходження бектестів для перевірки ефективності роботи алгоритмів існують декілька популярних середовищ й інструментів для розробки. Аналоги торгового агенту, які існують, використовують різні інструменти в залежності від задач, які повинен торговий агент виконувати, й мети, що поставлена. Порівняємо три різні середовища для розробки, а саме: фреймворк Backtrader, платформу QuantConnect Lean й MetaTrader 5, єдина схожість у цих інструментах це використання мови програмування Python.

Backtrader є одним з найпопулярніших фреймворків Python для локального бектестингу торгових ідей через свою гнучкість й легкість реалізації для будь-якої стратегії. Backtrader дозволяє повністю контролювати поведінку алгоритму. Але прийдеться самостійно підключати до коду історичні дані й вказувати спеціальні параметри, наприклад, спред, комісію, початковий капітал, валюту, тощо. Після завантаження характеристик й написання стратегії роботи, Backtrader відтворює симуляцію угод й виводить докладні звіти й графік за проведеною роботою на історичних даних.

Основною перевагою є простота реалізації, можливість гнучко налаштувати принцип роботи стратегії, а також жодні обмеження не прив'язують до конкретного брокера, бо всі параметри задатуються власноруч. Цей одночасно й недолік, потрібно власноруч збирати всі дані. А також робота система виконується локально, тому якість й швидкість обробки буде напряму залежити від продуктивності роботи машини. Й основний недолік, що всі угоди відбуваються на історичних даних, це й називається бектестинг, а для виконання задачі потрібно реалізувати агент для виконання угод в режимі реального часу.

QuantConnect Lean – це відкрита open-source платформа, яка поєднує можливість локального написання коду на Python й доступ до хмарного бектестингу на великому об’ємі історичних даних. Локальна розробка з хмарою поєднується через можливості Docker контейнерів. Платформа надає миттєвий доступ до історичних даних з їх постійним оновленням й багато вбудованих інструментів для перевірки ефективності стратегії й оптимізації гіперпараметрів. Lean має можливість виконання угод для бета-торгівлі в режимі реального часу, що є хорошим для демонстрації проєкту, але не дійсним в умовах справжнього завдання.

Основними перевагами QuantConnect Lean є доступ до тисячі різних історичних даних й інструменти для автоматичного аналізу стратегії на її ефективність, стійкість, тощо. Реалізація відносно складна, бо при локальному запуску доступ до платформи отримується за допомогою Docker контейнерів, але звітність й графіки можна налаштувати для демонстрації у додатку локально. Основним недоліком є відсутність можливості безкоштовно спробувати роботу платформи, треба заплатити й за історичні дані, які ви хочете використовувати, й за оренду хмари для самого бектестингу.

MetaTrader 5 Python Арі це поєднання коду з торговим терміналом й брокерським сервером, що дозволяє отримувати дані в режимі реального часу й автоматично відправляти угоди й їх ліміти для виконання. Для цього потрібно локально написати код на Python, до нього підключити фреймворк MetaTrader 5, завантажити однойменний додаток, підключити його до обраного брокера (від цього буде залежити й список доступних валютних пар, й спреди, й комісія), далі код зможе аналізувати графік, який отримує, й відправляти угоди на виконання в реальному часі.

Такий підхід може використовуватись і як для відпрацювати агенту для демо-торгівлі (торгівля відбувається на уявні гроші, але все інші дані справжні), так і реальної торгівлі на live-рахунку (використовуються ваші гроші). Таким чином дотримуються реальні умови ринку. Недоліком є

прямий зв'язок з обраним брокером, бо від цього також буде залежити якість й стабільність з'єднання, а також врахування умов конкретного торгового рахунку. А також можливість реалізації системи тільки на Windows, що є недоліком за відсутність мультисистемного підходу до використання агенту.

В таблиці 2.2 «Порівняння інструментів для реалізації агенту» приведено аналіз порівняння інструментів за важливими для створення торгового агента характеристиками.

Таблиця 2.2 – Порівняння інструментів для реалізації агенту

Інструмент	Доступ до даних	Масштабування	Виконання угод	Складність реалізації	Вартість
Backtrader	Самостійне завантаження	Локальне	Ні	Легко	Безкоштовно
Quant Connect Lean	Хмара	Високе (Docker)	Ні	Складно	Платно
MetaTrader 5 Арі	Надає брокер	Локальне	Так	Середньо	Безкоштовно

Таким чином, за аналізованими характеристиками найкращим інструментом для використання для виконання торгових рішень в режимі реального часу є MetaTrader5 Арі, який через брокера надає велику кількість даних, що оновлюються постійно, найбільш точно відтворює умови ринку, має можливість виконання угод миттєво й може не тільки використовуватись як приклад проєкту, а мати справжню цінність, бо має доступ до справжньої торгівлі на ринку Forex.

Додаток написаний з використанням MetaTrader5 можна в майбутньому допрацьовувати й оновлювати в залежності від поставленої

задачі. Єдиний суттєвий недолік це потреба обрати брокера через якого будуть виконуватись угоди на ринку Forex, а від цього будуть залежить й комісії на виконання угод, й кількість валютних пар до яких буде наданий доступ, а також потрібність реалізації агенту виключно на системі Windows.

А для виконання бектестингу й перевірки ефективності системи буде за найкраще використати фреймворк Backtrader, який є легким й ефективним інструментом для визначення продуктивності торгових рішень на історичних даних. Backtrader буде використовуватись як основа визначення найкращих параметрів для системи й перевірки її ефективності.

3 ПРОЄКТУВАННЯ СИСТЕМИ ТОРГОВОГО АГЕНТУ

3.1 Опис проєктованої системи

Проєктований об'єкт – інтелектуальний торговий агент для автоматизованої торгівлі на ринку Forex. Проєктований об'єкт складається з наступних складників:

- модулів збору даних (котирувань валютних пар, таймфреймів, історичних даних);
- модулів обробки й класифікації даних (підготовка й нормалізація ринкових даних, створення свічок Heiken Ashi);
- моделей машинного навчання (kNN й фільтр Гауса);
- модуль генерації торгових сигналів (на основі машинного навчання виконується обробка алгоритмів й знаходження сигналів для входу в угоду);
- модуль двигуну прийняття рішень (аналіз сигналів й відправка відібраних сигналів до MetaTrader з прописаними умовами виходу з угоди, лотами, тощо);
- модуль відображення результатів стратегії й візуалізації графіків валютних пар (відображення результатів аналізу стратегії, візуалізації графіків валютних пар);
- модуль бектестування (аналіз ефективності створених алгоритмів торгової системи);
- інтерфейс реєстрації й входу в систему для користувачів (створення й вхід користувача до акаунту в системі);
- інтерфейс взаємодії з користувачем (зміна параметрів акаунту, внесення грошей, початок й зупинення торгівлі);
- інтерфейс наладжиків (виконання бектестування з налаштування особливих параметрів й аналіз торгівельних угод користувачів).

Отже, завданням виконання проєкту є створення інтелектуального торгового агента для автоматичного виконання угод в режимі реального

часу з використанням описаних в даному розділі модулів для роботи системи й зручного для користувачів інтерфейсу.

3.1.1 Визначення типу користувачів

Користувачі торгового агенту – це особи, які використовують створену систему для отримання пасивного доходу без наявності особливих знань у сфері трейдингу на ринку Forex.

Цільова аудиторія даної проєктованої системи є користувачі у віці від 20 до 40 років, які добре вміють володіти комп'ютером й шукають шляхи пасивного заробітку. Тобто, мають гроші, але не мають часу на їхнє примноження. Користувачі можливо мають якісь знання про ринок Forex, але не є обов'язковим розбиратись у трейдингу. Головне це знання англійської мови для розуміння інтерфейсу й наявність грошей для вкладення для подальшого отримання пасивного доходу.

Головна ціль користувачів – це заробіток пасивних грошей без вкладення своїх знань та часу. Вони повинні внести певну суму грошей за тарифом для початку роботи боту, й після завершення роботи вивести зароблені гроші й оплатити відсоток за використання системи. Користувачі вносять гроші, обирають параметри (тарифний план, валютну пару, прийнятний ризик менеджмент), запускають агент й після отримання достатнього прибутку зупиняють роботу систему та виводять зароблені гроші й оплачують відсоток за використання системи. Також перед початком роботи користувач повинен ознайомитись з правилами роботи системи й підтвердити цей пункт натискання на галочку під час реєстрації акаунту, але при цьому не обов'язково потрібно поглиблюватись в саму стратегію заробітку.

Робоче місце користувача повинно бути тихим й зручним. Немає ніякого значення, де саме буде він знаходитись, в якому місці чи в якій країні. Головне робоче місце повинно мати персональний комп'ютер чи

ноутбук й стабільне підключення до інтернету. Воно повинно бути освітленим, тихим й комфортним для роботи. Користувач повинен усвідомлювати важливість процесу й уважно прочитати документи з правилами перед активацією системи. Якщо є бажання користувач може ознайомитись зі стратегією, яку саме використовує система, але це не є обов'язковим.

Вимоги користувачів до системи:

- зручний та інтуїтивно-зрозумілий інтерфейс;
- доступність для перевірки роботи системи й статистики в будь-який час доби;
- зручна функція виводу зароблених грошей;
- пасивний заробіток, без наявності спеціалізованих знань й витрати свого часу;
- технічна підтримка у вигляді боти для вирішення питань по оплаті.

Також для підтримки належного стану системи, виправлення помилок, проведення бектестування торгових рішень й аналізу виконаних угод у системі наявні наладчики. Наладчики – це працівники системи, які підтримують якісну й ефективну роботу торгового агента.

