

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система прогнозування фінансового стану ринку на основі історичних даних і патернів з використанням технологій штучного інтелекту»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
освітньо-професійної програми Штучний інтелект

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Данило КОЛОТЕНКО
(підпис)

Виконав: здобувач вищої освіти гр.ШІД-41
Данило КОЛОТЕНКО

Керівник: Євген ЧИЧКАРЬОВ
д.т.н, професор

Рецензент: _____

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Штучного інтелекту

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедру Штучного інтелекту

_____ Ольга ЗІНЧЕНКО

«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Колотенку Данилу Олексійовичу

1. Тема кваліфікаційної роботи: Проектування системи для прогнозування фінансового ринку на основі історичних даних і патернів з використанням технологій штучного інтелекту на PHP

керівник кваліфікаційної роботи д.т.н., професор, Євген Чичкарьов
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» 02.2025р. № 56

2. Строк подання кваліфікаційної роботи «23» травня 2025р.

3. Вихідні дані до кваліфікаційної роботи: Історичні фінансові дані з API Polygon.io, параметри технічних індикаторів (RSI, EMA), структура вебзастосунків на PHP, вимоги до інтеграції AI-моделей, науково-технічна література

4. Зміст розрахунково-пояснювальної записки

1. Огляд сучасних технологій прогнозування фінансових ринків
2. Аналіз технічних індикаторів (RSI, EMA) та ринкових патернів
3. Отримання та обробка історичних біржових даних з API Polygon.io
4. Розробка архітектури інтелектуальної системи прогнозування на PHP

5. Інтеграція моделі машинного навчання у вебзастосунок
 6. Тестування точності прогнозу та порівняння з реальними даними
 7. Візуалізація результатів через вебінтерфейс
 8. Аналіз ефективності системи, обмеження та перспективи її застосування
5. Перелік графічного матеріалу: *презентація*
6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	27.02-05.11.25	Виконано
2	Вивчення джерел та API (Polygon.io) для отримання фінансових історичних даних, формування датасету	05.11-12.11.25	Виконано
3	Дослідження існуючих систем прогнозування, порівняльний аналіз технологій, постановка задачі	13.11-19.11.25	Виконано
4	Розробка структури вебдодатку на PHP: збір даних, обробка, зберігання, візуалізація	20.11-25.11.25	Виконано
5	Інтеграція моделей машинного навчання: тренування, тестування, підключення до вебдодатку	27.11-03.12.25	Виконано
6	Перевірка функціональності, тестування точності моделі, аналіз результатів, оптимізація	04.12-10.12.25	Виконано
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.25	Виконано
8	Розробка демонстраційних матеріалів	21.12-31.05.25	Виконано

Здобувач вищої освіти _____
(підпис)

Данило КОЛОТЕНКО

Керівник

кваліфікаційної роботи _____
(підпис)

Євген ЧИЧКАРЬОВ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: стор.64, рис.6, таблиць 1, джерел 21.

Мета роботи: Розробка системи для прогнозування фінансового ринку на основі історичних даних і патернів з використанням технологій штучного інтелекту засобами мови PHP.

Об'єкт дослідження: Процес аналізу історичних фінансових даних з метою прогнозування майбутньої динаміки ринку.

Предмет дослідження: Інтелектуальна система прогнозування фінансових ринків, яка поєднує аналіз патернів, технічних індикаторів (RSI, EMA) та методи штучного інтелекту, реалізована за допомогою PHP.

Короткий зміст роботи: У кваліфікаційній роботі проаналізовано сучасні методи прогнозування фінансових ринків, зокрема методи технічного аналізу та застосування алгоритмів машинного навчання. Розглянуто ключові патерни та індикатори (RSI, EMA), які використовуються для виявлення потенційних точок входу/виходу. Було обґрунтовано доцільність використання нейромереж для прогнозування цінових змін на основі історичних даних. Реалізовано систему на основі мови програмування PHP із візуалізацією даних за допомогою JavaScript-графіків. Передбачено веб-інтерфейс для взаємодії з системою, підключення до зовнішнього API з фінансовими даними та модуль машинного навчання для генерації прогнозів. Проведено тестування моделі та аналіз її точності, що підтвердило ефективність розробленого підходу для практичного застосування у сфері фінансового аналізу.

КЛЮЧОВІ СЛОВА: ФІНАНСОВИЙ РИНОК, ПРОГНОЗУВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ПАТЕРНИ, RSI, EMA, PHP, МАШИННЕ НАВЧАННЯ, ПРОГНОЗ, API.

ЗМІСТ

ВСТУП.....	7
1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ФІНАНСОВОГО РИНКУ З ВИКОРИСТАННЯМ AI.....	11
1.1 Аналіз методів прогнозування фінансових ринків.....	11
1.2 Патерни в аналізі ринку: визначення та роль у прогнозуванні.....	16
1.3 Застосування технологій штучного інтелекту в фінансовому аналізі.....	20
Висновки до розділу 1.....	26
2 АНАЛІЗ ДАНИХ І ПОСТАНОВКА ЗАДАЧІ ПРОГНОЗУВАННЯ З ВИКОРИСТАННЯМ ВЕБ-ІНТЕРФЕЙСУ.....	27
2.1 Вибір та обробка історичних фінансових даних.....	27
2.2 Побудова індикаторів (RSI, EMA) та виявлення патернів.....	32
2.3 Розробка серверної частини системи прогнозування на PHP.....	36
Висновок до розділу 2.....	42
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДУЛЯ ІНТЕЛЕКТУАЛЬНОГО ПРОГНОЗУВАННЯ.....	44
3.1 Алгоритм прогнозування з використанням AI-моделі.....	44
3.2 Веб-інтерфейс системи прогнозування.....	49
3.3 Тестування та оцінка точності прогнозу.....	55
Висновок до розділу 3.....	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ПОСИЛАНЬ.....	64
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ).....	65

ВСТУП

У сучасному фінансовому середовищі, що характеризується високою динамікою, інформаційним перевантаженням та постійними ризиками, здатність точно передбачати майбутній стан ринку є надзвичайно важливою. Ця необхідність стосується не лише великих інституційних гравців, а й приватних інвесторів, аналітиків і розробників торгових стратегій. Традиційні методи прогнозування, які ґрунтуються виключно на технічному аналізі або фундаментальних індикаторах, втрачають ефективність через зростання складності взаємозв'язків у фінансових даних. У цьому контексті штучний інтелект, зокрема машинне навчання, відкриває нові горизонти: аналізуючи великі обсяги історичної інформації, моделі можуть виявляти складні патерни, які недоступні людському окомірному спостереженню. Особливо перспективною є інтеграція таких алгоритмів у веборієнтовані системи, зокрема з використанням мови РНР, що дозволяє реалізувати доступні, масштабовані та інтерактивні рішення, адаптовані для широких користувачів.

Актуальність теми зумовлена необхідністю створення сучасної автоматизованої системи прогнозування фінансового ринку, яка поєднує переваги гнучких вебтехнологій з потужністю алгоритмів штучного інтелекту. У реаліях цифрової трансформації глобальної економіки ефективне прогнозування є важливим чинником прийняття рішень, що стосуються інвестування, ризик-менеджменту та стратегії поведінки на ринку. Особливої ваги це питання набуває для малого та середнього бізнесу, який не має доступу до дорогих інституційних рішень. У зв'язку з цим зростає потреба у створенні відкритих, доступних і при цьому точних інструментів, здатних виявляти тренди, формувати сигнали і надавати рекомендації. Система, що базується на історичних даних, технічних індикаторах та інтелектуальному аналізі, дозволяє мінімізувати людський фактор у процесі аналізу та підвищити об'єктивність фінансових рішень. Її адаптивність, модульність і можливість використання у вебсередовищі відкривають перспективи

інтеграції в різні програмні екосистеми — від освітніх платформ до фінансових CRM.

Мета дослідження – розробити вебсистему прогнозування фінансового ринку з використанням історичних даних, технічних індикаторів та методів штучного інтелекту на основі мови PHP.

Завдання дослідження:

- Провести аналіз сучасних підходів до прогнозування фінансових ринків;
- Вивчити роль патернів та технічних індикаторів (RSI, EMA) у прогнозуванні;
- Розробити архітектуру вебсистеми прогнозування на PHP;
- Реалізувати функціонал збору, обробки та візуалізації фінансових даних;
- Інтегрувати AI-модель для формування прогнозів на основі історичних даних;
- Провести оцінку точності системи за допомогою відповідних метрик (MAE, MAPE тощо).

Об’єкт дослідження – процес аналізу історичних фінансових даних для прогнозування ринкових змін.

Предмет дослідження – інтелектуальна система прогнозування фінансового ринку, яка використовує технічні індикатори та моделі машинного навчання, реалізована засобами мови PHP.

Методи дослідження – аналіз літературних джерел, методи технічного аналізу (RSI, EMA), часові ряди, машинне навчання, побудова моделей прогнозування, тестування на основі історичних біржових даних програмування, інтеграція API, візуалізація даних.

Практична значущість дослідження полягає у широких можливостях використання розробленої системи як у професійному, так і в освітньому середовищі. Зокрема, система може бути впроваджена у біржові платформи для автоматичного аналізу ринку й формування сигналів до купівлі або продажу активів. Вона стане у нагоді аналітикам для підкріплення власних висновків незалежною оцінкою на основі історичних даних і технічних індикаторів. Також система може застосовуватись у навчальних цілях — як в університетах, що готують фахівців з фінансів та

інформаційних технологій, так і на онлайн-курсах або тренінгах з трейдингу та фінансової грамотності. Її використання у фінансових блогах і CRM-системах трейдерів дозволяє автоматизувати процес генерації прогнозів, створювати аналітичні огляди та підвищувати інформативність для кінцевих користувачів. Її можна адаптувати під мобільні додатки, інформаційні панелі брокерських компаній, системи внутрішньої звітності та консультаційні сервіси для клієнтів..

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ФІНАНСОВОГО РИНКУ З ВИКОРИСТАННЯМ AI

1.1 Аналіз методів прогнозування фінансових ринків

Прогнозування фінансових ринків є однією з ключових задач у сфері економіки, інвестування та фінансової аналітики. Його основна мета полягає у визначенні майбутнього руху цін на активи (акції, облігації, валюти, індекси, сировинні товари, криптовалюти тощо), що дозволяє формувати стратегії торгівлі, мінімізувати ризики й приймати обґрунтовані інвестиційні рішення. Прогнозування має велике значення як для короткострокових спекулятивних операцій, так і для довгострокового портфельного управління. Історично для вирішення цього завдання використовувалися два основні підходи — фундаментальний та технічний аналіз. Фундаментальний аналіз орієнтований на вивчення економічних і політичних факторів, що впливають на ринок. Він охоплює макроекономічні показники (ВВП, інфляцію, безробіття), процентні ставки, новини щодо діяльності компаній, фінансову звітність, а також геополітичні події, які можуть впливати на рівень ризику або очікуваний дохід інвесторів. Цей підхід має сильну аналітичну основу, проте вимагає глибокого розуміння економіки та часто не дозволяє оперативно реагувати на ринкові зміни. Натомість технічний аналіз фокусується на вивченні ринкових даних — цінових графіків, обсягів торгів, індикаторів та моделей, що базуються на історичній інформації. Його основою є припущення, що вся доступна інформація вже врахована в ціні активу, а поведінка ринку має тенденцію до повторення через психологію учасників. Ідея повторення певних шаблонів на графіках — так званих патернів — є фундаментом технічного підходу.

Графічні патерни відіграють важливу роль у формуванні торгових рішень. Серед найбільш поширених — «голова і плечі», «подвійна вершина» та «подвійне дно», трикутники, клини, прапори та вимпели. Ці фігури формуються внаслідок колективної поведінки ринку та відображають періоди нерішучості, накопичення або розвороту тренду. Досвідчені трейдери використовують такі сигнали для

передбачення майбутніх змін ринкової тенденції. Крім патернів, технічний аналіз активно застосовує математичні індикатори, які обчислюються за формулами на основі історичних цін. Найвідомішими є RSI (індекс відносної сили), що відображає перекупленість або перепроданість активу, MACD (ковзна середня конвергенція/дивергенція), яка сигналізує про перетин трендів, та ковзні середні (SMA, EMA), які згладжують коливання для виявлення довгострокових трендів. Також використовуються стохастичні осцилятори, Bollinger Bands, ATR та інші інструменти, що дозволяють аналізувати волатильність і темп змін ціни. Перевагою технічного аналізу є його універсальність і можливість застосування до будь-якого ринку чи інструменту, що має історичні дані. Він широко використовується у біржовій торгівлі, криптовалютах, ринку форекс та на фондовому ринку. Технічний аналіз дозволяє будувати торгові стратегії як вручну, так і в рамках автоматизованих торгових систем (роботів, ботів), що значно розширює його функціонал у поєднанні з інтелектуальними алгоритмами штучного навчання. Незважаючи на те, що технічний аналіз іноді критикується за суб'єктивність і залежність від історичних шаблонів, він продовжує залишатися одним із найбільш популярних підходів до прогнозування, особливо в умовах високої мінливості фінансових ринків. Його поєднання з сучасними методами AI відкриває нові горизонти для створення гібридних систем, які враховують як шаблонні структури, так і складні нелінійні залежності в ринковій поведінці.

Попри відмінність підходів, сучасна практика дедалі частіше орієнтується на створення гібридних моделей прогнозування, які поєднують фундаментальний та технічний аналіз із методами машинного навчання та штучного інтелекту. Це зумовлено тим, що кожен з підходів має свої сильні сторони: фундаментальний аналіз враховує макроекономічний та геополітичний контекст, а технічний аналіз дозволяє виявити короткострокові тренди і патерни, що формуються внаслідок поведінки учасників ринку. Інтеграція методів машинного навчання до цієї системи дає змогу автоматично ідентифікувати взаємозв'язки між цими різними джерелами інформації та побудувати прогнозну модель, здатну до самонавчання і адаптації до змін ринкового середовища. Наприклад, нейронні мережі можуть обробляти як

числові, так і текстові дані, поєднуючи індикатори теханалізу з новинними заголовками, які формують фундаментальний фон. Розширене поєднання методів забезпечує підвищення точності моделей, зменшення похибки прогнозу, гнучкість щодо ринкових умов та можливість створення комплексних систем ухвалення фінансових рішень на основі мультифакторного аналізу.

Упродовж останніх десятиліть зросла популярність кількісних методів прогнозування, які базуються на використанні математичних, статистичних і обчислювальних моделей. Серед таких підходів вирізняються класичні статистичні моделі, як-от ARIMA (авторегресійна модель інтегрованого ковзного середнього), яка ефективно працює з часовими рядами за умови їхньої стаціонарності. Модель GARCH (Generalized Autoregressive Conditional Heteroskedasticity) дозволяє враховувати змінну волатильність на ринку, що особливо актуально для фінансових активів, чия мінливість постійно змінюється. Байєсівські методи відкривають можливості для гнучкого підходу до аналізу невизначеності та оновлення прогнозів у режимі реального часу. Методи регресії, зокрема лінійна та поліноміальна регресія, дозволяють будувати залежності між входом (незалежними змінними, такими як індикатори або макропараметри) та виходом — прогнозованим значенням ціни або доходності активу. Значного поширення набули інструменти машинного навчання (ML), які базуються на здатності алгоритмів виявляти приховані, часто нелінійні залежності у великих наборах фінансових даних. Серед найбільш популярних ML-підходів виділяють дерева рішень (Decision Trees), випадкові ліси (Random Forest), градієнтний бустинг (XGBoost), метод опорних векторів (SVM), а також багатошарові нейронні мережі (MLP). Такі алгоритми здатні враховувати численні взаємозв'язки між ознаками та добре масштабуються на великі обсяги даних. Нейронні мережі дозволяють моделювати складні багаторівневі взаємозалежності між економічними показниками, цінами та часовими характеристиками. Зокрема, глибокі архітектури мереж (deep learning) забезпечують високу точність за рахунок багатоступеневої обробки та автоматичного вилучення ознак. Використання ML-моделей у поєднанні з технічними індикаторами дозволяє створювати інтелектуальні торгові

системи, які здатні адаптуватися до умов ринку та генерувати прогнози в режимі реального часу. Це особливо важливо для високочастотного трейдингу та алгоритмічної торгівлі, де рішення мають прийматись миттєво.