При реєстрації наладчики відрізняються від користувачів тим, що мають «.tech@gmail» у електронній пошті. Таким чином система розпізнає хто виконує вхід користувач чи наладчик, тим самим відкриваючи різні вікна в залежності від цього.

Вимоги наладчиків до системи:

- можливість працювати з системним меню при цьому не зупиняючи виконання угод користувачів;
- забезпечення поточною інформацією про стан системи, алгоритмів, брокера й індикаторів;
- збір статистики й аналіз проведених угод;
- бектестування системи з особливими параметрами, для різних валютних пар, таймфреймів й за різний проміжок часу.

Отже, людина реєструє акаунт й система за особливістю електронної пошти відрізняє користувача від наладчика. Користувач використовує система за основною функцією – отримати пасивний дохід, а наладчик налаштовує систему, слідкує за її ефективністю й аналізує виконанні системою угоди.

3.1.2 Функціональні вимоги

Функціональні вимоги системи описують, які функції система повинна виконувати.

Даний проєктований об'єкт має наступні функціональні вимоги:

- збір даних (збереження історичних котирувань валютних пар й котирувань, що відбуваються в режимі реального часу);
- обробка даних (нормалізація даних);
- генерація ознак (розрахунок індикаторів, використання kNN й методу Гауса по відношенню до даних);
- прийняття рішень (купувати, продавати чи чекати);
- виконання угод в режимі реального часу (створення, контроль й відправка угод до MT5 брокеру);
- виконання бектестування (за особливими параметрами для різних валютних пар й таймфремів);
- інтерфейс й звітність (UI/UX інтерфейси для користувачів й наладчиків).

Отже, проєктований об'єкт має такі перелічені функціональні вимоги для задовільнення мети проєкту й усіх складових системи.

3.1.3 Нефункціональні вимоги

Нефункціональні вимоги визначають як саме система повинна працювати й які загальні властивості мати.

Даний проєктований об'єкт має наступні нефункціональні вимоги:

- продуктивність й швидкість роботи (обробка нових свічок й відправка угод до брокера має займати не більше 100 мс);
- надійність (при відключенні від брокера, система повинна сама швидко перепідключатись без зупинки роботи системи);
- безпечність (шифрування облікових даних користувачів, контроль доступу);
- захищеність (система не зберігає дані про банківські картки після проведення транзакцій);
- інтуїтивно зрозумілий інтерфейс для користувачів;
- модульна архітектура для можливостей розширення системи;
- моніторинг й логування системи.

Отже, нефункціональні вимоги описують як саме повинна працювати системи й яким характеристикам відповідати.

3.2 Технічний стек

Технічний стек складається з основних компонентів за допомогою яких виконується реалізація інтелектуального торгового агенту.

Для створення проєктованого об'єкту використовується мова програмування Python, тому що вона є зрозумілою й гнучкою для реалізації будь-яких видів додатків й має сотні бібліотек, які облегшують процес створення систем.

В даній роботі основними бібліотека, що використовувались були:

- pandas (для обробки даних, підготовки DataFrame для моделей);
- numpy (для виконання векторних операцій, обчислення індикаторів, нормалізації даних й підготовки масивів для моделей машинного навчання);
- ufinance (для завантаження історичних котирувань з Yahoo Finance для бектестувань й проведення аналізу);

– ta-lib (для створення індикаторів й розробки модулів генерації ознак);

– scikit-learn (для швидкої реалізації алгоритмів машинного навчання).

Для бектестування й перевірки ефективності торгових рішень використовується фреймворк Backtrader, який визначає точки можливих входів в угоди на графіках різних валютних пар й таймфремів, які були підключенні за допомогою бібліотеки ufinance. Також Backtrader візуалізує оброблені графіки з історичними даними й може розрахувати різні характеристики якості проведеного бектестування.

Для виконання угод в режимі реального часу використовується фреймворк MetaTrader 5 Python Api, який потрібно власноруч під'єднати до середовища MT5 й обрати брокера через якого будуть виконуватись угоди. Підключення MT5 Api не є легким процес й більше про нього описано в розділі 4.4.

Для реалізації інтерфейсу користувача й налагодчика використовується GUI-фреймворк PyQt5, який є набором кросплатформних бібліотек за допомогою яких можна реалізувати високорівневі API для доступу до різних систем. PyQt5 є легким й гнучким для використання для створення різнопланових систем інтерфейсів, які будуть гармонічними й інтуїтивно зрозумілими для користувачів.

Для роботи з сховищами використовується фреймворк SQLite для управління базами даних, який дозволяє напряду з середовища розробки додатку легко управляти базами даних, за необхідністю змінювати чи додавати значення. А також Redis для кешування останніх котирувань у пам'яті системи для можливості моментального доступу й зниження затримок швидкості при виконанні угод в режимі реального часу.

Такий широкий набір інструментів забезпечує виконання необхідних вимог системи таких, як збір й обробка даних, побудова й тестування моделей алгоритмів, підключення й виконання угод за допомогою брокера,

проведення бектестувань, реалізацію інтерфейсу користувача, зберігання результатів в базах даних й стабільну роботу системи.

3.3 Використані методи, їх параметри й метрики оцінювання

3.3.1 Фільтр Гауса й налаштування параметрів

Після первинної обробки даних торгових агент отримує на вхід ціновий ряд з різними нестабільними значеннями. Тому на етапі препроцесінгу система застосовує фільтр Гауса до цінового ряду.

Фільтр Гауса – це технічний індикатор, побудований на основі однойменного методу машинного навчання, який використовує модель розподілення Гауса для того, щоб зменшити випадкові шуми в цінових даних, тим самим зробивши зміни трендів більш виразними [18].

Перед початком роботи фільтру Гауса треба налаштувати параметри: сігму й розмір вікна. Сігма – це кількість стандартних відхилень ядра, яка використовується для вибору більш ймовірних точок в даних. Також визначає ширину гаусівської кривої. Якщо значення сігми низьке, то графік буде трохи згладженим й зберігати незначні коливання, а якщо, навпаки, значення високе, то на графіку залишаться тільки довготривалі тренди. Також потрібно визначити розмір вікна, який дорівнює $2K+1$, щоб охопити все гаусівське ядро, приблизно використовується, що $K \approx 3\sigma$.

Після вибору параметрів відбувається процес побудови ядра. Для цього спочатку формується вектор ваг довжиною $2K+1$ як наведено у формулі 3.1. А потім відбувається його нормування так, щоб сума всіх ваг дорівнювала 1, що представлено у формулі 3.2.

$$G[k] = \exp\left(-\frac{k^2}{2\sigma^2}\right), \quad (3.1)$$

де $k = -K, \dots, 0, \dots, +K$.

$$G[k] = \frac{G[k]}{\sum_{j=-K}^K G[j]}. \quad (3.2)$$

Щоб коректно обробити початок й кінець цінового ряду, без створення викидів, розширюємо його на K точок з обох боків [19]. Далі на етапі згортки для кожного моменту t вираховуємо нове згладжене значення, як представлено у формулі 3.3:

$$\hat{p}_t = \sum_{k=-K}^K G[k] * p_{t+k}. \quad (3.3)$$

Після виконання згортки видаляємо додані точки, щоб отримати ціновий ряд тої самої довжини, як перед обробкою.

Фільтр Гауса, в порівняння з іншими видами фільтрації, забезпечує більш однорідну реакцію на зміну ціни, бо він є менш чуттєвим до непередбачуваного підвищення волатильності. Це допомагає зменшити частоту хибних сигналів.

Таким чином, налаштування параметрів фільтру Гауса є обов'язковим етапом перед початком його роботи. Погано підібрані параметри можуть навіть з хорошим алгоритмом зробити негативні результати. Зазвичай для різних валютних пар й таймфреймів підбір параметрів виконується окремо для більш точних результатів ефективності.

3.3.2 kNN й налаштування параметрів

Після використання фільтру Гауса, отримано згладжений ціновий ряд, з яким вже починає працювати kNN. Для торгового агента метод kNN використовується як первинний обробник цінового ряду для визначення подальшої тенденції руху. Використовуючи метод kNN стратегія виконує процес навчання на історичних даних, щоб надалі виконувати прогноз й аналізувати тенденції на реальних вхідних даних.

Ефективність методу kNN може залежати від використаної валютної пари й параметрів налаштування, таких як кількості сусідів й метрики відстані, що використовуються [20].

Значення k – це кількість ближніх сусідів, які аналізуються під час використання методу. Якщо значення k занадто мале, то метод буде занадто прискіпливим до шуму в ціновому ряді. Якщо, навпаки, занадто високе, то метод буде виконувати аналіз для точок, що знаходяться далеко й є менш актуальними.

Є багато різних способів, щоб обрахувати відстань між сусідами, але найбільш розповсюдженим є евклідова відстань (формула 3.4). Вона дозволяє виміряти подібність між поточним станом ринку (вектор ознак x) й історичними даними (вектор ознак y) [21]. Чим менше значення відстані, тим патерн ближчий, тому ймовірність схожості прогнозу вища.

$$d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}. \quad (3.4)$$

Процес роботи алгоритму kNN починається з побудови навчальної вибірки на історичних даних, які є вже згладженими після використання фільтру Гауса. Для кожної свічки формується вектор ознак, який містить ціну, відсоткові зміни й значення технічних індикаторів, які використовуються. Така сформована матриця зберігається в пам'яті агента, а прогноз, що робити купувати, продавати чи чекати, формується за майбутніми результатами руху ціни в наступні N свічок [21].