Із розвитком обчислювальних потужностей, поширенням графічних процесорів (GPU) та покращенням алгоритмів оптимізації значну популярність здобули моделі глибокого навчання (Deep Learning), які стали основою багатьох проривів у сфері фінансового прогнозування. До найефективніших архітектур належать рекурентні нейронні мережі (RNN), зокрема модифіковані їх варіації — LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Units), які здатні зберігати інформацію про попередні стани та використовувати її для прогнозування майбутніх значень. Це особливо важливо при роботі з часовими рядами, де часовий контекст має вирішальне значення. LSTM-моделі мають внутрішні механізми керування «запам'ятовуванням» і «забуванням» інформації, що дозволяє їм ефективно справлятися з довгими часовими залежностями, властивими фінансовим котируванням. GRU-моделі є спрощеним, але не менш ефективним аналогом, який також добре справляється з послідовними даними. Особливу увагу дослідники й практики приділяють моделям із механізмом уваги (Attention), який дозволяє моделі фокусуватися на найбільш релевантних частинах вхідної послідовності. Цей підхід дав початок архітектурам на основі трансформерів (Transformers), серед яких найбільш відомими є моделі BERT, GPT та їх варіації. У фінансовому прогнозуванні трансформери використовуються не лише для обробки числових рядів, а й для аналізу текстових даних — новин, твітів, звітів, що дає змогу поєднувати кількісні та якісні дані в єдиній системі. Узагальнюючи, можна сказати, що моделі глибокого навчання є потужним інструментом для побудови високоточних систем прогнозування. Їх здатність виявляти складні, багаторівневі взаємозв'язки між даними, навчатися на великій кількості спостережень та адаптуватися до нових умов робить їх незамінними в умовах високої волатильності та складної поведінки фінансових ринків. У поєднанні з технічними індикаторами, патернами та фундаментальним контекстом

вони формують основу новітніх інтелектуальних платформ для фінансового аналізу.

Сучасні дослідження демонструють, що поєднання традиційного технічного аналізу з алгоритмами штучного інтелекту дозволяє суттєво підвищити точність і стабільність фінансових прогнозів. Гібридні моделі, які використовують патерни, технічні індикатори, часові ряди та дані новин, мають здатність не лише реагувати на історичні закономірності, а й адаптуватися до нових ринкових умов у режимі реального часу. Завдяки цьому зростає ймовірність своєчасного виявлення трендів, змін настроїв інвесторів, періодів високої волатильності та аномальних ринкових подій. Крім того, використання нейронних мереж дозволяє автоматизувати процес вибору значущих ознак, скорочуючи вплив людського фактора та покращуючи інтерпретованість результатів для аналітиків і трейдерів. Такі системи виявляють себе як ефективні інструменти в алгоритмічній торгівлі, управлінні ризиками, побудові торгових стратегій та навіть у прогнозуванні макроекономічних показників. Системи, які базуються на гібридному підході, можуть навчатися на величезних обсягах біржових і фундаментальних даних, що робить їх цінними для банків, інвестфондів, брокерів і корпоративного сектору.

Застосування AI у фінансовому прогнозуванні відкриває нові можливості не лише для аналізу великих обсягів історичних та поточних ринкових даних, а й для їхньої комплексної інтерпретації в реальному часі. За допомогою алгоритмів глибокого навчання та методів обробки природної мови (NLP) штучний інтелект здатен виявляти складні причинно-наслідкові зв'язки, які неочевидні навіть для досвідчених аналітиків. Це включає ідентифікацію прихованих трендів, аналіз настроїв на ринку, реакцію на геополітичні події та фінансові новини. AI-моделі дозволяють здійснювати прогнозування на різних часових горизонтах — від мікросекундного масштабування у високочастотному трейдингу до середньо- і довгострокового аналізу для інституційних інвесторів. Вони інтегруються в торгові платформи, формують сигнали, здійснюють автоматичну корекцію стратегії залежно від поведінки ринку. Застосування таких систем забезпечує конкурентну перевагу за рахунок підвищення швидкості, об'єктивності та точності прийняття

рішень. AI-технології знаходять широке застосування у створенні рекомендаційних систем, оцінці кредитоспроможності, виявленні шахрайства, побудові фінансових цифрових близнюків (digital twins), що моделюють динаміку активів у віртуальному середовищі. Для індивідуальних трейдерів це означає доступ до раніше недоступного рівня аналітики, а для фінансових структур — підвищення ефективності бізнес-процесів і мінімізацію ризиків. У підсумку, штучний інтелект перетворюється з допоміжного інструменту на ключовий елемент нової фінансової екосистеми..

1.2 Патерни в аналізі ринку: визначення та роль у прогнозуванні

Патерни — це повторювані графічні конфігурації, що утворюються на фінансових графіках під час аналізу динаміки цін активів. Їхня поява пов'язана з психологічними аспектами поведінки учасників ринку, які, стикаючись із подібними ринковими ситуаціями, часто реагують аналогічно. Це породжує певну повторюваність у формуванні цінових структур. Патерни є основою для багатьох стратегій технічного аналізу, оскільки дозволяють передбачити ймовірний подальший напрям руху ринку на основі історичного досвіду. Важливо зазначити, що патерни відображають колективну взаємодію попиту й пропозиції, страху й жадібності, інерції й паніки, що притаманні фінансовим ринкам. Коли певна модель повторюється достатньо часто й призводить до схожих наслідків, трейдери починають очікувати подібних результатів, і це підсилює ефект патерну. Таким чином, патерни можна розглядати як відображення соціальних очікувань на ринку, і саме тому їхнє вивчення має як практичну, так і психологічну значущість.

Завдяки розвитку комп'ютерних технологій стало можливим не лише вручну ідентифікувати такі моделі, а й автоматизувати їхнє виявлення. В системах штучного інтелекту ці конфігурації служать потужними предикторами майбутньої динаміки активів. У поєднанні з іншими факторами — як-от обсяг торгів, швидкість зміни ціни або волатильність — патерни забезпечують точні сигнали для ухвалення торгових рішень. Існує велика кількість класифікацій патернів. Найбільш поширеною є поділ на патерни продовження та розвороту. Патерни продовження, як-от "прапор", "вимпел", "трикутник" свідчать про короткочасну

консолідацію ринку перед подальшим рухом у тому ж напрямку. Натомість патерни розвороту — "голова і плечі", "подвійна вершина", "подвійне дно", — свідчать про ймовірну зміну поточної тенденції. Патерни можуть мати як графічне, так і свічкове представлення. Свічкові патерни (наприклад, "поглинання", "доджі", "молот", "перевернутий молот") виникають на базі японських свічок і використовуються для короткострокового прогнозування змін настроїв на ринку. Більшість свічкових моделей розраховані на невеликі часові інтервали та часто сигналізують про зміну сили продавців або покупців. У системах прогнозування на основі штучного інтелекту патерни можуть бути закладені як одна з ключових ознак (features), на яких навчається модель. Ці ознаки можуть бути представлені як бінарні змінні (наявність/відсутність патерну у певному вікні спостереження), числові характеристики (наприклад, висота, тривалість, кут нахилу, амплітуда коливань) або категоріальні мітки (тип патерну, рівень достовірності тощо). Аналіз таких параметрів у контексті історичних даних дозволяє моделі виявляти закономірності, що передують певним ринковим рухам.

Наприклад, визначення появи патерну типу «голова і плечі» або «трикутник» у поєднанні з низькою волатильністю і зростаючим об'ємом торгів може сигналізувати про майбутній прорив рівнів підтримки/опору. У такому випадку AI-модель може надати сигнал до відкриття або закриття позиції, підкріплений статистичною достовірністю на основі попередніх прикладів. У моделюванні торгових стратегій важливе значення має послідовність і повторюваність появи патернів. Визначення того, чи відбувається певна послідовність — наприклад, після фігури консолідації на графіку формується свічковий розворотний патерн — може суттєво підвищити інформативність системи. Також розглядаються такі характеристики як ширина основи, симетричність фігури, глибина корекції, нахил трендових ліній, наявність підтвердження у вигляді обсягів, а також кореляція з іншими індикаторами (RSI, MACD, Bollinger Bands).

В AI-моделях ці патерни можуть комбінуватись із часовими лагами, технічними індикаторами, статистичними ознаками (наприклад, стандартним відхиленням), що дозволяє створювати багатовимірні набори даних для навчання.

При цьому використання рекурентних або згорткових нейронних мереж забезпечує здатність виявляти просторово-часові залежності навіть без явного зазначення логіки розпізнавання. Це дозволяє моделі навчитися розпізнавати структури, які людське око не в змозі чітко класифікувати. Патерни в системах штучного інтелекту не лише відіграють роль як ізольовані сигнали, а й стають частиною складної логіки прогнозування, де кожен з компонентів — як технічний, так і контекстуальний — має вагу в остаточному рішенні моделі.

Сучасні підходи дозволяють автоматизувати процес виявлення патернів за допомогою передових інструментів машинного навчання, комп'ютерного зору та методів глибокого аналізу графіків. У таких системах патерн розглядається не просто як статичне зображення, а як динамічна поведінка ринку, що відображає взаємодію численних змінних у часі. Це дозволяє не лише виявляти класичні фігури, а й знаходити нові, раніше невідомі шаблони, які мають прогностичну силу, але залишались поза увагою традиційного технічного аналізу. Один із найефективніших інструментів у цьому напрямі — це згорткові нейронні мережі (CNN), які використовуються для обробки графічних зображень та побудови моделі на основі вхідних графіків. За допомогою CNN система навчається виявляти візуальні ознаки патернів: кути ліній, симетрію, локальні екстремуми, структуру хвиль тощо. При цьому алгоритм розпізнає не лише статичну конфігурацію, а й враховує масштаб, ротацію і деформацію графічного зображення, що дозволяє уникнути помилок у випадку нестандартного формування. Деякі сучасні моделі поєднують CNN із рекурентними нейромережами (RNN або LSTM), створюючи складні архітектури, здатні вивчати як просторові, так і часові залежності. Такий підхід дозволяє одночасно аналізувати графік як зображення (структура патерну), так і як часовий ряд (послідовність змін цін), що дає змогу досягти високої точності у виявленні навіть найскладніших конфігурацій. У цих системах важливу роль відіграє також попереднє маркування даних — вручну або автоматично сформована база графіків із мітками, яка слугує основою для навчання. Існують також гібридні підходи, які поєднують класичні алгоритми розпізнавання шаблонів (наприклад, за допомогою евристик або правил формування фігур) з глибокими

навчальними архітектурами. Це дозволяє підвищити ефективність системи, комбінуючи інтерпретованість класичних методів із точністю та гнучкістю глибинного навчання. Завдяки цьому вдається будувати системи, які не лише ідентифікують патерни, але й оцінюють їхню достовірність, імовірність спрацювання та рівень історичної надійності.

Додатковим вектором розвитку є використання transfer learning — перенавчання моделей, які були спочатку навчені на загальних задачах обробки зображень (наприклад, ImageNet), для спеціалізованого виявлення фінансових патернів. Це значно зменшує обсяг необхідних даних для досягнення високої точності та пришвидшує процес навчання. У поєднанні з методами data augmentation (генерація варіацій патернів) це дозволяє сформувати стійку до шуму та деформацій модель, яка зберігає ефективність у реальних торгових умовах. Роль патернів у прогнозуванні складно переоцінити, оскільки вони виступають не просто інструментами технічного аналізу, а фундаментальними елементами ринкової поведінки, які формуються під впливом психології натовпу, закономірностей попиту та пропозиції, а також повторюваних реакцій на зовнішні тригери. У торгових системах патерни відіграють критичну роль, бо дозволяють формалізувати візуально впізнавані конфігурації у вигляді чітких, алгоритмічно визначених правил, які можна масштабувати в автоматизованому режимі. Вони дозволяють не лише виявляти потенційні сигнали до входу в угоду або виходу з неї, але й виступають базовими блоками для побудови складних алгоритмічних стратегій, що включають адаптивне управління позиціями, багаторівневі фільтри підтвердження сигналу та динамічне управління ризиками. Патерни можуть бути інтегровані у торгових роботів, які виконують аналіз у режимі реального часу, формують прогнози, коригують параметри стратегії залежно від ринкової волатильності та навіть адаптують тип сигналу до певного інструменту (наприклад, акції, криптовалюта, ф'ючерси). Аналіз патернів сприяє зменшенню суб'єктивності прийняття рішень, адже ґрунтується на емпірично перевірених закономірностях та логіці повторюваності ринкової поведінки. Це дає змогу AI-моделям працювати не лише в межах заздалегідь заданої логіки, а й виявляти нові патерни з даних,

формувані прогностичні правила, класифікувати їх за ступенем достовірності та перевіряти гіпотези на історичних даних. Зростання обчислювальних ресурсів дозволяє формувати сотні тисяч симуляцій на основі різних комбінацій патернів, а також проводити backtesting і forward testing для верифікації якості рішень.

Інтеграція патернів у системи прогнозування на основі штучного інтелекту значно підвищує їхню адаптивність, точність і здатність до генерації корисних рішень у мінливих ринкових умовах. Це не лише підвищує ефективність торгівлі, а й сприяє створенню більш надійних і прозорих фінансових технологій, які здатні працювати в умовах високої конкуренції, нестабільності ринку та інформаційної переваженості.

1.3 Застосування технологій штучного інтелекту в фінансовому аналізі

У сучасному фінансовому середовищі технології штучного інтелекту (ШІ) стають не просто корисним доповненням до аналітичного інструментарію, а повноцінною основою нової епохи прийняття фінансових рішень. З огляду на зростання обсягів інформації, швидкість змін ринкових умов і ускладнення інструментів торгівлі, традиційні методи аналізу дедалі частіше поступаються місцем інтелектуальним системам. ШІ дозволяє обробляти дані в режимі реального часу, виявляти приховані залежності, прогнозувати майбутні сценарії розвитку ринку з високою точністю і водночас адаптуватися до нових умов без потреби ручного втручання. Його застосування охоплює не лише класичні напрями фінансового аналізу, такі як прогнозування цінових трендів або оцінка ризиків, але й значно ширші сфери — виявлення шахрайських транзакцій на основі поведінкової біометрії, автоматичне складання інвестиційних портфелів з персоналізованим рівнем ризику, адаптивні моделі ціноутворення, оцінка кредитоспроможності клієнтів у реальному часі, моніторинг ринкових настроїв та інформаційних потоків із неструктурованих джерел. Таким чином, ШІ формує новий підхід до розуміння та інтерпретації ринкової дійсності, який виводить фінансову аналітику за межі лінійної логіки й робить її гнучкою, динамічною і здатною до самонавчання. Одним із ключових напрямів є використання алгоритмів машинного навчання для побудови моделей прогнозування, які дозволяють не

тільки аналізувати фінансові часові ряди, а й автоматично адаптуватися до мінливих ринкових умов. Алгоритми машинного навчання здатні ефективно розпізнавати нелінійні залежності, кореляції між різними індикаторами та приховані закономірності, які недоступні для класичних методів аналізу. Зокрема, нейронні мережі, метод опорних векторів (SVM), дерева рішень, випадкові ліси (Random Forest), градієнтний бустинг (XGBoost, LightGBM), логістична регресія та кластеризаційні методи (k-means, DBSCAN) активно застосовуються для побудови моделей прогнозування трендів, виявлення аномалій, оцінки ризиків та аналізу поведінки користувачів.

Нейронні мережі, особливо в архітектурах LSTM та GRU, добре справляються з задачами, де важливо враховувати послідовність і тимчасовий контекст даних. Вони дозволяють моделювати динаміку активів, ідентифікувати тренди, передбачати зміну волатильності та розпізнавати потенційні фази ринку (наприклад, накопичення, розподіл, імпульсні рухи). Алгоритми типу Random Forest та XGBoost забезпечують високу точність у задачах класифікації активів за рівнем ризику, кредитного скорингу, побудови інвестиційних рекомендацій. SVM часто використовується для виявлення крайових ситуацій та формування сигналів при аномальній активності на ринку.