При появі нової свічки створюється поточний вектор ознак й вираховується евклідові відстані до всіх точок з навчальної вибірки. Після того як всі обраховані вектори відсортовані за зростанням відстані, обирається найближчий з k сусідів.

Якість ефективності роботи методу kNN також залежить від вибору параметрів. Але для всіх валютних пар й таймфреймів прогноз руху

цінового ряду даних, який оброблений фільтром Гауса й проаналізований за допомогою алгоритму kNN, є ефективним й має високий відсоток заробітку на виконанні угод. В 4 розділі буде підтверджено ефективність використання поєднання даних алгоритмів на практиці.

3.3.3 Метрики оцінювання ефективності роботи методів

Для перевірки ефективності прогнозів найчастіше використовуються такі метрики оцінювання: SNR, MSE, Accuracy, Precision, Recall, F1-score й затримка.

Signal-to-Noise Ratio (SNR) – це показник відношення спектру низьких й високих частот до й після виконання фільтрації. Чим вище значення, тим краще фільтр відсікає шум, залишаючи чіткі зрозумілі тенденції.

Mean Squared Error (MSE) – це середнє квадратичне відхилення між початковими й згладженими даними. Чим значення менше, тим краще. В ідеалі близьке до нуля.

Accuracy – це точність класифікації, який показує який відсоток правильних прогнозів kNN серед усіх здійснених класифікацій.

Precision – це точність позитивного прогнозу, який показує частку справжніх прогнозів на покупку серед усіх прогнозів, які вказували на купівлю. Precision є дуже важливою метрикою в трейдингу, бо потрібно максимально уникати хибних сигналів прогнозованих угод.

Recall – це повнота алгоритму й показує частку справжніх прогнозів на покупку, які модель визначила правильно, серед усіх й виконаних, й пропущених прогнозів купівлі. Вказує на скільки алгоритм не пропускає можливості входу в угоди.

F1-score – це середнє гармонійне між Precision й Recall, показує баланс між точністю й повнотою.

Latency – це затримка, яка вказує на скільки збільшилась швидкість обробки даних від отримання ціни свічки до завершення виконання

прогнозу. Затримка бути мінімальною, в іншому випадку, затримка може негативно вплинути на якість торгівлі.

Метрики оцінювання допомагають зрозуміти якість й ефективність алгоритмів, які використовуються. Також при створенні оптимального підходу виникає потреба збалансувати згладженість даних й прогнозів до швидкості й ефективності роботи торгового агента.

3.4 Вхідні дані

Для коректної роботи торговий агент повинен отримувати певні вхідні дані для початку роботи. Основними вхідними даними для торгового агента звичайно є валютні пари й таймфрейми, для яких буде відбуватись торгівля, й структура передачі інформації про свічки графіку. А також 3 бази даних, які зберігають важливу для системи інформацію.

В розділі 1.2 «Валютні пари» вже було описано важливість правильно вибору валютних пар й їх види. На даний момент агент підтримує роботу з 3 основних валютних пар, які мають високу ліквідність й при цьому низький спред, що є важливими факторами для успіху автоматичної торгівлі:

- EUR/USD (євро/долар США);
- GBP/USD (британський фунт/долар США);
- USD/JPY (долар США/японська єна).

В майбутньому кількість валютних пар, що використовується можна буде легко збільшити, але пріоритет вибору пар буде для мажорних пар, які гарантують доступ до історичних даних за довгий проміжок часу, підтримуються всіма брокерами, мають мінімальні затримки для торгівлі в реальному часі й зазвичай мають більш високу ліквідність й низькі спреди, що впливає на кількість угод в день, які виконуються, й мінімальні проковзування ціни.

Таймфрейм – це час за який формується одна свічка. Торговий агент повинен використовувати декілька таймфреймів для роботи з однією

валютною парою, бо, по-перше, ситуація, яка відбувається на нижчих таймфреймах зазвичай кардинально відрізняється від вищих; по-друге, для збільшення кількості угод торгівля відбувається відразу на декількох таймфреймах для однією валютної пари.

Найпопулярнішими таймфреймами й тими, які буде використовувати торговий агент, є:

- M1 (1-хвилинна свічка) надає максимальну деталізацію графіку;
- M5 (5-хвилинна свічка) є балансом між шумом й трендом;
- M15 (15-хвилинна свічка) є середньострокових для денних угод;
- H1 (1-годинна свічка) для виявлення довготривалих трендів.

Чим коротше таймфрейм, тим кількість угод, які виконуються є більшою, але далеко не завжди їх якість є хорошою. Саме тому є ефективним варіантом використання декількох таймфреймів для торгівлі однієї валютної пари. Хоча й для кожного таймфрейму кожної пари треба окремо завантажувати й обробляти торгівлю для поєднання угод.

Також на вхід до агенту під час виконання торгівлі передається інформація по кожній новій сформованій свічці. Дані зберігаються у DataFrame у пам'яті системи. Кожна свічка містить такі поля:

- timestamp (час в який розпочато роботу свічки, залежить від таймфрейму);
- open, high, low, close ціни (максимум/мінімум які ціна досягала під час створення свічки й ціна при якій почалось й закінчилось формування);
- volume (обсяг торгів на свічку);
- ask/bid (ціна продажу/покупки на момент закриття);
- spread (різниця між ask й bid).

Саме таку структуру DataFrame у вигляді цінового ряду отримує фільтр Гауса на вхід й далі виконується згладження.

Для коректної роботи системи й зберігання важливої інформації про користувачів й торгівлі в системі використовуються 3 бази даних.

Перша база даних «account.db» містить інформацію про акаунти користувачів й їх обранні параметри для торгівлі й має таку структуру:

- id;
- name (ПІБ користувача);
- email;
- password;
- pair (валютна пара, на якій буде відбуватись торгівля);
- tariff (тариф, це депозиту грошей, на які треба поповнити акаунт й за допомогою яких буде відбуватись торгівля);
- risk (відсоток ризик менеджменту на 1 угоду);
- role (визначається автоматично, в залежності від ролі користувача).

База даних «account.db» заповнюється при реєстрації акаунта в системі й її параметри можуть бути зміненні впродовж роботи.

Друга база даних «financial_data.db» потрібна тільки для того, щоб зберігати початковий баланс користувача й кінцевий після проведення торгівлі. БД містить інформації тільки про користувачів й має таку структуру:

- email (який є ідентифікатором);
- initial_balance (початковий баланс);
- current_balance (баланс після торгівлі).

БД потрібна для порівняння заробітку й для виведення кінцевої суми грошей з системи.

Третя база даних містить дані про виконані угоди в процесі реальної торгівлі для користувачів й має таку структуру:

- id;
- user (електрона пошта користувача для спрощення їх пошуку);
- symbol (валютна пара, на якій відбувалась торгівля);
- side (покупка/продаж);
- open_time й close_time (час початку й закінчення угоди);
- open_price й close_price (ціна відкриття й закриття угоди);

- lot (лот, розмір якого залежить від обраного ризик менеджменту);
- tp_price й sl_price (ціна миттєвого закриття угоди при позитивному чи негативному результаті);
- profit (заробіток/втрата з угоди).

Всі угоди, які виконуються під час торгівлі в реальному часі записуються в цю БД й надалі отриману інформацію буде зручно використовувати наладчикам для оцінки ефективності стратегії торгівлі.

Вхідні дані для вибору параметрів користувачів важливі для ефективності й коректності роботи системи. На основі вхідних даних торговий агент виконує їх обробку й аналіз, який описано в розділі 4.1.

3.5 Архітектура системи

Архітектура системи побудована за принципом модульності, який означає, що кожен компонент виконує чітко визначену роль. Такий підхід полегшує процеси оптимізації, модифікацій й тестування системи через те, що не потрібно буде змінювати весь код торгового агента, а буде достатньо переробити потрібний модуль.

Також в системі наявний інтерфейс окремий для користувачів й наладчиків. Інтерфейс користувач містить інформацію параметрів системи, можливість ввести й вивести гроші, а також функцію запуску й запинку роботи торгового агента, але при цьому жодних графіків чи сигналів користувач не бачить. Графіки, бектестування, аналітика проведення торгівлі – все перелічене є тільки для інтерфейсу наладчика.

Архітектура торгового агента складається з 5 основних модулів як представлено на рисунку 3.1. Перший модуль «Завантаження даних» потрібний для підключення зовнішніх джерел, завантаження історичних даних, підключення до MT5 Арі, а також має уніфікований інтерфейс `get_candels` для єдиного підключення й реалізації свічок у графіках.



Рисунок 3.1 – Дерево архітектури системи

Другий модуль «Обробка даних» відповідає за нормалізацію даних, застосовується фільтр Гауса й реалізується DataFrame з відповідними полями для генерації ознак. Даний модуль викликається кожного разу при надходженні нових свічок.

Третій модуль «Генерація сигналів» на вхід отримує згладжений ціновий ряд й на його основі формує вектор ознак й виконує kNN алгоритм на його основі. В результаті роботи отримується сигнал на покупку, продаж чи очікування.

На вхід четвертий модуль «Виконання угод» отримує сигнал kNN й викликає MT5 Api, передаючи йому сигнал й установлений Take Profit й Stop Loss. Після виконання угоди перевіряє її результат й вносить дані торгівлі в базу даних «Інформація про торгівлю». У разі виникнення проблем в підключенні до MT5 запускає алгоритм реконекту й повторює спробу під'єднання.