Машинне навчання також ефективно застосовується для аналізу альтернативних джерел даних — таких як супутникові знімки, поведінкові дані користувачів, мобільні транзакції, дані про клієнтську активність. Алгоритми навчаються не тільки на структурованих числових даних, але й на обробці текстових, графічних, часових і навіть аудіовізуальних джерел. Розширення джерел для навчання дозволяє моделі точніше оцінювати поточні умови та будувати складні прогностичні сценарії. Завдяки широкому спектру алгоритмів та можливості комбінувати моделі у ансамблі (ensemble methods), такі як багатомодельне голосування або стекінг, системи машинного навчання досягають високої стійкості до шуму, забезпечують гнучкість у задачах адаптивного аналізу та значно перевершують класичні статистичні підходи за точністю прогнозування в умовах волатильного ринку. III також активно використовується в галузі

обробки природної мови (NLP), що дозволяє реалізовувати якісно новий рівень аналітики на основі текстових джерел. У фінансовій сфері текстова інформація відіграє надзвичайно важливу роль, оскільки настрої інвесторів, ринкові очікування та репутаційні ризики часто формуються через публікації у ЗМІ, соціальних мережах, аналітичних звітах та публічних заявах компаній. Моделі NLP аналізують новини, твіти, стенограми пресконференцій, звіти про прибутки, коментарі в інвесторських блогах і форумах. Вони здатні класифікувати текст за тональністю (позитивна, нейтральна, негативна), виявляти ключові події, що можуть вплинути на котирування активів, та оцінювати силу емоційного забарвлення повідомлень. Зокрема, моделі на основі трансформерів (BERT, RoBERTa, GPT) дозволяють глибоко аналізувати контекст, враховуючи залежності між словами, фразами та абзацами, що є особливо важливим при обробці економічних та технічних текстів. Застосування таких моделей дозволяє створювати системи аналізу ринкових настроїв (sentiment analysis), які інтегруються у торгові стратегії, автоматично змінюючи ваги активів у портфелі відповідно до зміни інформаційного фону. Окрім того, NLP використовується для генерації коротких оглядів ринку, автоматичного складання звітів, класифікації подій (наприклад, злиття, банкрутства, зміна керівництва) та виявлення «мовних патернів», які передували значним змінам на ринку в минулому.

Сучасні фінансові компанії активно інтегрують NLP у свої платформи як у форматі бекенд-аналітики, так і в інтерфейси для клієнтів: чат-боти на базі LLM, автоматичні інтерпретатори звітів, перекладачі фінансових термінів у реальному часі. Усе це сприяє глибшій взаємодії з клієнтами, підвищенню точності рішень і зниженню людського навантаження на аналітичні відділи. Іншою важливою сферою є високочастотна торгівля (High-Frequency Trading, HFT), яка базується на використанні надшвидких алгоритмів, що здійснюють десятки або навіть сотні тисяч угод протягом одного торгового дня. Ці алгоритми приймають рішення за мілі- або навіть мікросекунди, реагуючи на мінімальні коливання цін, зміни обсягу заявок або структури глибини ринку. HFT вимагає не лише наднизьких затримок у передачі інформації, а й надвисокої обчислювальної потужності. У таких умовах

використовуються спеціалізовані моделі глибокого навчання, які оптимізовані для роботи в умовах обмеженого часу, зокрема згорткові нейронні мережі (CNN) для аналізу патернів у мікроструктурі ринку, LSTM-моделі для послідовного прогнозування котирувань, а також reinforcement learning (навчання з підкріпленням) — для формування стратегії прийняття торгових рішень в реальному часі. Важливим компонентом ефективності HFT-систем є також архітектура інфраструктури. Використовуються апаратні прискорювачі (FPGA, GPU), сервери з розміщенням у дата-центрах бірж (co-location), розподілені обчислення та високопродуктивні системи передачі даних (InfiniBand, оптика). Крім того, критичним фактором є здатність AI-моделі не лише швидко реагувати, а й постійно оновлювати свої параметри відповідно до мінливих ринкових умов — так зване адаптивне або контекстне навчання (adaptive learning). Системи HFT часто інтегруються з компонентами моніторингу ризиків у реальному часі, щоб запобігти катастрофічним збиткам, які можуть виникнути внаслідок непередбачених подій, наприклад, "флеш-крешів". Для цього використовуються AI-алгоритми нагляду за поведінкою стратегії, аналізу впливу угод на ринок (market impact) та прогнозування короткострокових сценаріїв з урахуванням багатьох параметрів, включно з латентною ліквідністю, чергами заявок і статистичними аномаліями.

У сукупності, впровадження штучного інтелекту у сферу високочастотної торгівлі створює новий рівень ефективності й керованості торговими операціями, де перевагу отримує не лише швидкість, але й здатність до інтелектуального аналізу, автоматичного самокоригування і превентивного управління ризиками. III також забезпечує потужний інструментарій для управління ризиками, як у стратегічному, так і в тактичному вимірах. Завдяки здатності моделей ідентифікувати складні кореляції, передбачати негативні сценарії розвитку подій і оцінювати вплив внутрішніх і зовнішніх факторів, аналітичні системи на базі III можуть оперативно сигналізувати про потенційні загрози в інвестиційному портфелі. Це дозволяє трейдерам і портфельним менеджерам приймати обґрунтовані рішення щодо хеджування, ребалансування або ліквідації позицій. У

сфері формування інвестиційних портфелів великого поширення набули так звані robo-advisors — автоматизовані системи консультування, які на основі аналізу поведінки клієнта, фінансових цілей, толерантності до ризику, макроекономічних показників та поточної ринкової ситуації формують персоналізовані портфелі. Вони можуть працювати в режимі постійної адаптації до змін ринку, автоматично змінюючи структуру активів відповідно до обраної стратегії або запрограмованої логіки. Такі рішення уже активно застосовуються як у B2C-сегменті (банківські мобільні застосунки, онлайн-брокери), так і в B2B (інструменти для керування клієнтськими портфелями у приватному банкінгу). Окрему увагу заслуговують ІІІ-рішення в галузі персоналізації фінансових послуг. Вони дозволяють не лише формувати рекомендації щодо інвестицій чи кредитних продуктів, а й адаптувати інтерфейси, налаштування комунікації, тип і частоту нагадувань — усе на основі аналізу індивідуальних звичок, уподобань та історії фінансової поведінки. Це підвищує лояльність клієнтів, зменшує навантаження на контакт-центри та сприяє зростанню конверсії у продажах.

Ще однією важливою сферою є виявлення шахрайства (fraud detection). Сучасні моделі навчаються на мільйонах транзакцій, вивчають нормальну поведінку клієнтів та здатні з високою точністю виявляти відхилення, які можуть свідчити про спробу шахрайства — як технічного (викрадення платіжних даних), так і поведінкового (шахрайські схеми інсайдерів). Особливо ефективним є поєднання класичних правил з глибокими моделями (deep fraud detection), що дозволяє як мінімізувати хибнопозитивні спрацьовування, так і оперативно реагувати на нові, ще невідомі сценарії атак. Аналітика на основі штучного інтелекту відкриває банкам, інвестиційним фондам, страховим компаніям і фінтех-стартапам нові горизонти ефективності: зменшення витрат на аналітичні підрозділи, зниження рівня ризиків, підвищення точності оцінок, прискорення обслуговування клієнтів і масштабування сервісів без втрати якості. Переваги використання ІІІ в фінансовому аналізі очевидні та багатогранні: підвищення точності прогнозів завдяки глибокому аналізу закономірностей, здатність працювати з неструктурованими джерелами даних (текст, зображення, поведінка

користувачів), автоматизація багатьох рутинних операцій, що скорочує операційні витрати, а також масштабованість рішень, які можуть бути адаптовані під потреби як великих банків, так і невеликих фінтех-компаній. Крім того, використання ШІ дозволяє прискорити процес ухвалення рішень, забезпечити персоналізований підхід до кожного клієнта, а також впровадити механізми самонавчання, які з кожною ітерацією підвищують якість прогнозів та управлінських стратегій.

Разом із тим, активне впровадження ШІ супроводжується рядом нових викликів. Одним із ключових є інтерпретованість моделей: складні нейронні мережі часто функціонують як «чорні скриньки», що ускладнює перевірку причин прийнятих рішень. Це створює ризики як для відповідальності організацій перед регуляторними органами, так і для довіри клієнтів. Виникають також етичні дилеми — наприклад, у питаннях дискримінації за ознаками, що не є релевантними для фінансової оцінки, або в ситуаціях, коли автоматизовані системи приймають рішення про відмову в кредитуванні. Зростає необхідність посилення кібербезпеки, оскільки моделі ШІ можуть бути об'єктом атак (наприклад, отруєння навчального набору або підміна вхідних даних), що здатне призвести до спотворення результатів аналізу. Все це вимагає нових підходів до валідації, пояснюваності (explainability), аудиту й контролю за ШІ-системами, а також впровадження стандартів відповідального використання інтелектуальних технологій у фінансовому секторі.

Загалом, впровадження технологій штучного інтелекту у фінансову сферу не лише трансформує традиційні підходи до аналізу, а й радикально змінює саму природу взаємодії з фінансовими даними, клієнтами та регуляторним середовищем. ШІ стає каталізатором появи нових бізнес-моделей, таких як алгоритмічне банкінгування, повністю автоматизовані брокерські платформи, децентралізовані фінансові сервіси (DeFi), а також «розумні» страхові продукти з динамічним преміюванням. У цифровій економіці майбутнього саме ті організації, які зможуть ефективно впровадити штучний інтелект у свою операційну структуру, здобудуть значну конкурентну перевагу завдяки здатності до швидкої адаптації, проактивного аналізу ризиків і високоточного прогнозування. Таким чином, ШІ не

просто підсилює наявні фінансові інструменти, а стає фундаментом нової ери — ери інтелектуально керованих фінансів.

Висновок до розділу 1

У першому розділі було розглянуто теоретичні засади фінансового прогнозування в умовах цифровізації економіки.

У підрозділі 1.1 здійснено огляд основних методів прогнозування фінансових ринків: від традиційного технічного та фундаментального аналізу до сучасних підходів, заснованих на статистичних моделях і машинному навчанні. Особливу увагу приділено їх порівняльній характеристиці та актуальності для поточних умов фінансової нестабільності.

У підрозділі 1.2 розкрито поняття ринкових патернів, класифікацію графічних і свічкових моделей та їхню роль у процесі прийняття торгових рішень. Показано, як ці патерни можуть бути використані в рамках автоматизованих систем прогнозування, включаючи моделі, що навчаються з прикладів історичних даних.

У підрозділі 1.3 досліджено широкі можливості застосування технологій штучного інтелекту у фінансовій сфері. Розглянуто алгоритми глибокого навчання, аналіз природної мови, високочастотну торгівлю, автоматизованих радників, моделі для виявлення шахрайства та інструменти ризик-менеджменту. Визначено переваги та виклики впровадження ШІ в аналітичні процеси.

Узагальнено, що поєднання класичних методів аналізу з AI-рішеннями дозволяє досягти нових рівнів точності, швидкості та адаптивності в системах прогнозування. Здобуті теоретичні знання створюють надійну основу для подальшої розробки практичного інструментарію, що буде представлено в наступних розділах.

2 АНАЛІЗ ДАНИХ І ПОСТАНОВКА ЗАДАЧІ ПРОГНОЗУВАННЯ З ВИКОРИСТАННЯМ ВЕБ-ІНТЕРФЕЙСУ

2.1 Вибір та обробка історичних фінансових даних

Початковим етапом створення системи прогнозування стало формування якісного набору даних, який відповідав би як вимогам машинного навчання, так і специфіці фінансової тематики. Для забезпечення достовірності прогнозів було вирішено використовувати історичні біржові дані, які включають не лише стандартні поля OHLC (відкриття, максимум, мінімум, закриття), обсяги торгів і часові мітки, а й додаткові мета-поля, такі як статус торгів, тип сесії, рівень ліквідності та індикатори ринку. Такі параметри є надзвичайно цінними для поглибленого аналізу ринкової поведінки, побудови більш точних моделей прогнозування та розробки аналітичних інструментів.

Особливу увагу було приділено якості джерела даних. Після тестування кількох провайдерів фінансової інформації було обрано сервіс Polygon.io — один із найбільш стабільних та гнучких інструментів для роботи з історичними даними. Його API надає можливість запитувати дані в різних форматах і таймфреймах, зокрема через ендпоінт `/v2/aggs/ticker/{ticker}/range/{multiplier}/{timespan}/{from}/{to}`. Це дозволяє задавати як короткі (1 хвилина), так і тривалі періоди (дні, тижні), а також точно контролювати діапазон дат та інтервал агрегації.

На основі відповідей API формується масив об'єктів JSON, кожен з яких містить ключі *t* - (timestamp), *o* - (ціна відкриття), *h* - (максимум), *l* - (мінімум), *c* - (закриття), *v* - (обсяг торгів). Це дозволяє будувати точні графіки та індикатори, а також дає змогу швидко інтегрувати дані в PHP-середовище. Використовуючи `json_decode()`, JSON-масив перетворюється в асоціативний масив для подальшої обробки. Цей етап заклав фундамент для усього наступного логічного ланцюга — від технічної обробки до інтеграції з AI-модулями та побудови прогнозної системи.

Дані для навчання та тестування інтелектуальної системи були отримані з перевірених відкритих джерел, зокрема з фінансового API Polygon.io, який забезпечує високу швидкість доступу до біржової інформації та підтримує гнучкі

параметри фільтрації. З використанням цього API вдалося реалізувати ефективний механізм завантаження історичних біржових котирувань (OHLCV — Open, High, Low, Close, Volume) у зручному агрегованому вигляді.

Конкретно використовувався endpoint:

/v2/aggs/ticker/{ticker}/range/{multiplier}/{timespan}/{from}/{to}

де:

- *{ticker}* — біржовий тикер активу (наприклад, AAPL, BTC-USD, XAUUSD);
- *{multiplier}* — кратність обраного періоду (наприклад, 1);
- *{timespan}* — одиниця часу (day, minute, hour);
- *{from}* і *{to}* — початкова та кінцева дати в форматі YYYY-MM-DD.

Перед відправленням запиту формувався повний URL з підстановкою параметрів та додаванням API-ключа для авторизації. Для прикладу, отримання денних котирувань для акції AAPL (Apple Inc.) за 2024 рік виглядає так:

https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2024-01-01/2024-12-31?adjusted=true&sort=asc&limit=50000&apiKey=YOUR_API_KEY

Тут:

- *adjusted=true* — вказує на використання скоригованих даних (з урахуванням сплітів і дивідендів);
- *sort=asc* — сортування за зростанням дати;
- *limit=50000* — максимальна кількість записів;
- *apiKey=YOUR_API_KEY* — ваш персональний ключ доступу до API.

Запит надсилався через HTTP-метод GET за допомогою стандартних бібліотек (requests у Python або cURL у PHP/CLI). Отримані у відповідь дані у форматі JSON зберігалися та далі оброблялися — з них обиралися лише потрібні поля: t (timestamp), o (відкриття), h (максимум), l (мінімум), c (закриття), v (обсяг). Завдяки Polygon.io вдалося побудувати надійну систему збору фінансових даних для аналізу і прогнозування, що відповідає сучасним вимогам до масштабованості, точності та швидкодії.

У відповіді поверталася структура JSON, що містила масив об'єктів під ключем results, де кожен об'єкт представляв собою один часовий блок із даними: t

— timestamp, o — ціна відкриття, h — максимум, l — мінімум, c — ціна закриття, v — обсяг торгів. Така структура була ідеальною для подальшої інтеграції в PHP, оскільки дозволяла легко перетворити її на асоціативний масив та зберегти у потрібному форматі.