П'ятий модуль відповідає за інтерфейс користувача й налагодника. Цей модуль складається з 9 вікон торгового агента, які забезпечують повний

потрібний функціонал для користувача й налагодчика. Також UI формує графіки й відображення сигналів на них.

В архітектурі системи важливо продумати функціонал інтерфейсу користувача максимально закритим від непотрібних йому функцій, лишивши тільки основні. Для цього потрібно розробити UML-схему, яка буде наочно відображати можливі дії користувача в торговому агенті. Така схема представлена на рисунку А.1, який знаходиться в додатку А.

Перший етап 1.1 потрібен, щоб розпочати роботу користувача у системі. Для цього він повинен зареєструватись в системі увівши потрібні для цього параметри такі, як ПІБ, електронна пошта, придумати пароль, а також обрати особливі характеристики торгівлі: валютну пару, припустимий ризик менеджмент й тарифний план, а також натиснути відповідну (галочку), що користувач ознайомився з правилами користування системою.

На етапі 1.2 відбувається вхід у систему для цього потрібно ввести електронну пошту й пароль, які вводились на етапі реєстрації.

На наступному етапі 1.3 користувач повинен внести вибраний депозит для того, щоб система могла бути активованою. Перед початком роботи користувач може ознайомитись з вибраними параметрами торгівлі й якщо є потреба може їх змінити. Якщо ж користувача задовольняють параметри, то він натискає кнопку «Активувати систему», й якщо сума внесених грошей відповідає обраному тарифу, то торгівля розпочнеться.

Етап 1.4 потрібний для зупинку роботи торгівлі. Також на цьому етапі користувач може ознайомитись скільки грошей принесла йому торгівля й після зупинення торгівлі вивести зароблені кошти.

Далі відбувається етап 1.5, на якому користувач за потреби може змінити характеристики системи (валютну пару, тариф чи ризик менеджмент) на інші, чи обрати варіант повторної активації системи, в такому випадку користувач знову перейде до етапу 1.3. Етап 1.5 є фінальним

й у випадку, якщо користувач не бажає повторно активувати систему, то торговий агент завершує свою роботу.

Архітектура система є важливою складовою розробки системи, бо пояснює, які саме модулі будуть потрібні для подальшої реалізації. Модульність системи забезпечує можливість нескладної доробки чи зміни роботи певних алгоритмів й стратегій при потребі. Й проаналізувавши архітектуру системи, стає чітко зрозуміло, які саме модулі будуть потрібні для використання користувачем чи наладчиком. Це є важливим елементом для визначення вікон й інтерфейсів для користувача, сховавши всі непотрібні деталі системи й відкривши користувачу доступ до потрібних й зрозумілих дій.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТОРГОВОГО АГЕНТУ

4.1 Збір даних

Збір даних є першим важливим етапом перед початком процесу реалізації торгового агента. В розділі 3.4 вже було наведено аналіз бібліотек, які використовуються для ефективної роботи системи. В цьому розділі буде наведено практичні призначення й алгоритми збору даних, які використовуються.

До етапу збору даних входять наступні функції: збір ринкових даних для підключення до МТ5, для режиму бектестування, реальної торгівлі й демонстрації графіків з сигналами.

4.1.1 Збір даних для класу МТ5Connector

Клас МТ5Connector використовується для підключення до МТ5 Арі за допомогою, якого в подальшому буде відбуватись процес торгівлі в режимі реального часу. Код реалізації класу розміщено в додатку Б, лістинг Б.1.

Для підключення використовується офіційна Python-бібліотека MetaTrader5. Метод `mt5.initialize()` виконує ініціалізацію зв'язку з платформою МТ5, яка повинна бути увімкнена й підключена до відповідного брокера для налаштування зв'язку. В іншому випадку алгоритм вкажить на помилку початкового підключення. Метод `mt5.shutdown()` потрібний для коректного закриття сесії зв'язку без втрати особливих даних.

Алгоритм `_tf_constant` використовується для завантаження потрібного таймфрейму для різних валютних пар.

Алгоритм `get_historical` потрібний для завантаження певної кількості останніх свічок для вказаної валютної пари й таймфрейму. Основним методом є виклик `mt5.copy_rates_from_pos(symbol, timeframe, 0, bars)` для

отримання свічок починаючи з початкової позиції у вигляді DataFrame з конвертованим стовпцем часу.

Алгоритм `get_historical_range` завантажує усі свічки, які потрапляють у вказаний діапазон часу для певної валютної пари на заданому таймфреймі. Алгоритм працює за використання `mt5.copy_rates_range(symbol, timeframe, start, end)`, повертаючи DataFrame з конвертованим стовпцем часу. Цей алгоритм у торгового агенті використовується під час бектестування, при якому задається певний діапазон для аналізу торгівлі на історичних даних.

Алгоритм `get_symbol_info` повертає усі важливі технічні характеристики валютної пари, наприклад, мінімальний розмір кроку, лот, спред, тощо. Також алгоритм містить багато варіантів обробки помилки при виникненні різних ситуацій зв'язку з брокерами й MT5 при неможливості підключення до певної валютної пари. Алгоритм використовується й для реальної торгівлі, й для бектестування для роботи з характеристиками валютної пари, наприклад, для визначення SL й ТК.

Алгоритм `get_symbol_tick` повертає інформацію про останні котирування останньої ціни покупки (`ask`) й продажі (`bid`) для валютної пари. Використовується для торгівлі в реальному часі для точної передачі даних про угоди, їх відкриття й закриття.

Основна задача класу `MT5Connector` є підключення до MT5 Арі для торгівлі в реальному часі й збору потрібних ринкових даних.

4.1.2 Збір даних для класу `TradingEngine`

Клас `TradingEngine` це клас, який відповідає за реальну торгівлю й бектестування, саме через цей клас відправляються угоди до MT5, тому він потребує отримання даних для їх обробки для подальшої торгівлі. Код класу `TradingEngine` представлено у додатку В.

Метод `_run_loop` (лістинг В.4) використовується для торгівлі в реальному часі, тому потребує кожний період часу, в залежності від

таймфрейму, отримувати інформацію про мінімум й максимум свічки та ціни її відкриття й закриття. Метод `mt5.get_historical()` (лістинг 4.1) за кожної появи нової свічки повертає інформацію про певну кількість останніх свічок у вигляді `DataFrame`.

Лістинг 4.1 – Виклик `mt5.get_historical()`

```
df = self.mt5.get_historical(  
    symbol=self.symbol,  
    timeframe=self.timeframe,  
    bars=self.history_bars)
```

Метод `run_backtest` (лістинг В.4) створений для проведення бектестування, тому потребує даних про свічки валютну пари за певний період часу. Й використовує метод `mt5.get_historical_range()` (лістинг 4.2), який перед початком бектестування збирає вказані історичні дані у вигляді `DataFrame`.

Лістинг 4.2 – Виклик `mt5.get_historical_range()`

```
df = self.mt5.get_historical_range(  
    symbol = self.symbol,  
    timeframe = self.timeframe,  
    start = start,  
    end = end)
```

Отже, збір даних використовується як для реальної торгівлі, так і для проведення бектестування є потреба в даних про свічки. Різниця в тому, що при реальній торгівлі свічки передаються кожного разу при появі нової свічки для постійного отримання актуальної інформації про графік й формування на їх основі нових сигналів, а для бектестування перед його початком завантажуються історичні дані за обраний період часу й на їх основі визначаються можливі сигнали й ефективність стратегії.

4.2 Обробка даних

Після виконання завантаження даних цінових рядів, постає потреба в обробці даних. По-перше, звичайні методи стандартизації не підходять для обробки даних у трейдингу через те, що при таких нормалізаціях використовується стандартна формула при якій стираються й важливі, й не потрібні шуми й відхилення даних. Щоб цього уникнути в торговому агенті використовується фільтр Гауса (лістинг 4.3) для зменшення шуму в даних, але незводячи їх до прямої лінії ціни. Також для отримання більш плавного тренду використовується свічки Heiken Ashi замість звичайних.

Лістинг 4.3 – Реалізація фільтру Гауса

```
class GaussianFilter:
    def __init__(self, length=40, sigma=20):
        self.length = length
        self.sigma = sigma
        centers = np.arange(self.length)-(self.length-1)/2
        self.weights=np.exp(-0.5*(centers/self.sigma)** 2)
        self.weights /= self.weights.sum()
    def apply(self, series: pd.Series) -> pd.Series:
        data = series.values
        half = self.length // 2
        padded = np.pad(data, (half, half), mode='edge')
        filtered=np.convolve(padded,self.weights,
mode='valid')
        filtered = filtered[: len(data)]
        return pd.Series(filtered, index=series.index)
```

Метод `init` приймає стандартні дані довжина (кількість точок у фільтрі) дорівнює 40, а сигма (стандартне відхилення при розподілі Гауса) – 20. Ці дані за виконанням бектестування є оптимальними для торгівлі в реальному часі на обраних валютних парах. Далі виконується

нормалізація ваги фільтру, щоб гарантувати, що сумарний масштаб вихідного ряду не зміниться.

Метод `apply` на вхід отримує масив початкових даних цінового ряду. Через те, що кожна точка потрібна мати по 20 сусідів з обох боків, потрібно додати залишкові значення для всіх точок, яким їх не вистачає. Далі до збільшеного ряду застосовуємо фільтр Гауса з масивом ваг. На вихід потрібно подати масив з тією ж довжиною, тому сусіди, що були додані, обрізаються для повернення їх початкової кількості. Повертається результат як `pd.Series` з оригінальним індексом для збереження дати й часу свічок без втрати важливих даних.