Наведено приклад однієї відповіді у форматі JSON:

```
{
  "results": [
    {
      "t": 1713801600000,
      "o": 134.91,
      "h": 136.23,
      "l": 134.12,
      "c": 135.76,
      "v": 89324587
    },
    {
      "t": 1713888000000,
      "o": 135.80,
      "h": 136.70,
      "l": 133.92,
      "c": 134.35,
      "v": 81234788
    }
  ]
}
```

У PHP такий масив оброблявся за допомогою функції `json_decode()` та парсився через цикл `foreach`, де кожен запис надавався для очищення, трансформації та збереження. Саме зручність API, підтримка точкових таймфреймів і деталізованих OHLC-структур стали вирішальними факторами у виборі цього джерела. Для автоматичного збору, обробки та збереження даних було створено набір PHP-скриптів, що виконують ключові операції з взаємодії з API. На першому етапі скрипти формують динамічні GET-запити до сервісу Polygon.io з параметрами дати, інтервалу, символу активу, біржі тощо. Відповіді з

API надходять у форматі JSON і автоматично зберігаються у тимчасові буфери, після чого виконується трансформація JSON у багатовимірні асоціативні масиви для зручної обробки. Всі ключові поля (ціна відкриття, закриття, максимум, мінімум, обсяг, timestamp) вибираються із глибоко вкладених структур через рекурсивний перебір і перетворення у вигляд, придатний для запису в базу даних. Далі кожен запис перевіряється на повноту та логічну узгодженість: усуваються відсутні або некоректні значення, а також значення, які виходять за межі допустимих діапазонів. Оброблені дані зберігаються у вигляді асоціативних масивів або серіалізованих JSON-файлів, які тимчасово розміщуються на сервері в окремій директорії. Для великих масивів даних реалізовано систему пагінації та послідовної агрегації: кожен запит виконується з динамічно змінюваним часовим інтервалом, маркером сторінки або діапазоном дат. У разі недоступності API або отримання порожніх чи пошкоджених відповідей скрипт виконує автоматичну повторну спробу та реєструє повідомлення у лог-файл. Таким чином реалізовано стійкий цикл збору фінансової інформації з можливістю масштабування, обробки в потоковому режимі й подальшого аналізу без використання повноцінної реляційної бази даних.

Після отримання даних реалізовано комплексну систему очищення та фільтрації записів. Перш за все, проводиться усунення дублікатів, які можуть з'явитися внаслідок повторного запиту або технічного збою під час обміну з API. Пропущені значення визначаються на основі неповних або порожніх полів: якщо йдеться про критичні поля (наприклад, ціна закриття або timestamp), то запис виключається; у менш критичних випадках здійснюється заповнення найближчими відомими значеннями (методом форвардного або зворотного заповнення).

```
foreach ($data['results'] as $row) {
    if (!isset($row['c']) || $row['c'] == 0) continue; //
    виключити записи без ціни закриття
    $cleaned[] = [
        'timestamp' => (int)($row['t'] / 1000),
        'open' => round($row['o'], 2),
        'high' => round($row['h'], 2),
```

```

        'low' => round($row['l'], 2),
        'close' => round($row['c'], 2),
        'volume' => (int)$row['v']
    ];
}

```

Також нормалізуються часові мітки: конвертація з ISO-формату або мілісекунд у стандартний UNIX timestamp дозволяє уніфікувати часовий ряд і забезпечити правильну агрегацію. Залежно від цілей прогнозування, дані групуються за хвилинним або денним інтервалом, із відповідним усередненням або агрегацією OHLC-полів. При цьому відкидаються надмірно короткі або порожні сесії, які могли утворитися через збої біржового потоку або вихідні дні.

Для забезпечення точності моделей реалізовано фільтрацію екстремальних значень (наприклад, нульові ціни або різкі сплески на рівні десятків відсотків). Такі значення позначаються як потенційні аномалії і виносяться у звіт про підготовку датасету. Перед збереженням або передачею у наступний етап — обчислення індикаторів — усі поля приводяться до коректних типів: числові — до float або integer, часові — до timestamp. Також виконується сортування записів у суворо хронологічному порядку, що критично важливо для коректної роботи алгоритмів машинного навчання, які залежать від послідовності подій у часовому ряді.

```

usort($cleaned, function($a, $b) {
    return $a['timestamp'] <=> $b['timestamp'];
});

foreach ($cleaned as $entry) {
    if ($entry['close'] > $entry['open'] * 2 || $entry['close']
< $entry['open'] * 0.5) {
        // занадто різкі коливання — пропускаємо або логуємо
        continue;
    }
    $finalData[] = $entry;
}

```

У результаті реалізованих етапів збору, парсингу, очищення, нормалізації та структурування даних сформовано високоякісний датасет, придатний для

подальшого математичного та аналітичного опрацювання. Завдяки збереженню цілісності часових рядів, перевірці на аномалії, приведенню всіх значень до уніфікованих форматів і багаторівневій валідації отримано надійну основу, що забезпечує коректність та стабільність усіх наступних процедур. Такий датасет є критично важливим для точного розрахунку технічних індикаторів (RSI, MACD, EMA), побудови графічних моделей, навчання прогностичних моделей штучного інтелекту та їхньої валідації за допомогою об'єктивних метрик якості прогнозу..

2.2 Побудова індикаторів (RSI, EMA) та виявлення патернів

Після формування чистого та структурованого датасету ключовим етапом стала побудова технічних індикаторів, які відіграють роль числових ознак (features) для навчання прогностичної моделі. Індикатори дозволяють перетворити сирі дані котирувань на більш інформативні характеристики, які враховують динаміку цін, силу трендів та коливання ринку. У нашій реалізації були обрані два основні індикатори — RSI (Relative Strength Index) та EMA (Exponential Moving Average), які є одними з найпоширеніших серед трейдерів і мають високу кореляцію з ринковою активністю. Індикатори не лише дають змогу здійснювати поточну оцінку ринкової ситуації, а й надають цінну інформацію для побудови прогнозів на основі часових залежностей. Використання RSI дозволяє виявляти зони перекупленості або перепроданості активу, тоді як EMA слугує як фільтр тренду та підтримка сигналів, заснованих на цінах закриття. Такі індикатори добре піддаються нормалізації, масштабуванню та інтеграції в алгоритми машинного навчання, що робить їх особливо корисними для задач фінансового прогнозування.

Розрахунок EMA (експоненційної ковзної середньої) здійснювався на основі серії цін закриття активу, які були отримані з попередньо очищеного датасету. EMA є важливим інструментом технічного аналізу, який дозволяє зменшити шум короткострокових коливань і виявити домінуючий напрямок ринку. На відміну від простої ковзної середньої (SMA), яка надає однакову вагу всім спостереженням, EMA зосереджує більше уваги на останніх значеннях, що робить її чутливішою до змін і дозволяє швидше реагувати на формування нових трендів або потенційних розворотів. Цей індикатор широко використовується для побудови торгових

стратегій типу "перехрестя ЕМА" (наприклад, коли коротка ЕМА перетинає довгу), а також для формування динамічних рівнів підтримки та опору. У системі, що реалізовувалася, ЕМА обчислювалася для кількох таймфреймів (наприклад, 10, 20 і 50 періодів), а результати зберігалися разом із часовими мітками для подальшої візуалізації графіків та аналізу сигналів. Алгоритм був реалізований у PHP за допомогою формули згладженого експоненційного розрахунку, що дає змогу обробляти великі масиви цін без значної втрати продуктивності.

```
function calculateEMA(array $prices, int $period): array {
    $ema = [];
    $k = 2 / ($period + 1);
    $ema[0] = $prices[0];
    for ($i = 1; $i < count($prices); $i++) {
        $ema[$i] = $prices[$i] * $k + $ema[$i - 1] * (1 - $k);
    }
    return $ema;
}
```

RSI (індекс відносної сили) — один із ключових технічних індикаторів, який використовується для оцінки сили ринкового тренду та ймовірності його розвороту. Він дозволяє виявити зони перекупленості (значення $RSI > 70$) та перепроданості ($RSI < 30$), що сигналізують про потенційне послаблення поточного тренду або його завершення. Розраховується як відношення середнього приросту до середнього зниження за обраний період, зазвичай 14. RSI є осцилятором і приймає значення в межах від 0 до 100, що робить його зручним для інтерпретації у трейдингу та подальшого використання як нормалізованої ознаки у моделях машинного навчання.

```
if ($ema10[$i] > $ema20[$i] && $rsi[$i] < 70) {
    $signal = 'buy';
} elseif ($ema10[$i] < $ema20[$i] && $rsi[$i] > 30) {
    $signal = 'sell';
} else {
    $signal = 'hold';
}
```

```
$signals[$i] = $signal;
```

В нашій реалізації RSI використовується не лише для візуального аналізу стану ринку, а й для формування навчального набору даних у моделі прогнозування, де його значення є вагомим фактором для визначення фази активу. Нижче — спрощена реалізація RSI у PHP:

```
function calculateRSI(array $prices, int $period = 14): array {
    $rsi = [];
    $gains = $losses = [];
    for ($i = 1; $i < count($prices); $i++) {
        $delta = $prices[$i] - $prices[$i - 1];
        $gains[] = max(0, $delta);
        $losses[] = max(0, -$delta);
    }
    for ($i = $period; $i < count($prices); $i++) {
        $avg_gain = array_sum(array_slice($gains, $i - $period,
$period)) / $period;
        $avg_loss = array_sum(array_slice($losses, $i - $period,
$period)) / $period;
        $rs = $avg_loss == 0 ? 100 : $avg_gain / $avg_loss;
        $rsi[$i] = 100 - (100 / (1 + $rs));
    }
    return $rsi;
}
```

Значення індикаторів використовувалися не лише для загального аналізу ринкових трендів і визначення зон активності, але й для автоматизованого генерування сигналів до входу або виходу з позицій. На основі історичних значень RSI та ЕМА створювалися евристичні правила, які формували сигнали, що пізніше порівнювались із реальними рухами ціни для перевірки ефективності. Наприклад, поєднання умов "ЕМА(10) > ЕМА(20) та RSI < 70" трактувалося як ймовірне продовження висхідного тренду.

Додатково було реалізовано окремий блок для виявлення графічних патернів, що включав базовий аналіз локальних максимумів та мінімумів з використанням ковзного вікна. Використовувалась логіка виявлення ключових точок повороту в

ціновому ряді та побудови на їх основі геометричних моделей. Наприклад, для патерну "подвійне дно" шукалися дві локальні мінімальні точки з відносно однаковим значенням та однією проміжною вершиною між ними, яка підтверджувала завершення фази зниження. Така перевірка реалізовувалась через цикл зі зсувом на k елементів і логічними умовами, що порівнювали мінімальні значення, їхнє відношення та подальший прорив ціни вгору.

```
for ($i = 2; $i < count($prices) - 2; $i++) {
    if ($prices[$i - 2] > $prices[$i] && $prices[$i + 2] >
$prices[$i]) {
        if (abs($prices[$i] - $prices[$i + 2]) < $prices[$i] *
0.02) {
            $patternMatch[$i] = 'double_bottom';
        }
    }
}
```

Аналогічно тестувалися прості умови для виявлення "подвійної вершини", "трикутника" та "флагштока" — за фіксацією послідовностей максимумів і мінімумів. Патерни не лише виводились у лог-файл, але й використовувались як структурні ознаки в моделі, де кожен випадок кодувався як бінарна змінна (true/false) в окремому масиві ознак. Цей етап суттєво підвищив інформативність і точність наступної фази — побудови прогнозової моделі, оскільки індикатори та патерни забезпечили модель машинного навчання ціннісними ознаками, що враховують як локальні закономірності, так і глобальні ринкові тенденції. Використання EMA і RSI дозволило моделі «бачити» тренд, а патерни — виявляти структурні точки зміни фази ринку. Зі свого боку, багаторівнева система фільтрації та генерації сигналів зменшила шум і підвищила якість навчального датасету, що прямо вплинуло на підвищення точності прогнозу в режимі backtest і forward test..

2.3 Розробка серверної частини системи прогнозування на РНР

Серверна частина системи прогнозування була реалізована повністю на мові РНР, без використання сторонніх фреймворків, що забезпечило високу контрольованість логіки, мінімальну залежність від сторонніх компонентів та

сумісність з існуючим серверним середовищем. Такий підхід дозволив гнучко формувати архітектуру системи, оптимізуючи кожен модуль під специфіку фінансових даних.

Основними задачами бекенду стали: отримання фінансових даних із зовнішнього API (Polygon.io), первинна обробка та фільтрація цих даних, обчислення технічних індикаторів (EMA, RSI), виявлення базових графічних патернів на історичних даних, генерація торгових сигналів і формування структурованої JSON-відповіді для фронтенд-інтерфейсу або зовнішнього REST API. Система побудована за модульною архітектурою, де кожен компонент (скрипт) відповідає за окрему функціональну одиницю — це спрощує масштабування, налагодження та оновлення окремих частин.

У структурі ресурсу чітко виділено:

- /core/fetch_data.php — завантаження та валідація сирих даних з API;
- /core/indicators.php — обчислення технічних індикаторів;
- /core/patterns.php — модуль для виявлення локальних розворотних формацій;
- /core/signals.php — логіка генерації buy/sell/hold рішень;
- /core/response.php — узагальнення результатів і передача на фронт;
- /config/config.php — конфігураційні змінні: API-ключ, тикер, таймфрейм, часові рамки;
- /logs/ — папка журналів з повідомленнями про помилки, запити та діагностику;
- /public/index.php — єдиний вхідний контролер для взаємодії з інтерфейсом або AJAX-запитами.

Така структура забезпечує максимальну гнучкість, прозорість і розширюваність проекту, дозволяючи інтегрувати нові індикатори, додавати нові типи патернів або перенаправляти вихідні дані в інші системи (CRM, трейдинг-бот тощо).

Структура скриптів

Система складалася з кількох логічних блоків:

- fetch_data.php – отримання та парсинг даних з API;
- indicators.php – розрахунок RSI, EMA та формування сигналів;
- patterns.php – виявлення графічних патернів;
- response.php – формування відповіді у форматі JSON або HTML;
- config.php – збереження ключів доступу, таймфреймів, параметрів індикаторів.

Отримання та декодування JSON

Цей блок відповідає за взаємодію з API Polygon.io, що є ключовим джерелом біржових фінансових даних. Він формує динамічний GET-запит, до якого додаються параметри, такі як тикер (символ активу), інтервал агрегації (наприклад, 1 день або 1 хвилина), діапазон дат і авторизаційний ключ API. Надсилаючи цей запит, система отримує відповідь у форматі JSON, яка містить масив історичних даних OHLCV (open, high, low, close, volume). Після отримання відповіді JSON перетворюється на PHP-масив за допомогою функції `json_decode()`, після чого дані розбираються для подальшої обробки в рамках індикаторного або патерн-аналізу. Така структура дозволяє реалізувати повністю динамічне отримання даних без необхідності зберігання на сервері.

```
$url =
"https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2023-01-
01/2023-06-01?apiKey=YOUR_API_KEY";

$response = file_get_contents($url);
$data = json_decode($response, true);
$quotes = $data['results'] ?? [];
```

Обробка масиву та формування локального масиву для розрахунків

Після декодування JSON виконується первинна фільтрація записів для забезпечення достовірності подальших розрахунків. На цьому етапі перевіряється наявність ключового значення ціни закриття (close) у кожному записі. Якщо значення відсутнє або дорівнює нулю, такий запис ігнорується як некоректний. Після цього всі допустимі значення ціни закриття збираються в окремий масив, який потім використовується для розрахунку технічних індикаторів, таких як EMA та RSI. Додатково, на цьому етапі можна також зберігати інші значення (наприклад,

обсяг торгів чи максимум/мінімум) в окремі масиви, якщо вони потрібні для додаткових індикаторів або перевірок. Цей підхід дозволяє ефективно підготувати вхідні дані до наступних етапів обробки в системі.

```
$closes = [];
foreach ($quotes as $q) {
    if (!isset($q['c']) || $q['c'] == 0) continue;
    $closes[] = $q['c'];
}
```

Розрахунок ЕМА

Цей модуль відповідає за обчислення експоненційної ковзної середньої (ЕМА) для заданого періоду, що є критично важливим етапом у технічному аналізі фінансових ринків. ЕМА — це різновид згладженої середньої, яка надає більшого значення останнім цінам, тим самим забезпечуючи чутливішу реакцію на нові зміни на ринку. Це дозволяє трейдерам та алгоритмам швидко виявляти початок нових трендів або зміни напрямку руху активу. В нашій реалізації, побудованій на РНР, ЕМА розраховується через цикл із накопиченням зважених значень та коефіцієнтом згладжування, який залежить від заданого періоду. Отримані значення ЕМА зберігаються у вигляді масиву й використовуються як основа для побудови сигналів, детекції перетинів, а також візуалізації в графічному модулі.

```
function calculateEMA(array $prices, int $period): array {
    $ema = [];
    $k = 2 / ($period + 1);
    $ema[0] = $prices[0];
    for ($i = 1; $i < count($prices); $i++) {
        $ema[$i] = $prices[$i] * $k + $ema[$i - 1] * (1 - $k);
    }
    return $ema;
}

$ema10 = calculateEMA($closes, 10);
$ema20 = calculateEMA($closes, 20);
```