Основна ціль фільтру Гауса це обережно, але ефективно згладження цінового ряду для усунення шумів даних, виявлення чіткого тренду й тим самим зменшення кількості хибних сигналів торгівлі.

На основі отриманого цінового ряду після застосування фільтру Гауса створюємо свічки в стилі `Heiken Ashi`.

Свічки `Heiken Ashi` – це вид свічок, який використовується для представлення й використання ринкових даних для прогнозування сигналів. Такі свічки використовують середні ціни, що допомагає ще більше зменшити ринковий шум й зробити тренди більш чіткими для сприйняття.

`Heiken Ashi` використовують особливий спосіб підрахунку свічок [22], який використовує дані про ціни відкриття й закриття за попередній період й дані про відкриття, мінімум, максимум й закриття (OHLC) за теперішній період. Розрахунки свічок наведені у формулі 4.1.

$$\left\{ \begin{array}{l} \text{Heiken Ashi Close} = \frac{\text{Open}_0 + \text{High}_0 + \text{Low}_0 + \text{Close}_0}{4} \\ \text{Heiken Ashi Open} = \frac{\text{HA Open}_{-1} + \text{HA Close}_{-1}}{2} \\ \text{Heiken Ashi High} = \max(\text{High}_0, \text{HA Open}_0, \text{HA Close}_0) \\ \text{Heiken Ashi Low} = \min(\text{Low}_0, \text{HA Open}_0, \text{HA Close}_0) \end{array} \right., \quad (4.1)$$

де HA – `Heiken Ashi`;

-1 – попередня свічка;

0 – поточна свічка.

HA Open це середня точка тіла попередньої свічки, а HA Close як середня ціна теперішньої свічки. HA High – це найвище значення поміж теперішнього максимуму, HA Open й HA Close. А HA Low, навпаки, мінімальне значення серед теперішнього мінімуму, HA Open й HA Close.

Кольори свічок залежать від тренду графіку: при нисхідному тренді червоні свічки, при висхідному – зелені.

Лістинг 4.4 – Вирахування Heiken Ashi

```
def compute_heiken_ashi(df: pd.DataFrame) -> pd.DataFrame:
    ha = df.copy()
    ha['HA_Close'] = df[['open', 'high', 'low',
    'close']].mean(axis=1)
    ha['HA_Open'] = (df['open'].shift(1) +
    df['close'].shift(1)) / 2
    ha.at[ha.index[0], 'HA_Open'] = (df.iloc[0]['open'] +
    df.iloc[0]['close']) / 2
    ha['HA_High'] = ha[['HA_Open', 'HA_Close',
    'high']].max(axis=1)
    ha['HA_Low'] = ha[['HA_Open', 'HA_Close', 'low'
    ]].min(axis=1)
    return ha
```

Функція `compute_heiken_ashi` (лістинг 4.4) перетворює звичайні свічкові дані у формат свічок Heiken Ashi. Спочатку створюється копія вхідного DataFrame й для всі нові стовпці будуть записуватись у копії, щоб не змінювати оригінал. Далі за формулами 4.1 вираховуються значення закриття, відкриття, мінімуму й максимуму ціни свічки й повертається новий DataFrame, у якому зберігаються й початкові стовпці про дані свічки й нові показники ціни свічок Heiken Ashi.

Свічки Heiken Ashi використовуються для усунення ще більшою кількістю шуму без втрати важливих даних й створення більш чіткого тренду. Генерація свічок Heiken Ashi за функцією `compute_heiken_ashi` використовується як для прогнозування сигналів торгівлі, так і для наочної візуалізації під час бектестів для чіткішого тренду.

4.3 Генерація сигналів

Генерація сигналів виконується з використанням обробленого цінового ряду, яке було проведено за допомогою фільтру Гауса й свічок Heiken Ashi, на основі стандартних технічних індикаторів й алгоритму kNN.

Насправді Heiken Ashi також відносять до технічних індикаторів, але вони напряду змінюють й усувають шуму з цінового ряду, а технічні індикатори SMA й RSI не впливають прямо на дані, а є допоміжними параметрами.

Проста ковзаюча середня (SMA) – це технічний індикатор, який вираховує просту середню за вказаний період [23]. SMA часто використовують для точного виявлення тренду, а також для згладжування цінових показників. Вираховується за формулою 4.2.

$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n}, \quad (4.2)$$

де A_n – ціна валютної пари в період n ;

n – загальна кількість періодів.

Індекс відносної сили (RSI) – це технічний індикатор, який використовується для визначення умов перекупленості чи перепроданості ринку [24]. RSI генерує імпульс, за яким визначається швидкість й волатильність руху ціни валютної пари. Значення RSI варіюється від 0 до 100 у вигляді лінійної діаграми. Значення вище 70 означають, що валютна пара в даний момент знаходиться в зоні перекупленості, а якщо значення

нижче 30, то в зоні перепроданості. В розрахунках RSI використовується параметр період, який по стандарту дорівнює 14, але при бажанні для кращих результатів його можна буде додатково налаштувати. Для кожного періоду RSI вираховується за формулою 4.3.

$$RSI_n = 100 - \frac{100}{1 + \frac{AvgProfit_n}{AvgLoss_n}} \quad (4.3)$$

де n – певний період формування свічки;

$AvgProfit_n$ – середній прибуток за період;

$AvgLoss_n$ – середній збиток за період.

Спосіб створення RSI й SMA наведений нижче за допомогою бібліотеки `ta-lib` й вбудованих до неї методів.

```
df['SMA'] = ta.SMA(df['HA_Close'].values, timeperiod=200)
df['RSI'] = ta.RSI(df['HA_Close'], timeperiod=14)
```

Перед використання методу `kNN` потрібно сформувані з даних, які отримано (згладжений фільтром Гауса ціновий ряд, ціна закриття й відкриття `Heiken Ashi`, значення RSI й SMA) вектор ознак для кожної свічки. Формування вектору фіксованої довжини є важливою складовою для методу `kNN`, щоб процес обчислення відстані до сусідів відбувався швидше. Також масив векторів є повноцінним описом поточного стану ринку й трендів валютної пари.

Лістинг 4.6 – Функція для формування вектору ознак й виконання `kNN` методу

```
class KNNTrader:
    def __init__(self, n_neighbors = 3):
        self.model=
KNeighborsClassifier(n_neighbors=n_neighbors)
    def fit(self, X, y):
        self.model.fit(X, y)
    def predict(self, X)
```

Продовження лістингу 4.6

```

        return self.model.predict(X)

    def build_feature_vector(df: pd.DataFrame, idx: int,
history: int = 5) -> np.ndarray:
        cols = ['smoothed_close', 'RSI', 'SMA', 'HA_Open',
'HA_Close']
        start = idx - history + 1
        if start < 0:
            return None
        segments = []
        for col in cols:
            segment = df.iloc[start:idx+1][col].values
            segments.append(segment)
        return np.concatenate(segments)

```

Лістинг 4.6 містить функцію `compute_feature_vector` саме вона відповідає за формування вектору ознак. Сенс функції це отримати потрібні дані для вектору й записати їх в одному місці для спрощення їх подальшого використання. Параметри `idx` – це індекс, який вказує на поточну свічку в `DataFrame`, а `history` – це параметр, який вказує за який період використовуються значення індикаторів.

Клас `KNNTrader`, який наведено у лістинг 4.6, описує роботу методу `kNN`, використовуючи бібліотеку `sklearn` й функцію `kNeighborsClassifier` для спрощеної реалізації методу `kNN`.

Метод `init` приймає параметр `n_neighbours`, який визначає кількість сусідів, які будуть враховуватись під час класифікації, й створює внутрішній об'єкт, який зберігає тренувальні приклади й алгоритми пошуку у просторі ознак.

Метод `fit` приймає дві матриці `X` й `y`. `X` – це матриця ознак, а `y` – вектор міток, розмір якого відповідає кількості рядків у матриці `X`. Й далі викликається стандартна функція для збереження даних в алгоритмі.

Метод `predict` приймає нові вектори ознак X . Викликає стандартну функцію для пошуку для кожного вектору з матриці X найближчих k прикладів у тренувальних даних й повертає масив прогнозованих міток, розмір якого відповідає кількості рядків у X .

Отже, процес генерація сигналів виконується наступним чином: при надходженні нової свічки, формується новий відповідний вектор ознак, викликається функція `knn.predict` й повертається сигнал купувати, продавати чи чекати.

Таким чином, метод `kNN` дозволяє швидко виконати прогнозування за вектором ознак при цьому використовуючи тренувальні дані ситуацій, які вже відбувались на ринку, й зрозуміти за якими саме сусідами було обрано варіант прогнозування.

4.4 Виконання угод

Після використання `kNN` класифікатору стає визначено приблизний напрямок руху графіку й починається один з найважливіших етапів роботи торгового агента це створення й передачу угод на виконання до MT5.

Основа функція для виконання угод це поєднання стратегії, описаної в проєкті, в функцію `generate_signals` (лістинг Б.2), яка на вхід отримує `DataFrame` з ціновим рядом й особливі налаштування торгівлі. Спочатку відбувається базове очищення даних, вираховуються свічки Heiken Ashi.

Далі застосовується фільтр Гауса до ціни закриття усіх свічок Heiken Ashi й результат зберігається до стовпця [`'Smoothed'`]. Відбувається розрахування класифікатору `kNN` для отриманого згладженого стовпця, результат якого буде записуватись до стовпця [`'kNN_MA'`].

На основі нових даних й будуть формуватись сигнали. Існує 3 шляхи прогнозувань: купівля (+1), продаж (-1) чи очікування (0). У випадку якщо поточна ціна закриття вище за попередню, згладжена ціна зростає й `kNN-MA` збільшується, то відбувається сигнал на купівлю. Якщо, всі дії

відбуваються абсолютно навпаки, ціна закриття нижче за попередню, згладжена ціна спадає, а kNN-MA зменшується, то виконується сигнал на продаж. В будь-якому іншому випадку сигнал не на купівлю, не на продаж не створюється, а очікується наступна свічка й процедура генерації сигналів повторюється.

Далі додаються стовпці, які відповідають за Stop Loss й Take Profit, які вимірюються у піпсах. Й в результаті повертається оновлений набір даних, який доповнений новими розрахованими стовпцями й переліком згенерованих сигналів.

Саме за допомогою цієї функції відбувається прогнозування й поєднання всіх вище описаних класів для отримання сигналу на купівлю, продаж чи очікування.

Наступний клас TradeExecutor, який відповідає за відправку угод до MT5 й їх запису до бази даних для можливості подальшої аналітики. Програмний код класу TradeExecutor наведено у додатку Б, лістинг Б.3.

Базовою функцією класу TradeExecutor є процес ініціалізації, який приймає на вхід об'єкт, під'єднаний до брокера в MT5 й відриває таблицю для зберігання історії угод.

Ключовим методом є `send_order`, який приймає на вхід параметри майбутньої угоди, такі як валютну пару, тип угоди: купити чи продати, поточна ціна, кількість лоту, ціну виходу з угоди при позитивному чи негативному результаті (SL й TK). Спершу відбувається перевірка наявності у брокера заданої валютної пари й додаткове налаштування Stop Loss й Take Profit для закінчення угоди. Основний етап це формування `request`, який містить всі потрібні параметри для угоди й відправка сформованої угоди до MT5 на виконання.

Далі в класі TradeExecutor прописані методи для формування таблиці для зберігання історії торгівлі й правильно запису відтворених угод.

Метод `_create_trade_logs_db` відповідає за створення таблиця для зберігання угод, більше про базу даних було описано в розділі 3.4. Методи

`log_open_trade` й `log_close_trade` відповідають за правильне логування угод при відкритті й закритті відповідно. При відкритті позиції в базу даних записується основна інформація про угоду: користувач для якого виконається угода, валютна пара, купити чи продати, лот, встановленні SL й TK, а також час й ціну відкриття угоди. Це базові дані про угоду, які не зміняться протягом виконання торгівлі цієї угоди. При закритті записуються такі дані: час й ціну закриття угоди й заробіток чи збиток, який принесла угода. Кожна угода містить свій унікальний ідентифікатор, за яким процес логування відкриття й закриття угод поєднується.

Наступні методи `open_long` й `close_log` і `open_short` й `close_short` слугують для створення, відправки й запису до бази даних угод. Функції `open_long` й `open_short` відправляють угоди на купівлю й продаж відповідно й викликають `log_open_trade` для запису параметрів угоди при її відкритті. Функції `close_long` й `close_short` спочатку шукають відкриту угоду, яку потрібно закрити при досягненні умов виходу з позиції, й закривають її. Вираховують профіт й записують угоду, викликаючи функцію `log_close_trade`.

Логіка класу `TradeExecutor` відповідає за відправку потрібних угод до брокера й запису кожної угоди, що гарантує, що всі позиції будуть зафіксовані й доступні в базі даних для історії аналізу торгівлі.

Клас `TradingEngine` – це головний клас торгового агента, який збирає в єдине циклічне завантаження даних, запуск стратегії й отримання сигналів як для торгівлі в реальному часі, так і для бектестування. Програмний код класу містяться у додатку Б, лістинг Б.4.

При ініціалізації класу приймаються на вхід параметри, які відповідають за основні характеристики торгівлі, виконується підключення до MT5.

Методи `start` й `stop` мають базову функцію початку й зупинки роботи системи відповідно, й зберігають інформацію про статус системи в даний момент часу.

Метод `_run_loop` – це метод, який відповідає за виконання торгівлі в режимі реального часу. Він циклічно аналізує ринок при появі нових свічок, створює сигнали, відправляє їх до брокера й відкриває та закриває прогнозовані угоди.

Спочатку метод `_run_loop` завантажується історичні дані для обраної валютної пари й таймфрейму. Далі відбувається генерація сигналів за допомогою методу `generate_signals` (лістинг Б.2). Значення `row` – це останній сформований сигнал й далі зберігаємо його параметри: ціну закриття, `stop loss` й `take profit`. Перевіряємо чи не відбулося змінити сигналу від останнього зафіксованого, якщо ні то виконується перевірка значення поточного сигналу. Якщо значення `sig` дорівнює 1, то відкриваємо угоду на купівлю, а якщо значення -1, то угоду на продаж. Аналогічно, якщо значення 2, то закриваємо угоду на купівлю, й навпаки, якщо значення -2, то закриваємо угоду на продаж. Й останнє це очікування інтервалу перед прогнозування наступної угоди. Очікування потрібне для того, щоб не відбувалось більше однієї угоди на одну свічку, одне прогнозована ситуація дорівнює одній угоді.

Метод `run_backtest` – це процес бектестування, тобто симуляції реальної торгівлі на історичних даних, який потрібен для аналізу ефективності й якості стратегії з можливістю підбору значень параметрів й валютних пар чи таймфреймів для виявлення тих, які надають найкращі результати.

Спочатку завантажується історичний діапазон в проміжку від вказаних користувач початкової й кінцевої дати. Наступний етапом є генерація сигналів за вказаними користувачем параметрами на відповідному діапазоні. Потім підготовлює важливі дані для запису торгівлі. Й починається цикл перебору сигналів. Якщо `sig` дорівнює 1 й позиції дорівнює 0, що означає, що на даний момент відкриті позиції відсутні, то виконується угода на купівлю. Якщо ж позиція дорівнює 1, то починається вихід з позиції купівлі. При спрацюванні `Take Profit` чи `Stop Loss`,

перевіряємо яке з них ближче до поточного значення ціни. Далі вираховується різницю між ціною входу й виходу з позиції, а також прибуток чи витрати від виконаної угоди, й додається запис в список виконаних угод. Вхід й вихід з позиції на продаж відбувається аналогічно, тільки тепер перевірка, що sig дорівнює -1 й шлях вниз. Також треба зауважити, що у випадку купівлі Take Profit повинен знаходитись вище за ціну входу в позицію, а Stop Loss нижче, а при продажі навпаки. Це важливий момент, який потрібно враховувати під час реалізації досягання ціною ТК чи SL й вирахування різниці між цінами.

Клас TradingEngine це основний механізм торговго агенту, який поєднує в собі важливі реалізовані методи для виконання торгівлі в режимі реального часу й бектестувань.

4.5 Інтерфейси користувача

Реалізація UI/UX інтерфейсів для користувача й наладчика є важливими складовими для створення інтуїтивно-зрозумілої й ефективної системи торгового агенту.

Початковим завданням було проектування прототипів інтерфейсів. А потім реалізація програмного коду й з'єднання вікон з методами й класами, які виконують торгівлю. Всі реалізовані вікна системи наведені у додатку Г, а в цьому розділі описуються прототипи вікон, їх основна задача, й порівняння з реалізованими.

Більшість систем починаються з вікна реєстрації користувача й торговий агент не є виключенням. Прототип інтерфейсу представлено на рисунку 4.1, з якого в подальшому було створено вікно в торговому агенті, візуальний вигляд якого знаходиться на рисунку В.1.

Назва (№1.1.1) вікна «Реєстрація акаунта» вирівняно по центру. Інформаційний блок (№1.1.2), в якому знаходять поля, які потрібно заповнити для створення акаунту. Поле для вводу ПІБ (№1.1.3),

email (№1.1.4), пароля (№1.1.5). Поля для вибору валютної пари (№1.1.6), тарифного плану (№1.1.7), ризик менеджменту (№1.1.8). Елемент управління галочка (№1.1.9) для погодження з правилами системи, поки галочка не натиснута, кнопка (№1.1.11) недоступна. Елементи управління кнопки для входу в акаунт (№1.1.10) (при наявності зареєстрованого акаунта) й для реєстрації акаунта (№1.1.11) розміщені внизу вікна, при натисканні кнопки (№1.1.11) відбувається збереження даних про акаунт в базу даних й перехід до вікна «Вхід в акаунт». У випадку, якщо всі поля заповнені правильно, то система повідомить про успішну реєстрацію акаунту, в іншому випадку видасть повідомлення про наявність помилок у введених даних. При натисканні кнопки (№1.1.10) відбувається тільки перехід до вікна «Вхід в акаунт».

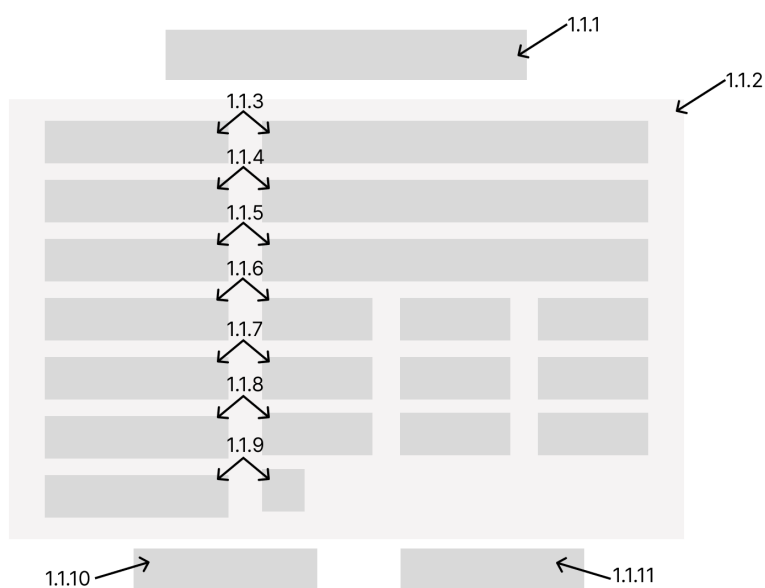


Рисунок 4.1 – Прототип вікна реєстрації

Спроектований інтерфейс повністю відповідає реалізованому, тільки доповнений наявністю підказок, що саме потрібно вводити в поля при наведенні курсору на поле для вводу.