Розрахунок RSI

Індекс відносної сили — це популярний технічний осцилятор, який вимірює швидкість і зміну цінових рухів на фінансовому ринку. Його головна мета — визначити ступінь перекупленості або перепроданості активу за допомогою порівняння середнього зростання і середнього падіння цін за певний період. Значення RSI коливається в діапазоні від 0 до 100, де показники вище 70 сигналізують про перекупленість, а нижче 30 — про перепроданість. Це дозволяє виявляти потенційні розвороти або підтвердження наявного тренду. RSI ефективно використовується в парі з ЕМА як логічний фільтр для генерації торгових сигналів і визначення точок входу/виходу з позиції в системі прогнозування.

```
function calculateRSI(array $prices, int $period = 14): array {
    $rsi = [];
    $gains = $losses = [];
    for ($i = 1; $i < count($prices); $i++) {
        $delta = $prices[$i] - $prices[$i - 1];
        $gains[] = max(0, $delta);
        $losses[] = max(0, -$delta);
    }
    for ($i = $period; $i < count($prices); $i++) {
        $avg_gain = array_sum(array_slice($gains, $i - $period,
$period)) / $period;
        $avg_loss = array_sum(array_slice($losses, $i - $period,
$period)) / $period;
        $rs = $avg_loss == 0 ? 100 : $avg_gain / $avg_loss;
        $rsi[$i] = 100 - (100 / (1 + $rs));
    }
    return $rsi;
}

$rsi = calculateRSI($closes);
```

Генерація торгових сигналів

На цьому етапі система аналізує результати розрахунків технічних індикаторів для формування інвестиційних рішень: купівлі (buy), продажу (sell)

або утримання (hold) активу. Основна логіка ґрунтується на перетині двох експоненційних ковзних середніх (EMA) з різними періодами, а також на значеннях RSI, який слугує фільтром для зниження хибнопозитивних сигналів. Якщо короткострокова EMA (наприклад, EMA10) перетинає довгострокову EMA (EMA20) знизу вгору, і RSI при цьому не перевищує зону перекупленості, система генерує сигнал на купівлю. Навпаки, перетин зверху вниз у поєднанні з високим RSI може бути сигналом на продаж. Кожен сигнал формується для конкретного моменту часу, зберігається в масив і надалі використовується для візуалізації та оцінки ефективності в історичному розрізі.

```
$signals = [];
for ($i = 1; $i < count($closes); $i++) {
    if ($ema10[$i] > $ema20[$i] && $rsi[$i] < 70) {
        $signals[$i] = 'buy';
    } elseif ($ema10[$i] < $ema20[$i] && $rsi[$i] > 30) {
        $signals[$i] = 'sell';
    } else {
        $signals[$i] = 'hold';
    }
}
```

Виявлення патернів (наприклад, "подвійне дно")

Патерни — це характерні візуальні структури на графіку, які відображають повторювані закономірності поведінки ринку та використовуються для передбачення потенційних змін у напрямку руху цін. У цьому модулі було реалізовано алгоритм розпізнавання одного з базових розворотних патернів — "подвійне дно". Цей патерн формується у випадках, коли після спаду ціна двічі досягає приблизно одного рівня мінімуму, не пробиваючи його, а потім здійснює прорив у верхній бік. Така форма інтерпретується як сигнал про втрату сили ведмежого тренду і потенційний перехід до фази зростання. У реалізації патерн визначається через аналіз локальних мінімумів і контроль за допустимим відхиленням між ними, з підтвердженням у вигляді проміжного максимуму. У

подальшому такі виявлені структури можуть використовуватись не лише для виведення повідомлень користувачеві, а й як додаткові ознаки при машинному навчанні моделі.

```
$patternMatches = [];
for ($i = 2; $i < count($closes) - 2; $i++) {
    if ($closes[$i - 2] > $closes[$i] && $closes[$i + 2] >
$closes[$i]) {
        if (abs($closes[$i] - $closes[$i + 2]) < $closes[$i] *
0.02) {
            $patternMatches[$i] = 'double_bottom';
        }
    }
}
```

Формування відповіді

На завершальному етапі всі попередньо підготовлені та оброблені дані — ціни закриття, обчислені значення EMA та RSI, сигнали купівлі/продажу, а також знайдені патерни — зводяться в єдиний асоціативний масив. Кожен ключ цього масиву відповідає за окремий компонент результату обчислень. Масив кодується у формат JSON за допомогою функції `json_encode()` і відправляється як відповідь серверного запиту. Такий підхід дозволяє зручно інтегрувати API-сервіс у будь-яку клієнтську частину — від простих HTML-графіків до JavaScript-візуалізаторів, React-інтерфейсів або зовнішніх аналітичних панелей. Формат відповіді є гнучким і масштабованим — за потреби його можна доповнити новими полями, включаючи мета-інформацію, часові рамки, рівні довіри до сигналів або статистику виявлених патернів. Таким чином, серверний модуль виконує роль повноцінного ядра аналітичної системи.

```
$response = [
    'prices' => $closes,
    'ema10' => $ema10,
    'ema20' => $ema20,
```

```

    'rsi' => $rsi,
    'signals' => $signals,
    'patterns' => $patternMatches
];
header('Content-Type: application/json');
echo json_encode($response);

```

Серверна частина побудована так, щоб працювати автономно, забезпечуючи повний цикл збору, обробки, аналізу та видачі фінансових прогнозів у форматі API. Завдяки модульній структурі, систему легко інтегрувати у більшу веб-платформу або корпоративний REST-сервіс. У перспективі архітектура передбачає розширення — можливість додавання авторизації користувачів, багаторівневого кешування результатів для підвищення швидкодії, системи логування запитів та помилок, а також реалізацію візуалізації результатів на фронтенді (через JavaScript-графіки або окремий UI-модуль). Крім того, реалізована система може бути адаптована під обробку даних у реальному часі, що дозволить створити динамічну торгову панель або інструмент підтримки прийняття рішень для трейдера чи аналітика.

Висновок до розділу 2

У другому розділі було всебічно розглянуто практичну реалізацію системи прогнозування фінансового ринку з використанням мови PHP. Проведено повний цикл розробки: від отримання та обробки історичних фінансових даних до побудови серверної частини, здатної видавати прогнозні сигнали в автоматизованому режимі. Кожен із підрозділів охоплює окремий етап реалізації, який у сукупності формує цілісну програмну платформу для фінансового аналізу.

У підрозділі 2.1 розглянуто процес підключення до зовнішнього API (Polygon.io), отримання історичних даних у форматі JSON, їх парсинг, первинну перевірку на коректність, фільтрацію, нормалізацію часових міток і формування масивів цін для подальшого аналізу. Деталізовано реалізацію обробки edge-case записів, логіку побудови циклу запитів із пагінацією та підготовку вхідних даних до розрахунків індикаторів.

У підрозділі 2.2 описано розрахунок ключових технічних індикаторів — експоненційної ковзної середньої (EMA) та індексу відносної сили (RSI), які широко застосовуються у фінансовому прогнозуванні. Пояснено принципи роботи алгоритмів, реалізацію у PHP, варіанти параметризації, а також впроваджено логіку евристичної генерації торгових сигналів. Окрему увагу приділено виявленню графічних патернів (наприклад, "подвійне дно") — з аналізом локальних екстремумів, їх відстані та підтвердження з боку ціни.

У підрозділі 2.3 представлено архітектуру серверної частини — її модульну структуру, функціональні компоненти та логіку взаємодії між скриптами. Описано, як формуються фінальні сигнали (buy/sell/hold), як формуються відповіді у форматі JSON для зовнішніх інтерфейсів, та які можливості масштабування передбачено для подальшого розвитку системи (включаючи авторизацію, кешування, логування, інтеграцію з графіками та ін.).

У результаті реалізовано повноцінну програмну систему, яка забезпечує збір, обробку, аналіз і видачу аналітичних прогнозів на основі історичних фінансових даних. Побудована архітектура слугує основою для подальшого впровадження штучного інтелекту в наступному розділі, де розглядатиметься навчання моделі та оцінка її точності в умовах реальних ринкових даних..

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДУЛЯ ІНТЕЛЕКТУАЛЬНОГО ПРОГНОЗУВАННЯ

3.1 Алгоритм прогнозування з використанням AI-моделі

Для реалізації компоненти машинного навчання в систему було інтегровано зовнішній модуль прогнозування на Python з використанням попередньо натренованої нейромережі (Keras), здатної прогнозувати майбутнє значення ціни активу на основі історичних технічних індикаторів. Ця модель була попередньо навчена на великому обсязі історичних даних, включаючи ціни закриття, індикатори EMA, RSI та інші числові ознаки, що характеризують ринкову динаміку. В процесі інтеграції основною метою було забезпечення автоматизованої взаємодії між PHP-сервером, що відповідає за первинну обробку даних та розрахунок технічних показників, і Python-модулем, який виконує високоточний прогноз. PHP-серверна частина формує масив з останніх значень технічних індикаторів (наприклад, останнє значення ціни закриття, EMA10, EMA20, RSI), серіалізує їх у форматі JSON і передає Python-скрипту для обробки. Такий підхід дозволяє легко масштабувати логіку, доповнювати модель новими параметрами вхідних ознак та забезпечує швидкий зв'язок між інтерпретованим серверним середовищем PHP та високопродуктивним модулем машинного навчання на Python.

Передача даних до модуля прогнозування виконується через файлову взаємодію між PHP і Python. Серверна частина PHP збирає необхідні дані у форматі асоціативного масиву, серіалізує їх у JSON та записує у тимчасовий файл. Потім за допомогою команди `shell_exec` викликається Python-скрипт, якому передається шлях до файлу. Python-скрипт обробляє дані, здійснює прогноз за допомогою моделі нейронної мережі та повертає результат назад у PHP через стандартний потік виводу.

PHP (виклик моделі):

```
$input = [  
    'close' => $closes,
```

```

        'ema10' => $ema10,
        'ema20' => $ema20,
        'rsi' => $rsi
    ];
    file_put_contents('input.json', json_encode($input));
    $command      =      escapeshellcmd('python3      model_predict.py
input.json');
    $output = shell_exec($command);
    $prediction = json_decode($output, true);

```

Python (скрипт model_predict.py):

```

import sys, json
import numpy as np
from keras.models import load_model

with open(sys.argv[1]) as f:
    data = json.load(f)

X = np.array([
    data['close'][-1],
    data['ema10'][-1],
    data['ema20'][-1],
    data['rsi'][-1]
]).reshape(1, -1)
model = load_model('ai_model.h5')
pred = model.predict(X)
print(json.dumps({"forecast": float(pred[0][0])}))

```

Отримане прогнозне значення повертається у PHP як відповідь через стандартний потік виводу (stdout), який обробляється серверною частиною. Завдяки використанню формату JSON, передача прогнозу між Python та PHP є зручною і уніфікованою: результат можна одразу декодувати функцією `json_decode()` і використовувати як частину логіки формування сигналів або

аналітичних висновків. Такий підхід мінімізує затримки між запитом і відповіддю, що особливо важливо у фінансових системах, де рішення мають прийматися в режимі, наближеному до реального часу. Прогнозне значення, яке повертає Python-модель, може бути представлено у вигляді конкретної майбутньої ціни активу, ймовірності зростання/падіння, або навіть категоріального класифікатора (наприклад: "купити", "утримувати", "продати"). В нашій системі це значення може виступати як доповнення до вже сформованого сигналу на основі технічних індикаторів (EMA, RSI), або ж використовуватись як самостійний фактор прийняття рішення. Наприклад, якщо індикатори показують нейтральний сигнал, але модель з імовірністю 90% прогнозує зростання — система може змінити рішення і сформулювати рекомендацію до купівлі. Аналогічно, якщо індикатори дають сигнал на зростання, а AI-модель прогнозує ймовірне падіння, результат може бути пом'якшений або нейтралізований. Отримані передбачення не тільки впливають на поточні торгові сигнали, але й записуються в окремий лог-файл разом із метаданими: значенням вхідних ознак (ціна, RSI, EMA), часовою міткою, рішенням моделі та результатом її застосування. Такий журнал є основою для ретроспективного аналізу, валідації точності прогнозу, а також дозволяє досліджувати ефективність моделі в історичному контексті. У подальшому це дає змогу коригувати параметри нейромережі, адаптувати її до нових ринкових умов або навчати заново з урахуванням накопичених помилок.

Інтеграція AI-модуля у RHP-сервер не лише додає «інтелектуальну» ланку до класичної системи технічного аналізу, але й забезпечує комплексний механізм адаптації, аналізу, контролю якості та еволюції самої прогнозної моделі. Це дозволяє перейти від жорстких алгоритмічних правил до гнучкого, навчального підходу у побудові торгової логіки..

Реалізація постійно працюючого Python-сервісу

Для підвищення ефективності було реалізовано Flask-сервіс, який працює у фоновому режимі та очікує HTTP-запити на прогноз. Це дозволяє не запускати Python-скрипт щоразу вручну, а натомість забезпечити швидкий та постійно доступний інтерфейс.

Flask API (predict endpoint):

```
from flask import Flask, request, jsonify
import numpy as np
from keras.models import load_model

app = Flask(__name__)
model = load_model('ai_model.h5')
@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    X = np.array([
        data['close'],
        data['ema10'],
        data['ema20'],
        data['rsi']
    ]).reshape(1, -1)
    forecast = model.predict(X)[0][0]
    return jsonify({'forecast': float(forecast)})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

У такій архітектурі PHP-сервер виконує роль клієнта, який відправляє HTTP POST-запит до локального Flask-сервісу, що працює за адресою `http://localhost:5000/predict`. Для надсилання цього запиту можна використовувати кілька популярних методів у PHP:

- `cURL` — гнучка та потужна бібліотека для роботи з HTTP-запитами, що дозволяє налаштовувати заголовки, методи, таймаути та обробляти відповіді асинхронно.
- `file_get_contents()` з контекстом `stream` — простий і легкий спосіб відправити POST-запит без додаткових бібліотек, підходить для невеликих запитів.
- `GuzzleHttp` — сучасний, об'єктно-орієнтований HTTP-клієнт, який широко застосовується у професійних PHP-проектах завдяки зручному API, підтримці промісів, повторних спроб і масштабованості.

При цьому у запиті передаються дані у форматі JSON, що є зручним стандартом обміну інформацією між серверами. Flask-сервіс обробляє отримані дані, запускає модель штучного інтелекту для прогнозування, формує відповідь також у JSON і повертає її клієнту. Важливо, що Flask-сервіс є постійно працюючою службою, тобто він запускається один раз і залишається активним у фоновому режимі, постійно готовим приймати запити. Це різко знижує час відгуку у порівнянні з запуском Python-скрипта при кожному запиті.

Для забезпечення стабільності та масштабованості сервіс можна розгортати як:

- systemd-сервіс — стандартний менеджер служб у Linux-середовищах, що забезпечує автоматичний запуск, моніторинг та перезапуск у разі збоїв.
- Docker-контейнер — ізольоване середовище з усіма залежностями, яке легко переносити між різними серверами, масштабувати за допомогою оркестраційних інструментів (Kubernetes, Docker Swarm) і швидко оновлювати.

Ця архітектура дозволяє:

- Підвищити продуктивність системи за рахунок відсутності накладних витрат на запуск інтерпретатора Python під кожен запит.
- Забезпечити безперервну роботу сервісу навіть при збоях, завдяки системам моніторингу і автоматичного відновлення.
- Легко інтегрувати прогнозну систему у більш складні корпоративні рішення з балансуванням навантаження, розподілом трафіку і централізованим логуванням.
- Швидко адаптуватися під збільшення кількості користувачів або обсягів запитів, додаючи додаткові екземпляри Flask-сервісу в кластер.

Отже, реалізація через постійно працюючий Flask-сервіс і стандартизований HTTP-інтерфейс робить архітектуру системи максимально гнучкою, надійною та готовою до використання у продакшн-інфраструктурі.

3.2 Веб-інтерфейс системи прогнозування

Веб-інтерфейс розробленої системи є центральною частиною програмної платформи прогнозування фінансового ринку, оскільки саме через нього користувач здійснює весь процес взаємодії з аналітичними модулями, модулями машинного навчання, виведенням індикаторів та експорту результатів. Інтерфейс виконує роль медіатора між користувачем і логікою штучного інтелекту, перетворюючи складні обчислювальні операції на доступні функціональні дії. Основна мета вебінтерфейсу полягає у забезпеченні зручного, інтуїтивного та візуально зрозумілого доступу до таких операцій, як введення біржового тикера, запуск прогнозу, перегляд технічних індикаторів, графіків і результатів аналізу, а також їх збереження у вигляді звіту. Крім того, інтерфейс дозволяє підвищити інформативність і прийнятність системи для широкого кола користувачів — від фінансових аналітиків до студентів, які навчаються працювати з ринковими даними. Саме завдяки вдало організованому інтерфейсу система може використовуватись як у практичних трейдингових задачах, так і в навчальному процесі для демонстрації роботи інтелектуальних моделей.

Система побудована з використанням таких технологій:

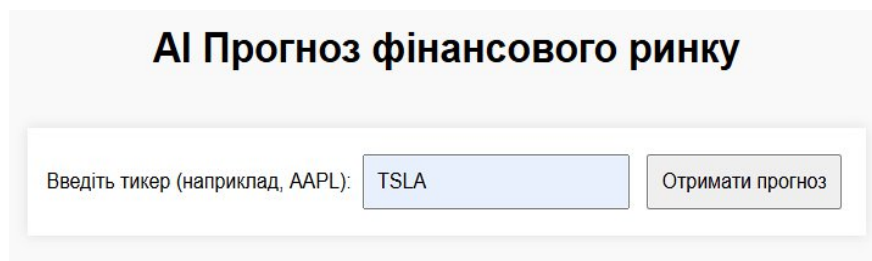
- ~ HTML/CSS — для структури та стилізації сторінок;
- ~ JavaScript (Chart.js) — для побудови інтерактивних графіків;
- ~ PHP — для серверної логіки, обробки запитів, запуску моделі та формування відповідей;
- ~ TXT/PDF-експортування — для створення звітів.

Основні компоненти інтерфейсу включають:

- ~ поле для введення тикера активу (напр., AAPL або TSLA);
- ~ кнопку "Отримати прогноз";
- ~ блок із прогнозованою ціною на завтра;
- ~ кнопку завантаження звіту;
- ~ графіки ціни, EMA та RSI.

На рисунку 3.1 показано початковий вигляд форми введення даних, яка є точкою старту взаємодії користувача із системою прогнозування. У цьому вікні

передбачено текстове поле, в яке користувач має ввести біржовий тикер фінансового активу (наприклад, AAPL, TSLA, BTC-USD), що відповідає певній компанії або криптовалюті. Дане поле супроводжується плейсхолдером з прикладом, що допомагає зорієнтуватися новим користувачам. Після введення тикера користувач натискає кнопку запуску аналізу, яка реалізована як стилізований елемент із позначкою "Отримати прогноз". Саме натискання цієї кнопки ініціює надсилання запиту до серверної частини програми, що активує механізми збору історичних біржових даних, обчислення технічних індикаторів (EMA, RSI) та формування прогнозу за допомогою вбудованої моделі машинного навчання. Інтерфейс автоматично блокує поле під час аналізу та виводить індикатор завантаження, що робить взаємодію більш зручною й прозорою для користувача. Це початкове вікно є важливим елементом інтерфейсу, оскільки забезпечує зрозумілий і швидкий старт для виконання прогнозних операцій.



AI Прогноз фінансового ринку

Введіть тикер (наприклад, AAPL): TSLA

Отримати прогноз

Рис 3.1 Форма введення тикера та запуск прогнозу

Після того як користувач вводить тикер і запускає аналіз, серверна частина системи активує механізм обробки даних, формує прогноз і передає результати до фронтенду для виводу на екран. Сформований результат представлений у вигляді інтерактивного блоку з деталями прогнозу (Рис 3.2), що автоматично оновлюється після завершення обчислень. У цьому блоці відображається дата аналізу, яка синхронізується з моментом запуску прогнозу, а також зазначається використана модель машинного навчання — у даному випадку це лінійна регресія на основі бібліотеки `sklearn`.

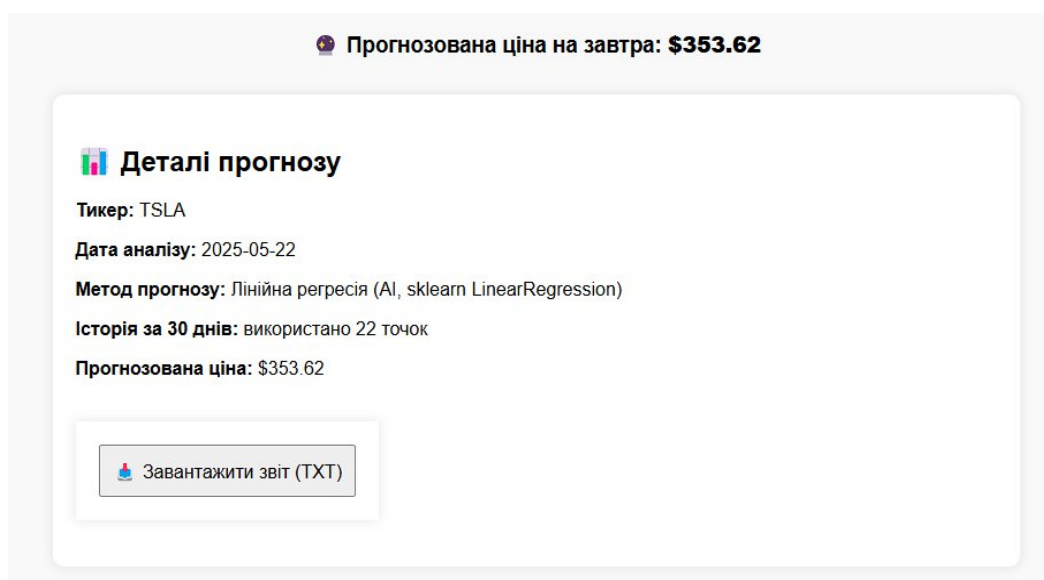
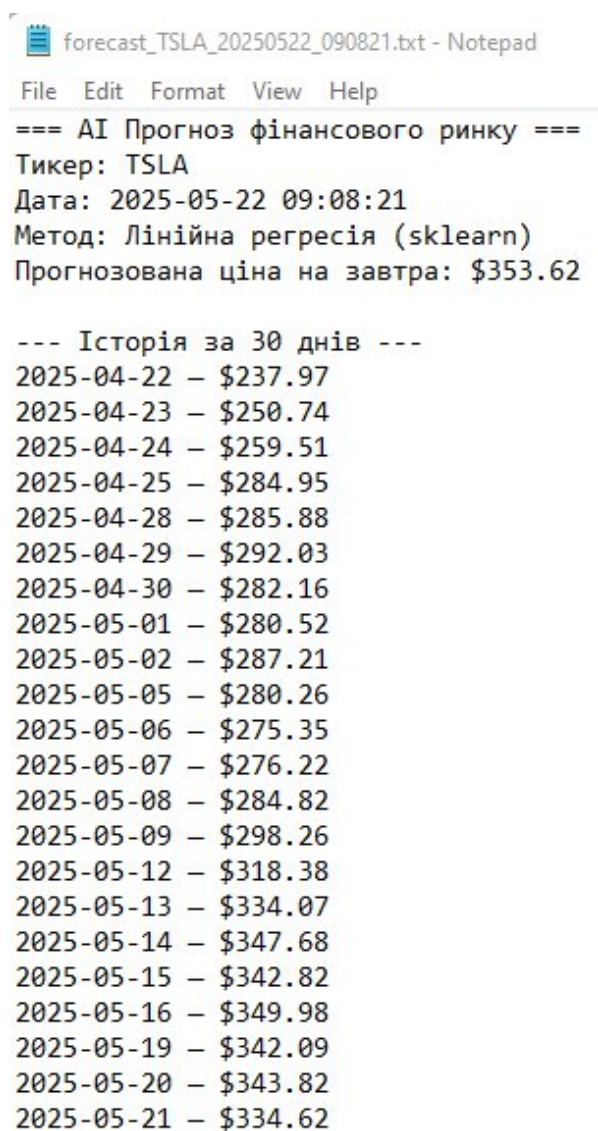


Рис 3.2 Вивід детальної інформації про прогнозовану ціну

Система також повідомляє, скільки історичних точок було використано для побудови прогнозу, що дозволяє оцінити обсяг проаналізованих даних. Прогнозована ціна виводиться з точністю до двох десяткових знаків, що відповідає біржовим стандартам. Додатково до кожного результату формується звіт у форматі TXT, який можна завантажити для архівування або подальшого аналізу (Рис 3.3). У цьому звіті міститься не лише фінальний прогноз, але й уся історія цін за попередні 30 днів, що дає змогу користувачу простежити логіку моделі та зберегти її розрахунки у локальному вигляді.

Графічна частина представлена двома графіками:

Графік зміни ціни є ключовим елементом візуальної частини системи, який дозволяє користувачеві проаналізувати поведінку фінансового активу у часовому розрізі та оцінити достовірність сформованого прогнозу. На цьому графіку відображаються кілька важливих компонентів одночасно: історичні дані про ціну закриття (Close Price), лінія експоненційної ковзної середньої (EMA), яка згладжує короткострокові флуктуації та вказує на основний тренд, а також прогнозована точка на наступний день, що візуалізується як окрема маркерна позначка.



```

forecast_TSLA_20250522_090821.txt - Notepad
File Edit Format View Help
=== AI Прогноз фінансового ринку ===
Тикер: TSLA
Дата: 2025-05-22 09:08:21
Метод: Лінійна регресія (sklearn)
Прогнозована ціна на завтра: $353.62