Наступне вікно відповідає за вхід користувача в систему. Прототип вікна розміщений на рисунку 4.2, а реалізований варіант – на рисунку В.2.

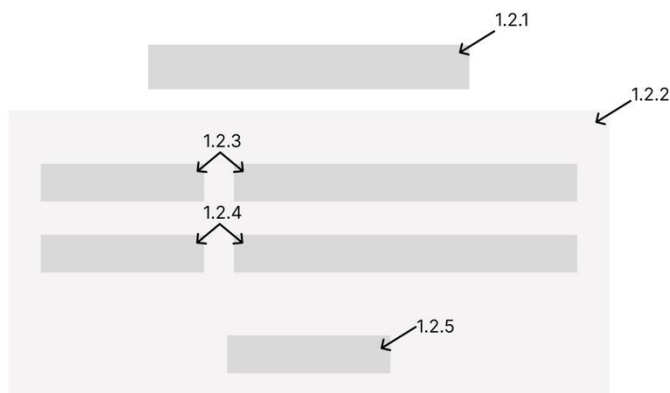


Рисунок 4.2 – Прототип вікна входу в акаунт

Назва (№1.2.1) вікна «Вхід в акаунт» вирівняно по центру. Інформаційний блок (№1.2.2), в якому знаходять поля, які потрібно заповнити для входу в акаунт. Поле для вводу email (№1.2.3) й пароллю (№1.2.4). Елемент управління кнопка (№1.2.5), яка здійснює вхід в акаунт, розміщена внизу вікна й вирівняна по центру. При натисканні цієї кнопки відбувається перехід до вікна в залежності від типу клієнта: користувача чи налагодчика. Користувач переходить до вікна «Активация системи», а налагодчика до вікна «Інформація про систему». Також вікно доповнено наявністю підказок при наведенні курсору на поле вводу. При правильному вводі даних система видасть повідомлення про успішний вхід в акаунт, в іншому випадку про неправильно введені електронну адресу чи пароль.

З практичної точки зору спроектована система повністю відповідає реалізованому варіанту, різниця полягає тільки у трохи зміненому візуальному вигляді вікна для кращого сприйняття інтерфейсу.

Наступне вікно для роботи користувача це вікно для активації системи. Реалізований вигляд вікна представлений на рисунку В.3, а прототип – на рисунку 4.3.

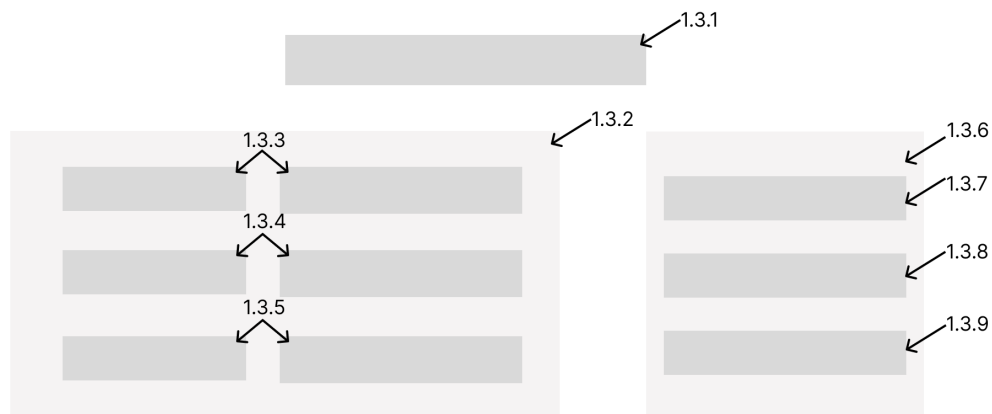


Рисунок 4.3 – Прототип вікна активації системи

Назва (№1.3.1) вікна «Активация системи» вирівняна по центру. Інформаційний блок (№1.3.2), в якому знаходиться параметри, які були обрані користувачем для роботи системи. Обрані параметри такі, як валютна пара (№1.3.3), тарифний план (№1.3.4) й ризик менеджмент (№1.3.5). Блок (№1.3.6) для елементів управління. Кнопка (№1.3.7) для внеску грошей, відповідно до обраного тарифного плану, й активації кнопки (№1.3.8), яка потрібна для запуску торгівлі й переносу до вікна зупинки системи. Кнопка (№1.3.9) потрібна, щоб змінити параметри системи, якщо в цьому є потреба.

У реалізованому варіанті вікна зміни відбулись з точки зору візуального вигляду вікна й доповнення інформаційного блоку даними про поточну дату й час й електронною адресою користувача, якому належить акаунт.

Наступне вікно потрібне для зупинки роботи системи. Поки користувач не вирішить натиснути кнопку зупинити торгівлю, вона буде

продовжуватись. Прототип вікна наведено на рисунку 4.4, а реалізований варіант – на рисунку В.4.

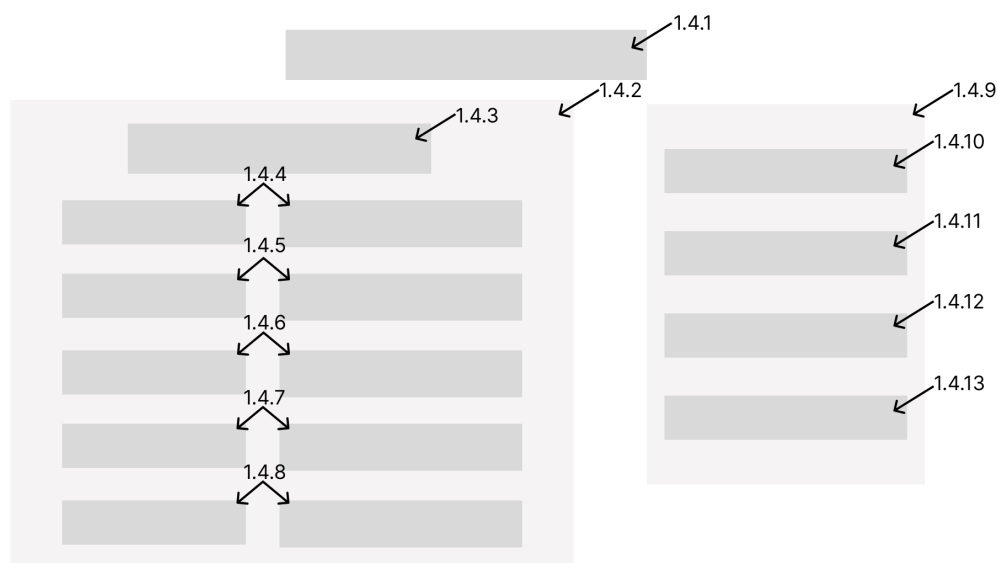


Рисунок 4.4 – Прототип вікна для зупинки торгівлі

Назва (№1.4.1) вікна вирівняна по центру. Інформаційний блок (№1.4.3), який містить інформацію про статистику системи. Назва блоку (№1.4.3) «Статистика системи» вирівняна по центру блока. Інформаційні поля, які містяться в блоку, такі, як електронна адреса користувача (№1.4.4), початкова сума депозиту (№1.4.5), теперішня сума депозиту (№1.4.6), зароблені кошти (№1.4.7) й відсоток зароблених грошей від початкової суми (№1.4.8). Блок (№1.4.9) для елементів управління. Кнопка (№1.4.10) для зупинки системи, після натиснення якої кнопка (№1.4.11) стає доступною. Кнопка (№1.4.11) для виводу зароблених грошей. Кнопка (№1.4.12) для зміни параметрів стратегії й при натисканні відбувається перехід до вікна для зміни параметрів. Кнопка (№1.4.13) потрібна для переходу до вікна активації системи, для повторної активації системи при потребі.

Єдина між реалізованим вікном і його прототипом це видалення елемента (№1.4.3) заголовку інформаційного блоку. Усі інші параметри й кнопки реалізовані за прототипом.

Останнє вікно користувача – це інтерфейс для зміни параметрів системи. Прототип представлено на рисунку 4.5, а реалізований варіант – на рисунку В.5.