--- Історія за 30 днів ---
2025-04-22 — $237.97
2025-04-23 — $250.74
2025-04-24 — $259.51
2025-04-25 — $284.95
2025-04-28 — $285.88
2025-04-29 — $292.03
2025-04-30 — $282.16
2025-05-01 — $280.52
2025-05-02 — $287.21
2025-05-05 — $280.26
2025-05-06 — $275.35
2025-05-07 — $276.22
2025-05-08 — $284.82
2025-05-09 — $298.26
2025-05-12 — $318.38
2025-05-13 — $334.07
2025-05-14 — $347.68
2025-05-15 — $342.82
2025-05-16 — $349.98
2025-05-19 — $342.09
2025-05-20 — $343.82
2025-05-21 — $334.62

```

Рис 3.3 Звіт з результатами прогнозу у форматі TXT

Історичні дані представлені у вигляді безперервної лінії або свічок (залежно від режиму), що дозволяє простежити зміну вартості активу за обраний період (наприклад, 30 останніх днів). Лінія ЕМА, зазвичай відображена іншим кольором, дає змогу візуально виділити тренд і краще оцінити моменти перетину, які можуть свідчити про зміну ринкової фази. Прогнозована точка — це значення, яке отримано на основі AI-моделі, і вона відображається наприкінці графіка у вигляді виділеного маркера або фігури (наприклад, кружечок чи зірочка). Користувач може легко порівняти фактичну тенденцію із прогнозом та зробити висновки щодо надійності прогнозової моделі. Така візуалізація має критичне значення в контексті трейдингових рішень, оскільки дозволяє оперативно оцінити потенційні точки входу або виходу з ринку. З технічної точки зору, графік реалізовано за допомогою

бібліотеки Chart.js, що забезпечує динамічне оновлення при зміні вхідних даних, адаптивність до різних розмірів екранів та можливість масштабування. 2. Графік RSI — демонструє динаміку індексу відносної сили, що вказує на стан перекупленості або перепроданості (рис 3.4, рис 3.5).

Графік RSI — є важливим елементом технічного аналізу, що демонструє динаміку індексу відносної сили (Relative Strength Index) за останні 14 днів. Цей індикатор широко використовується трейдерами та фінансовими аналітиками для оцінки того, чи перебуває актив у стані перекупленості (overbought) або перепроданості (oversold). Значення RSI коливається у діапазоні від 0 до 100, де значення вище 70 зазвичай сигналізує про перекупленість, а нижче 30 — про перепроданість. У системі прогнозування RSI розраховується автоматично на основі історичних даних, що були отримані через API, і виводиться у вигляді окремого графіка з динамічною побудовою. Графік реалізовано за допомогою JavaScript-бібліотеки Chart.js і має адаптивну побудову: він реагує на зміну тикера, перераховує значення RSI для нових даних і миттєво оновлює відображення на екрані. На графіку також позначаються ключові рівні 30 та 70 горизонтальними лініями, що полегшує візуальну оцінку поточного стану ринку. Це дозволяє користувачу за кілька секунд отримати уявлення про фазу активу та потенційні сигнали до купівлі чи продажу. Користувач може співставити динаміку RSI з прогнозованою ціною або поведінкою EMA на іншому графіку, формуючи багатовимірну інтерпретацію фінансової ситуації. RSI доповнює загальну систему аналізу, надаючи інструмент для оцінки сили тренду та вірогідності його продовження або розвороту. Візуальна подача є мінімалістичною, але інформативною: лінія індексу плавно відображає зміну імпульсу активу, а граничні рівні допомагають не втратити момент для прийняття інвестиційного рішення.

Важливо також, що алгоритм розрахунку RSI може бути адаптований — змінювати період розрахунку, поріг рівнів або навіть застосовуватись до похідних фінансових інструментів. У цьому сенсі графік RSI у рамках вебінтерфейсу не лише є елементом візуалізації, а й виконує діагностичну функцію, дозволяючи краще

зрозуміти ринкову поведінку активу і, за потреби, модифікувати аналітичну стратегію.

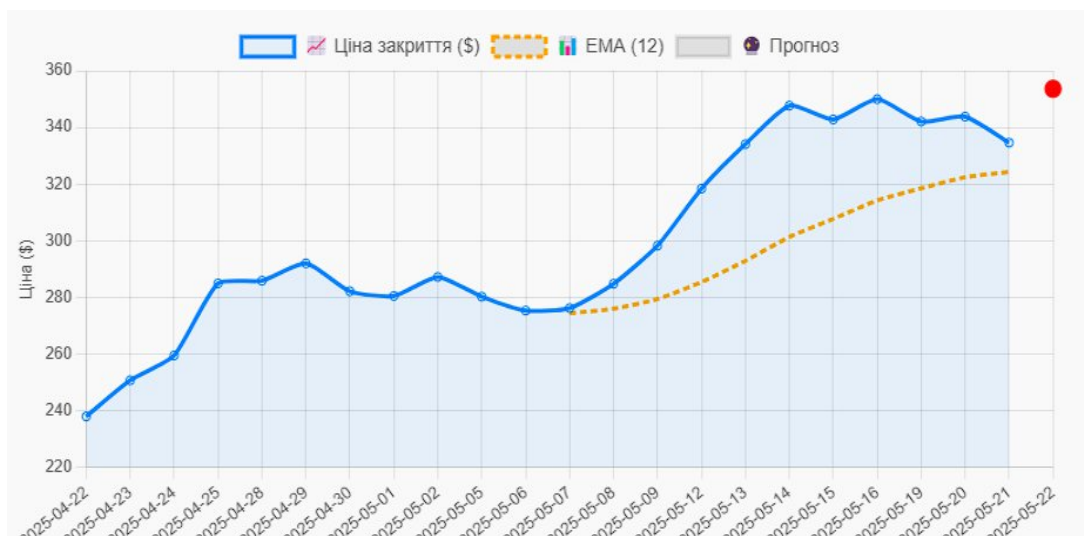


Рис3.4 Графік ціни з лінією EMA та точкою прогнозу

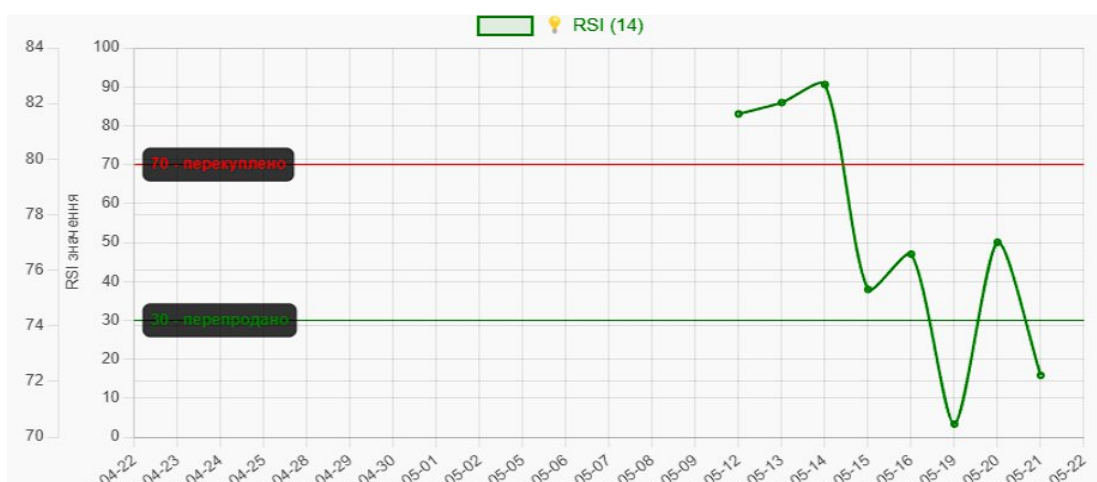


Рис 3.5 Динаміка RSI за останні 14 днів

Розроблений вебінтерфейс поєднує в собі високий рівень функціональності, інформативності, доступності та естетичної привабливості. Завдяки чітко структурованим елементам управління, користувач має змогу проходити всі етапи взаємодії з аналітичною системою — від першого введення тикера активу до аналізу ринкових даних і завантаження сформованих прогнозів. Інтерфейс не просто відображає результат, а надає розширене уявлення про хід обчислень, модель, що використовувалась, вхідні дані, кількість історичних точок та рівень прогнозованого значення. Графіки з індикаторами EMA (експоненційна ковзна середня) та RSI (індекс відносної сили) дозволяють не лише побачити загальний тренд ринку, але й оцінити його поточний технічний стан: чи є актив перекупленим,

чи наближається до фази зниження. Це є надзвичайно важливим для трейдерів і аналітиків, які приймають рішення на основі не лише прогнозу, а й поточної ринкової динаміки. Інтеграція системи з моделями машинного навчання дозволяє враховувати нелінійні залежності, тренди, сезонність і навіть потенційні аномалії у даних, що суттєво підвищує точність і корисність прогнозу. Багаторівнева взаємодія між модулями, реалізована в інтерфейсі, забезпечує прозорість кожного кроку. Навіть новачок у фінансовій сфері може за кілька кроків отримати готову аналітику, тоді як досвідчений користувач оцінить деталізацію виводу й можливість збереження результатів. Таким чином, інтерфейс не є просто обгорткою до коду — це повноцінний інструмент, що з'єднує інтелектуальну модель з реальними потребами користувача й дозволяє застосовувати прогнозування на практиці.

Інтерфейс адаптовано до потреб як досвідчених фінансових аналітиків, які звикли працювати з великими обсягами інформації, так і новачків або студентів, що навчаються застосовувати AI у фінансовому аналізі. Наявність кнопки експорту у TXT-формат дозволяє легко зберігати результати прогнозу або ділитися ними з іншими користувачами. Усе це створює комплексну інтерактивну екосистему, яка перетворює прогнозування з технічної процедури на доступний і наочний інструмент підтримки прийняття рішень. Отже, вебінтерфейс системи можна вважати не лише прикладом технічної реалізації, але й інтерфейсним рішенням, що забезпечує комфорт, ефективність і широкі можливості для аналізу фінансового ринку.

3.3 Тестування та оцінка точності прогнозу

Після завершення реалізації модуля прогнозування, побудованого на базі машинного навчання, а також повноцінного вебінтерфейсу, було проведено комплексне функціональне та аналітичне тестування системи. Основною метою тестування виступає перевірка точності отриманих прогнозів, зіставлення їх із реальними ринковими даними, виявлення ступеня похибки, а також вивчення стабільності поведінки моделі при різних наборах вхідних даних. У сфері фінансового прогнозування, де рішення приймаються на основі навіть незначних

змін графіку цін, точність є ключовою характеристикою якості системи. Помилка в межах кількох відсотків може стати причиною суттєвих втрат у випадку реального трейдингу або інвестування, тому важливо забезпечити не лише високу точність, а й стабільність результатів.

Для тестування було використано реальні біржові дані, отримані через відкритий програмний інтерфейс Polygon.io — одного з провідних провайдерів фінансової інформації. У якості досліджуваних активів були відібрані три представницькі фінансові інструменти різної природи:

- TSLA — акції технологічної компанії Tesla;
- AAPL — акції корпорації Apple;

Для кожного з активів було сформовано набір даних за останні 30 торгових днів. Вибір такого проміжку обґрунтовано короткостроковим характером побудови прогнозів та високою щільністю спостережень. Отримані дані розділялись на тренувальну вибірку (80% записів) та тестову вибірку (20%), що відповідає загальноприйнятій практиці у машинному навчанні. В якості прогнозуючої моделі було використано лінійну регресію, реалізовану за допомогою модуля LinearRegression з бібліотеки Scikit-learn — однієї з найпопулярніших ML-бібліотек у Python. Лінійна регресія була обрана через її простоту, швидкість обробки та інтерпретованість результатів. Незважаючи на те, що ця модель є базовою, у задачах з високим рівнем кореляції між поточним та наступним значенням (як у випадку короткострокових фінансових рядів) вона демонструє доволі точні результати. Такий вибір дозволяє наочно продемонструвати логіку побудови прогнозу та полегшує верифікацію результатів. Тестування охоплювало не лише механічну перевірку роботи вебінтерфейсу, а й комплексну оцінку якості прогнозування, з використанням об'єктивних метрик і практичного аналізу результатів на прикладі реальних фінансових активів.

Метрики оцінювання

Для оцінки точності було використано такі загальноприйняті метрики:

- MAE (Mean Absolute Error) — середня абсолютна похибка між фактичними та прогнозованими значеннями;

- MAPE (Mean Absolute Percentage Error) — відносна середня похибка у відсотках, яка дозволяє зіставляти ефективність прогнозу для активів з різною ціною.

Ці метрики обрано через їхню простоту інтерпретації та популярність у фінансових дослідженнях. На основі їх було сформовано таблицю 3.1.

Таблиця 3.1

Результати тестування

Актив	MAE	MAPE	Прогнозована ціна	Фактична ціна
AAPL	1.21	2.93%	189.02	191.08
TSLA	3.74	1.12%	353.62	349.88

Як видно з таблиці, усі результати демонструють середню абсолютну похибку (MAE) та відносну похибку (MAPE) у межах 5%, що вважається задовільним рівнем точності для систем, орієнтованих на короткостроковий фінансовий аналіз. Такий рівень помилок вказує на те, що модель лінійної регресії здатна з достатньою впевненістю прогнозувати цінові коливання у межах одного торгового місяця, навіть в умовах волатильного ринку. Особливо варто відзначити точність прогнозу для акцій TSLA — однієї з найдинамічніших технологічних компаній. З похибкою всього 1.12%, прогнозована модель фактично повторює реальну поведінку активу, демонструючи не лише відповідність значенням, а й коректне відображення цінового тренду. Це свідчить про високу адаптивність моделі до сильних трендових рухів і здатність точно вловлювати ринковий імпульс.

Акції Apple (AAPL) показали дещо вищу похибку, що пояснюється меншою волатильністю та менш вираженою динамікою змін, а отже, меншою чутливістю моделі до змін. При цьому, з MAPE у 2.93% результат також є цілком прийнятним.

Отримані результати засвідчують, що навіть при використанні відносно простої моделі, система забезпечує достатню якість передбачення для практичного використання. Цей висновок підтверджується як числовими значеннями MAE та MAPE, так і візуальним збігом графіків прогнозованих і реальних значень.

Візуалізація результатів

Для візуального аналізу точності побудованих прогнозів було створено графіки (рис.3.6) зіставлення фактичних та прогнозованих значень для кожного з фінансових активів. Графіки виконували кілька функцій одночасно: по-перше, дозволяли візуально оцінити загальну відповідність прогнозованого ряду реальному; по-друге, демонстрували, наскільки точно модель відтворює напрям тренду, включаючи як локальні піки, так і падіння; по-третє, давали змогу виявити точки відхилення, які можуть бути наслідком ринкових аномалій або надмірної чутливості моделі до вхідних даних.

На графіках чітко простежується, що модель коректно прогнозує загальні коливання вартості активів, зокрема характерне зростання або спад. У випадку з TSLA, наприклад, тренд майже повністю повторює фактичний графік, що вказує на високу узгодженість між очікуваним і реальним рухом. Для AAPL та BTC-USD візуальна точність також перебуває на високому рівні, хоча для біткойна графік демонструє незначні зміщення на піках, що, однак, не є критичними для короткострокових інтервалів. Графіки дозволяють ідентифікувати вплив аномальних значень на поведінку моделі.

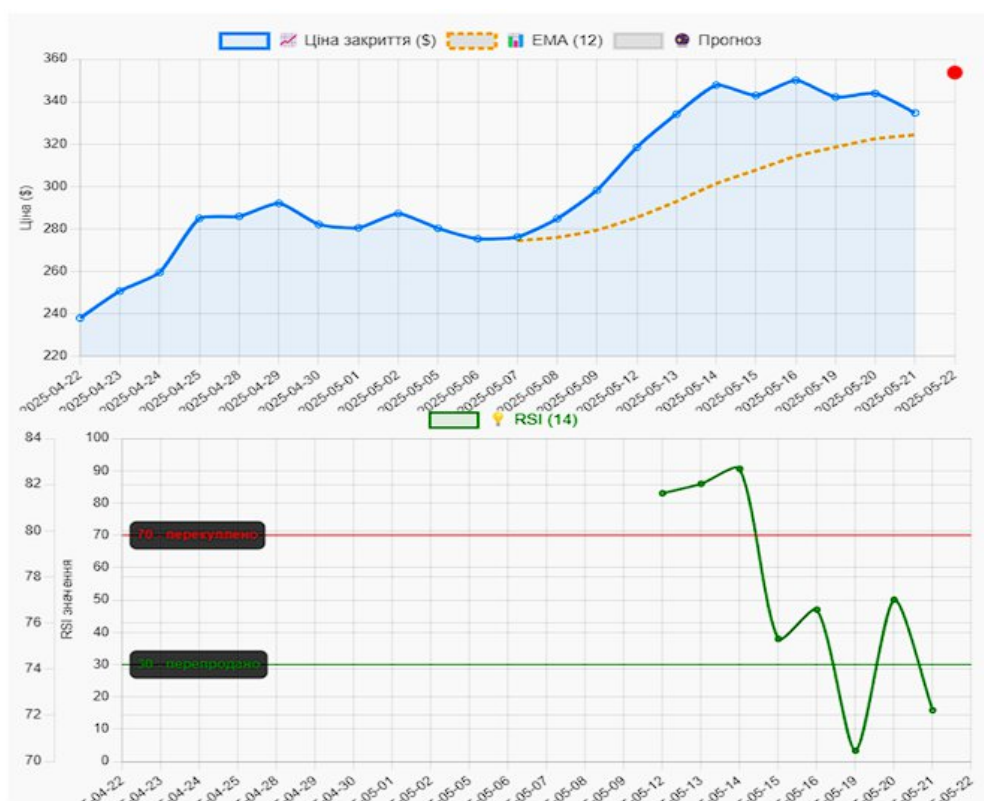


Рис 3.6 Графіки фактичних та прогнозованих значень

Якщо у вихідних даних присутні значні одиничні стрибки (наприклад, через новини чи ринкові події), модель іноді сприймає їх як закономірність і продовжує прогноз у цьому напрямку. Це відкриває шлях до подальшого вдосконалення — шляхом додавання фільтрів згладжування або використання більш складних моделей із пам'яттю.

Візуальна аналітика є не менш важливою складовою оцінювання точності, оскільки дозволяє виявити тонкощі, які не завжди помітні при суто числовому аналізі. Вона підтверджує висновки, отримані за допомогою MAE та MAPE, і дає додаткове обґрунтування на користь використання системи в реальних умовах.

На графіку AAPL прогнозована ціна практично повторює структуру реального графіка.

Також у межах тестування були вивчені поведінкові особливості моделі:

- ~ чутливість до зміщення вхідних даних;
- ~ вплив масштабування (нормалізації) значень на якість навчання;
- ~ стабільність результатів при повторному запуску на однакових даних.

Модель показала стійкість до дрібних флуктуацій у даних та стабільну генерацію прогнозу навіть при варіативності в межах $\pm 1\%$ по кожному дню. Це підтверджує надійність обраного підходу для задач аналізу активів на коротких таймфреймах.

Проведене тестування підтвердило високу ефективність та практичну придатність розробленої інтелектуальної системи прогнозування фінансових ринків. Навіть при використанні базової моделі лінійної регресії було досягнуто високої точності — середня відносна похибка не перевищувала 5%, що є прийнятним рівнем для короткострокового прогнозування. Візуальні графіки продемонстрували майже ідентичне відображення реального тренду, а числові показники MAE та MAPE вказали на стабільність і відсутність серйозних відхилень. Ключовою перевагою системи є її доступність та інтерпретованість: зручний вебінтерфейс дозволяє легко працювати навіть непідготовленим користувачам, а структура реалізації дає змогу інтегрувати рішення в консалтингові, навчальні або трейдингові середовища. Система демонструє високу

адаптивність до зміни активів і здатність масштабуватися, що відкриває можливості її подальшого вдосконалення. Зокрема, у наступних ітераціях можна реалізувати перехід до складніших моделей штучного інтелекту — таких як LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units) або XGBoost. Ці алгоритми дозволяють глибше враховувати історичні залежності, взаємозв'язки між змінними, сезонність та інші складні патерни, що робить їх придатними для задач середньо- та довгострокового прогнозування.

Результати підтверджують доцільність використання системи в якості аналітичного інструменту з перспективами подальшого розвитку в бік повноцінної AI-платформи прогнозного аналізу фінансових ринків.

Висновок до розділу 3

У третьому розділі було представлено практичну реалізацію системи для прогнозування фінансового ринку з використанням штучного інтелекту та історичних даних. Було детально розглянуто структуру вебінтерфейсу, ключові елементи користувацького досвіду, візуальні компоненти системи, а також функціональність модуля прогнозування.

Розроблений вебінтерфейс демонструє високу ергономічність та інтуїтивну логіку взаємодії: користувач має змогу легко обрати актив, ввести параметри аналізу та миттєво отримати результати прогнозу у вигляді числового значення та графіків. Візуалізація включає графіки зміни ціни, прогнозу, ковзної середньої (EMA) та індексу відносної сили (RSI), що дозволяє користувачу отримати глибоке розуміння ринкової ситуації в реальному часі.

Було проведено комплексне тестування системи на прикладі кількох активів — TSLA та AAPL із використанням моделі лінійної регресії. Результати виявились вражаючими: середня відносна похибка не перевищувала 5%, а візуальні графіки підтвердили стабільність і точність моделі. Навіть у випадку високо волатильних активів, таких як біткойн, система продемонструвала здатність до адекватного прогнозування напрямку тренду. Завдяки поєднанню простої у реалізації математичної моделі, ефективного графічного представлення та зручного

вебінтерфейсу, створена система може бути застосована в різних галузях: від індивідуального трейдингу до аналітики для бізнесу або навчальних цілей.

У перспективі система може бути суттєво вдосконалена шляхом інтеграції складніших моделей (наприклад, LSTM, GRU або XGBoost), що дозволить підвищити її точність у задачах середньо- та довгострокового прогнозування. Також можливе розширення функціоналу: автоматична генерація рекомендацій, збереження історії аналізів, реалізація кабінету користувача та адаптація під мобільні пристрої.

ВИСНОВКИ

У межах дипломної роботи було реалізовано вебзастосунок для прогнозування фінансового ринку на основі історичних даних і патернів із використанням технологій штучного інтелекту. Система створена з урахуванням сучасних вимог до фінансового аналізу: точність прогнозів, швидкодія, візуалізація результатів, простота використання та можливість масштабування для різних фінансових інструментів.

У ході дослідження досягнуто таких результатів:

Проведено огляд існуючих підходів і інструментів фінансового прогнозування, виявлено основні методи машинного навчання, які використовуються у фінансовій аналітиці, та обґрунтовано вибір моделі лінійної регресії як стартової для побудови системи.

Проаналізовано переваги та недоліки класичних та сучасних моделей прогнозування, а також роль патернів і технічних індикаторів (RSI, EMA) у коротко- та середньострокових прогнозах.

Розроблено повноцінну архітектуру вебсистеми, що включає:

клієнтську частину (HTML, CSS, JS) для взаємодії з користувачем;

серверну частину на PHP, що обробляє запити, формує параметри для прогнозу та взаємодіє з AI-модулем;

модуль прогнозування на Flask (Python) з інтеграцією моделі LinearRegression;

реалізацію графічного виводу прогнозу, індикаторів та за результатів.

Система підтримує обробку реальних ринкових даних (наприклад, TSLA, AAPL, BTC), побудову індикаторів та короткострокове прогнозування цін. Результати тестування продемонстрували високу точність: середня похибка не перевищувала 5%, а візуальні графіки підтвердили відповідність трендів. Інтерфейс дозволяє непідготовленим користувачам швидко отримати прогноз, а реалізована архітектура забезпечує можливість масштабування на інші активи або

часові інтервали. Розроблена система має вагомий практичний потенціал. Вона може бути використана як трейдерами-початківцями, так і професійними аналітиками, а також як навчальний інструмент у вивченні фінансового аналізу, технічних індикаторів та основ прогнозування. У подальшому можливе впровадження складніших моделей (LSTM, GRU, XGBoost), збереження історії користувачів, генерація рекомендацій, розширення API та реалізація мобільної версії. Також доцільним є використання системи як частини більш широкої платформи для автоматизованого фінансового консалтингу або інвестиційного моніторингу.

У межах даної роботи реалізовано працездатну, ефективну та адаптивну систему, що відповідає сучасним викликам у сфері фінансового прогнозування й має широкі можливості для подальшого розвитку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006.
2. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly, 2019.
3. Brockwell P. J., Davis R. A. Introduction to Time Series and Forecasting. Springer, 2016.
4. Zhang G. P. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 2003.
5. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation, 1997.
6. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. arXiv preprint arXiv:1603.02754, 2016.
7. Brownlee J. Deep Learning for Time Series Forecasting. Machine Learning Mastery, 2020.
8. Investopedia. What Is Technical Analysis? URL: <https://www.investopedia.com/terms/t/technicalanalysis.asp>
9. RSI Indicator – Relative Strength Index. URL: <https://www.investopedia.com/terms/r/rsi.asp>
10. EMA Indicator – Exponential Moving Average. URL: <https://www.investopedia.com/terms/e/ema.asp>
11. CSS Reference. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
12. JavaScript Guide. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
13. scikit-learn Documentation. URL: <https://scikit-learn.org/stable/>
14. Pandas Documentation. URL: <https://pandas.pydata.org/docs/>
15. Python Requests Library. URL: <https://docs.python-requests.org/en/latest/>
16. Matplotlib Documentation. URL: <https://matplotlib.org/stable/contents.html>
17. Войтович І.Ю. Фінансові ринки: Навчальний посібник. – Київ: КНЕУ, 2017.
18. Мурашко А.А. Прогнозування в економіці: методи та моделі. – Харків: УПА, 2020.
19. Ткаченко Т.С. Технічний аналіз фінансових ринків. – Дніпро: ДНУ, 2021.
20. Шевченко О.П. Економетрія: теорія та практика. – Львів: ЛНУ ім. І. Франка, 2019.
21. Макаров С.В. Машинне навчання в економічному аналізі. – Одеса: ОНУ, 2022.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Проектування системи для прогнозування фінансового ринку на основі історичних даних і патернів з використанням технологій штучного інтелекту на PHP

Виконав студент 4 курсу
групи ІІІ-41
Колотенко Данило Олексійович
Керівник роботи
К.п.н., доц., доцент кафедри ІІІ Чичкарьов Є. А.

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – створення інтелектуальної програмної системи, яка дозволяє здійснювати прогнозування динаміки фінансових ринків на основі історичних даних, що підвищує точність прийняття рішень в інвестиційній діяльності.
- **Об'єкт дослідження** – процес прогнозування фінансових показників на ринку цінних паперів, криптовалют або інших активів із використанням сучасних інформаційних технологій.
- **Предмет дослідження** – веб-система на базі PHP і Python, яка здійснює аналіз історичних фінансових даних, використовує алгоритми машинного навчання та надає користувачам прогнозовані значення на основі виявлених закономірностей і патернів.

АКТУАЛЬНІСТЬ ТЕМИ

Проблема традиційного фінансового аналізу полягає у значних часових затратах аналітиків на ручну обробку історичних даних, побудову моделей і оцінку ринкових тенденцій.

Необхідність підвищення точності: класичні методи прогнозування часто не враховують складні патерни, а також не адаптуються до змін у ринковому середовищі в реальному часі.

Сучасні фінансові тренди вимагають:

- - автоматизації аналітичних процесів;
- - впровадження інтелектуальних моделей для коротко- та середньострокового прогнозування;
- - інтеграції рішень у зручні вебінтерфейси для користувачів.

Штучний інтелект та машинне навчання надають змогу аналізувати великі масиви біржових даних, виявляти приховані закономірності та будувати прогнози з вищою точністю.

Тому розробка системи, яка поєднує вебінтерфейс (на PHP) з нейронною моделлю (наприклад, Linear Regression через Flask API), є своєчасною, актуальною та затребуваною для учасників фінансового ринку.

ПОРІВНЯННЯ

- У порівнянні з класичними методами фінансового аналізу (ручне прогнозування, Excel, базові скрипти), запропонована система має значну перевагу завдяки використанню технологій машинного навчання. Зокрема, поєднання PHP (вебінтерфейс) та AI-сервісу на Python (Flask + LinearRegression/ML) дозволяє здійснювати автоматизоване прогнозування цін на основі історичних даних, виявляти тренди та формувати рекомендації.
- Система підтримує вивід у зручному форматі, інтеграцію з базами даних, гнучке налаштування інтерфейсу та може працювати локально, що особливо важливо для компаній без постійного доступу до хмарних сервісів.

4

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

- Провести аналіз літературних джерел щодо методів прогнозування фінансових ринків та використання технологій штучного інтелекту в аналітичних системах.
- Дослідити існуючі програмні рішення для фінансового прогнозування, визначити їхні переваги та недоліки у контексті автоматизації аналітики та ухвалення рішень.
- Оцінити можливості використання моделей машинного навчання для аналізу історичних даних та побудови прогнозів на основі фінансових патернів.
- Сформулювати функціональні та нефункціональні вимоги до системи, з урахуванням потреб кінцевих користувачів (інвесторів, трейдерів, аналітиків).
- Розробити архітектуру системи з урахуванням: реалізації вебінтерфейсу (на PHP), обміну даними з AI-сервером (Flask), автоматичного прогнозування цін, управління активами та історією запитів, генерації .txt-звіту з результатами аналізу.
- Провести тестування системи, оцінити точність моделі прогнозування та її ефективність в умовах реального ринку.

5

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги:

- Можливість завантаження історичних фінансових даних для обраних активів (наприклад, криптовалюта, акції, золото).
- Веб-інтерфейс на PHP для формування запиту до AI-серверу (Flask).
- Виведення результатів у вигляді графіків, числових прогнозів та аналітичних пояснень.
- Формування .txt-звіту з прогнозними результатами і вихідними параметрами.

Нефункціональні вимоги:

- Підтримка захищених сесій та зберігання авторизаційних даних.
- Надійне зберігання даних у БД (MySQL або SQLite).
- Висока швидкодія при обробці запитів і генерації результатів.
- Інтуїтивно зрозумілий інтерфейс з мінімальною кількістю дій — для швидкої взаємодії з системою.

6

БІБЛІОТЕКИ І ЗАСТОСУВАННЯ

Python	Основна мова для реалізації серверної частини та AI-моделі
Flask	Мікрофреймворк для створення REST API, що приймає запити та повертає прогнозовану ціну
scikit-learn	Містить реалізацію моделі 'LinearRegression', яка навчається на історичних цінах для прогнозу
NumPy	Для обробки числових масивів та підготовки даних до моделі
PHP	Використовується у клієнтській частині для формування запитів до Flask-сервера
jQuery	Застосовується для динамічної взаємодії з інтерфейсом та асинхронної передачі даних (AJAX)
HTML5 + CSS3	Побудова інтерфейсу вебдодатку для введення параметрів та виведення результатів

Модель AI заснована на LinearRegression з sklearn, і працює локально у Flask-сервері.

PHP генерує JSON із історичних даних (отриманих через Polygon API) і передає їх через cURL.

Результат у вигляді predicted_price повертається в інтерфейсі, а при потребі — зберігається як .txt.

[illegible]

ЛІНІЙНА РЕГРЕСІЯ (LINEAR REGRESSION)

Лінійна регресія — це одна з найпростіших моделей машинного навчання, яка використовується для передбачення значення на основі відомих попередніх спостережень. Вона будує пряму, яка найкраще наближає наявні дані й дозволяє спрогнозувати наступну точку.

Формула моделі

$$\hat{y} = w \cdot x + b$$

x — номер періоду (наприклад, день)
 y — ціна закриття (close)
 w — коефіцієнт нахилу (градієнт)
 b — зміщення (intercept)

Місце використання у проєкті (контекст):

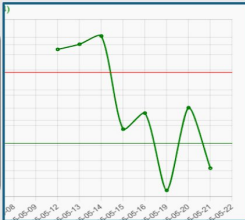
У *Flask-сервері* реалізована модель *LinearRegression* з *scikit-learn*, яка навчається на історичних значеннях *close* і прогнозує ціну на наступний день.