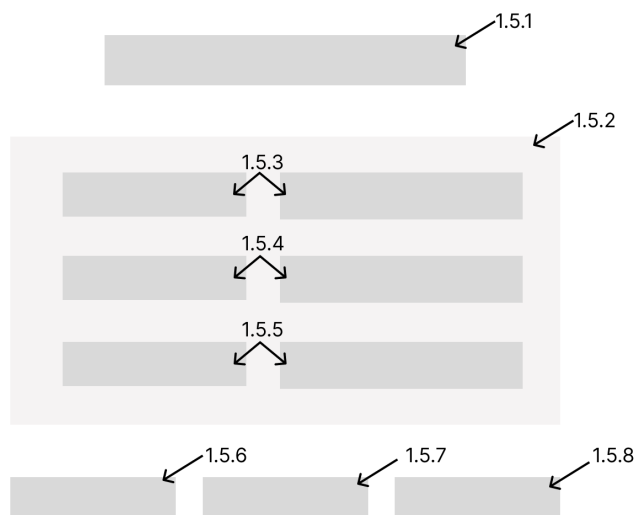


Рисунок 4.5 – Прототип вікна для зміни параметрів торгівлі

Назва (№1.5.1) вікна «Зміна параметрів системи» вирівняна по центру. Інформаційний блок (№1.5.2), в якому знаходяться параметри, які можна змінити. Поля для вибору валютної пари (№1.5.3), тарифного плану (№1.5.4) й ризик менеджменту (№1.5.5). Під інформаційним блоком розміщено три кнопки. Перша кнопка (№1.5.6) потрібна для скидування змінених параметрів до початкових, які були до початку зміни параметрів й не були збереженні. Друга кнопка (№1.5.7) для збереження змінених параметрів. Третя кнопка потрібна для повернення до вікна активації системи.

Отже, система має 2 вікна, які належать й користувачам, й налагодчикам, це вікна реєстрації й входу в систему, 3 вікна, які доступні тільки користувачам, це вікна активації, зупинки й зміни параметрів, а також 4 вікна, які доступні тільки налагодчикам.

Перше вікно, яке бачить наладчик при вході в систему це вікно «Інформація системи». Прототип вікна наведено на рисунку 4.6, а реалізований варіант – на рисунку В.6.

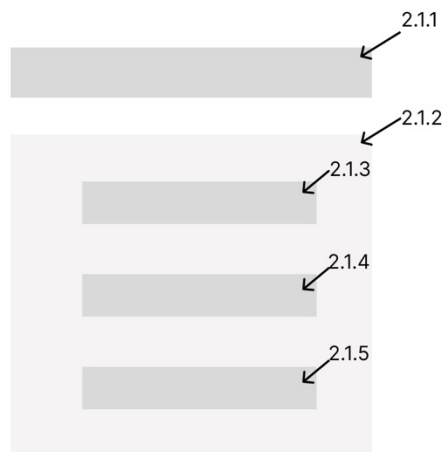


Рисунок 4.6 – Прототип вікна інформації системи для наладчика

Назва (№2.1.1) вікна «Інформація про систему» вирівняна по центру. Блок (№2.1.2), який містить 3 кнопки: кнопка (№2.1.3) для переходу до вікна бектестувань, кнопка (№2.1.4) для переходу до вікна статистики торгівлі в реальному часі, кнопка (№2.1.5) для виведення зароблених системою грошей з відсотку, які платить користувач за користування системою.

Інтерфейс був реалізований за прототипом, за виключенням того, що прибрано візуально блок (№2.1.2) для більшої наочності.

Наступне вікно потрібне для вибору параметрів для початку бектестування. Реалізований інтерфейс вікна продемонстровано на рисунку В.7, а прототип – на рисунку 4.7.

Назва (№2.2.1) вікна «Бектестування» вирівняна по центру. Інформаційний блок (№2.2.2), в якому знаходяться параметри, які потрібні для проведення бектестування. Блок (№2.2.2) містить наступні поля для вибору валютної пари (№2.2.3), таймфрейму (№2.2.4), лоту (№2.2.5), дата початку (№2.2.6), дата закінчення (№2.2.7), Take Profit (№2.2.8), Stop

Loss (№2.2.9), довжина фільтру Гауса (№2.2.10), сигма Гауса (№2.2.11), розмір вікна (№2.2.12), період ATR (№2.2.13). Під інформаційним блоком розміщені дві кнопки: кнопка (№2.2.14) для початку бектестувань й переходу до вікна результатів бектестувань й кнопка (№2.2.15) для повернення до вікна «Інформація про систему»

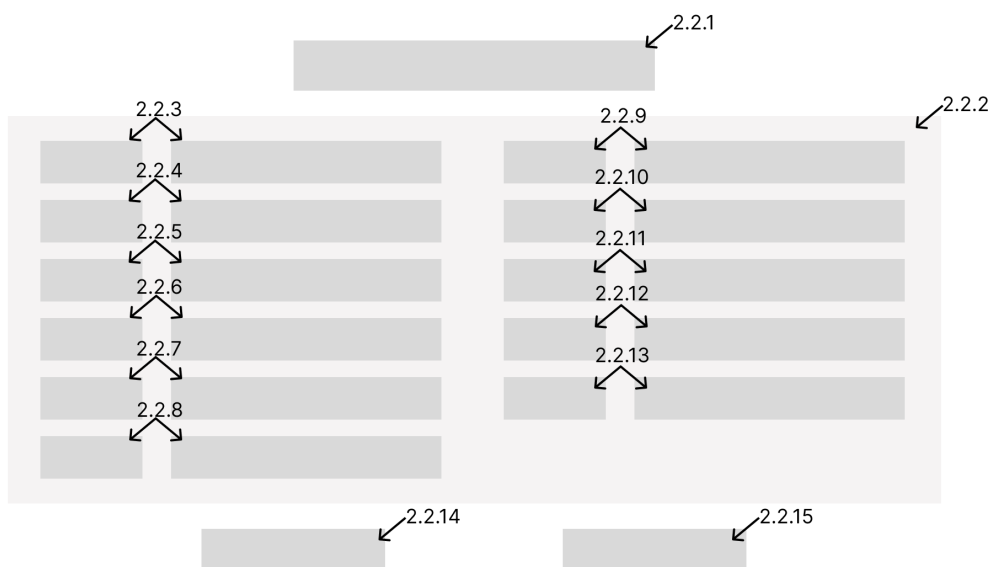


Рисунок 4.7 – Прототип вікна для вибору параметрів бектестування

В вихідному варіанті реалізованого вікна було змінено візуальну складову інтерфейсу, але практичну, кнопки й поля для вводу, залишено, як на прототипі.

Після вибору параметрів для бектестування відбувається перехід до вікна результатів бектестування. Реалізований варіант розміщено на рисунку В.8, а прототип – на рисунку 4.8.

Назва (№2.3.1) вікна «Результати бектестування» вирівняна по центру. Інтерактивна картинка (№2.3.2) демонструє графік обраної валютної пари з таймфреймом за певний проміжок часу. На графіку відображено угоди на продаж й купівлю, які були зроблені. Поле (№2.3.3) містить інформацію про прибуток, який було зароблено впродовж

бектестування. Таблиця (№2.3.4) містить інформацію про угоди, які були виконанні за бектестування, й такі стовпці: дата й час початку й закінчення угоди, продаж чи купівля, ціна входу й виходу з позиції й прибуток отриманий за одну угоду. Реалізований варіант повністю повторює прототипом.

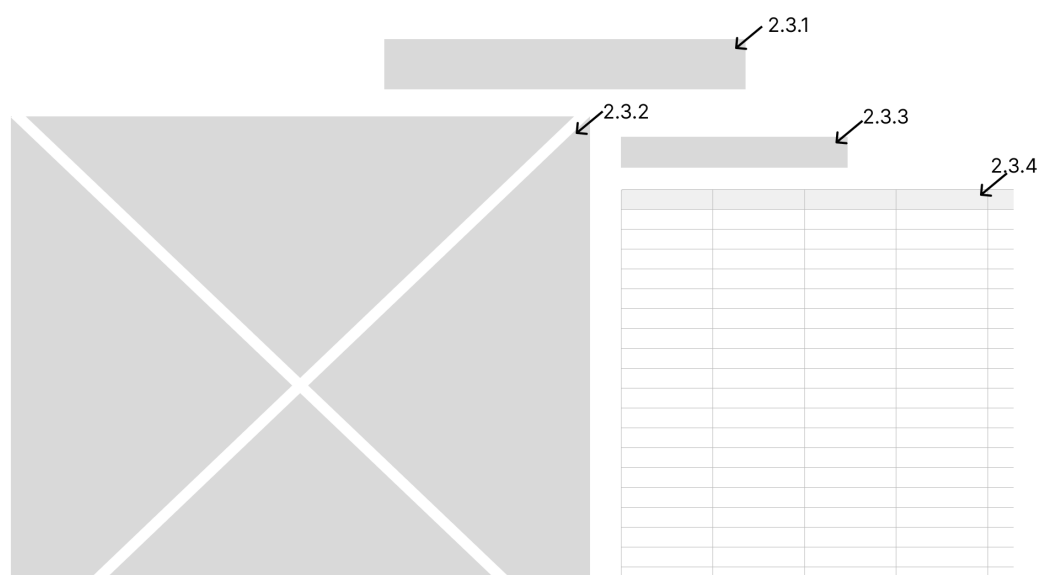


Рисунок 4.8 – Прототип вікна результатів бектестувань

Реалізований варіант повністю відповідає прототипу й за візуальним, й за технічним аспектами..

Останнє вікно, яке містить торговий агент, це вікно з результатами торгівлі в реальному часі обраного користувача. Реалізований варіант розміщено на рисунку В.9, а прототип – на рисунку 4.9.

Назва (№2.4.1) вікна «Результати live-торгівлі» вирівняна по центру. Поле для вибору (№2.4.2) користувача, чию статистику буде показано. Кнопка (№2.4.3) для повторного завантаження списку користувачів. Кнопка (№2.4.4) потрібна для показування статистики угод. Таблиця (№2.4.5) містить статистику угод, які були виконанні для обраного користувача за час проведення торгівлі. Таблиця повинна містити такі