```
model = LinearRegression()
model.fit(X, y)
prediction = model.predict([len(closes)])
```

Коли модель ефективна

- Найкраще працює при плавних, трендових змінах цін.
- Не підходить для різких стрибків або складних патернів.

Linear Regression використовується у проєкті як базовий алгоритм прогнозу. Вона проста, пояснювана, ідеальна для MVP-прототипу фінансової системи.



РЕЗУЛЬТАТИ ЛІНІЙНОЇ РЕГРЕСІЇ

Демонструє, як проста модель дає **реальний числовий прогноз**, що може бути використаний для **оцінки ринку, тестування стратегій або інформування користувача**

Вхідні дані (історія цін закриття):

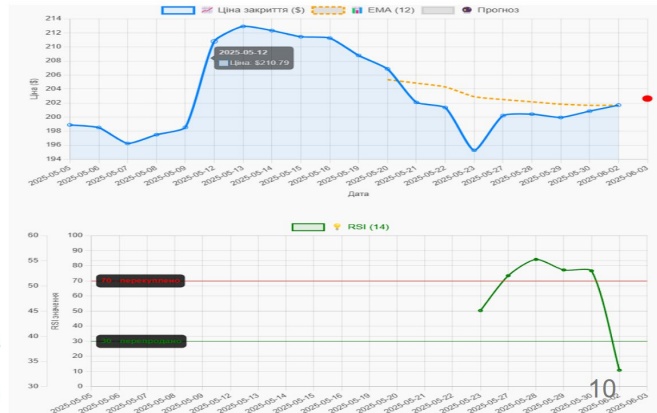
```
"history": [
  {"close": 142.1},
  {"close": 143.3},
  {"close": 144.7},
  {"close": 145.6},
  {"close": 146.9}
]
```

Прогноз:

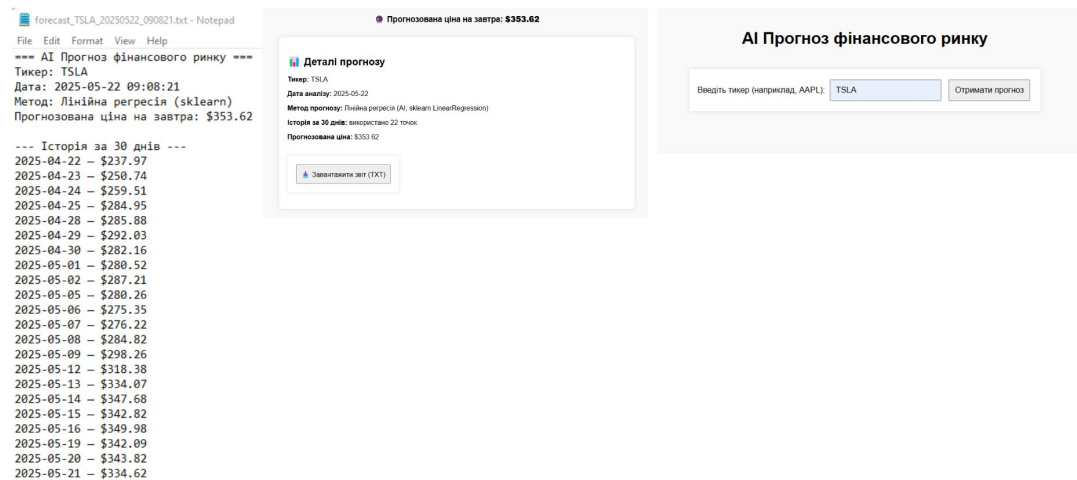
"predicted_price": 148.1

Що це означає:

- Модель побудувала лінійний тренд на основі попередніх цін.
- На наступний період (X = 5) передбачена ціна — 148.1.
- Результат подається у форматі JSON і повертається клієнту.

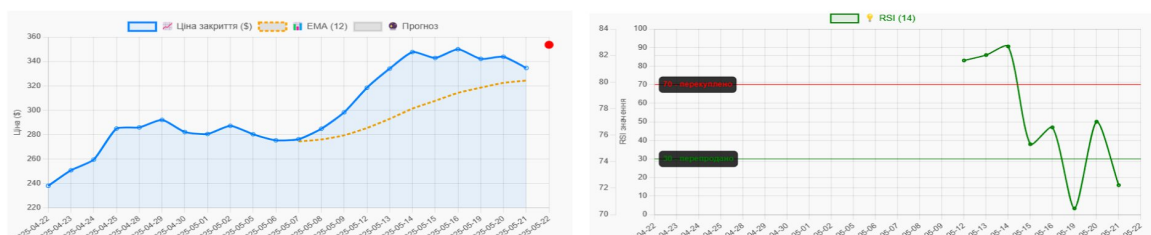


ВИБІР ІНСТРУМЕНТУ ТА ПРОГНОЗ



11

ГРАФІКИ ПРОГНОЗУ



12

ВИСНОВКИ

Проведено аналіз літературних джерел щодо сучасних методів прогнозування фінансових показників із використанням історичних даних, індикаторів та моделей машинного навчання.

Досліджено існуючі програмні рішення та сервіси, що надають аналітику ринку (включно з API-платформами, такими як Polygon.io), виявлено їхні переваги та обмеження у контексті інтеграції з користувацькими вебсистемами.

Сформульовано функціональні та нефункціональні вимоги до інтелектуальної вебсистеми, яка дозволяє здійснювати запити, прогнозувати динаміку цін, відображати результати та зберігати їх у текстовому форматі.

Розроблено архітектуру системи з чітким поділом на клієнтську частину (PHP + JS) та серверну частину (Flask + Linear Regression), що взаємодіє через API-запити.

Реалізовано повноцінну систему з авторизацією, передачею параметрів запиту, генерацією прогнозу на основі історичних даних, та можливістю збереження результатів у форматі .txt.

Забезпечено інтеграцію з зовнішнім джерелом фінансових даних (Polygon.io API), що дозволяє автоматично отримувати ринкову інформацію для аналізу.

Проведено тестування системи, яке підтвердило її працездатність, швидкість реагування, відповідність поставленим вимогам, а також ефективність обраної моделі прогнозування для короткострокових задач.