

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

(повна назва)

Кафедра Системотехніки

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Інтелектуальні технології аналізу динаміки цін криптовалют

(тема)

Виконав:

здобувач 2 року навчання,

групи ІТІм-24-2

Владислав МАЦЮЦЬКИЙ

(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки

(код і назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології

проекткування

(повна назва освітньої програми)

Керівник доцент каф. СТ Інна УРНЯЄВА

(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри

(підпис)

Ігор ГРЕБЕННИК

(прізвище, ініціали)

2025 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



Владислав МАЦЮЦЬКИЙ

(дата, підпис, прізвище студента)

21.12.2025

Кваліфікаційна робота не містить відомостей, заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено.

Керівник кваліфікаційної роботи



I.A. Урняєва

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра СистемотехнікиРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування

(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Мацюцькому Владиславу Руслановичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інтелектуальні технології аналізу динаміки цін криптовалют
затверджена наказом по університету від "24" листопада 2025 р. № 1058 Ст
2. Термін здачі студентом роботи до екзаменаційної комісії 24.12.2025
3. Вихідні дані до роботи розробити інформаційну систему та програмні рішення для аналізу та прогнозування криптовалютних ринків. Система повинна бути реалізована у вигляді веб-додатку та надавати користувачеві інтерфейс доступу до всіх функцій аналітики, обробки ринкових даних і прогнозування цін криптовалют. Перелік використовуваних програмних засобів: Microsoft Windows 10, Python, FastAPI, TensorFlow, XGBoost, LightGBM, Docker, IntelliJ IDEA, Draw.IO, AllFusion ERwin Data Modeler, Process Modeler.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз предметної області, інформаційні системи у сфері аналізу динаміки фінансових даних та криптовалютних ринків, огляд існуючих розробок та підходів, аналіз існуючих підходів, постановка задачі дослідження, розробка вимог до розроблювальної системи, аналіз функціональних вимог до системи, розробка моделі потоків даних

системи, розробка системних вимог до системи, розробка діаграми варіантів використання системи, опис прийнятих проектних рішень при розробці системи, опис архітектури розробленої системи, логічне та фізичне моделювання даних, створення бази даних на платформі СУБД Postgress, джерела даних та методи їх отримання, аналіз та вибір моделей машинного навчання, проектування взаємодії компонентів системи, реалізація ML сервісу на Python, Реалізація Java бекенду, реалізація фронтенду, результати дослідження та ефективність системи, порівняння моделей та їх статистичні результати, перелік та характеристика досліджуваних моделей, методика оцінювання якості моделей, кількісні результати, візуальний аналіз прогнозів, результати експериментів з різними наборами даних, технічні індикатори vs повні дані, вплив sentiment-аналізу, вплив ончейн-метрик, аналіз ефективності системи прогнозування, стресс-тести та навантаження.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни	Примітка
1.	Отримання завдання на кваліфікаційну роботу	17.10.2025	виконано
2.	Аналіз предметної області	26.10.2025	виконано
3.	Постановка задачі на розробку системи	5.11.2025	виконано
4.	Проектування та розробка компонентів системи	25.11.2025	виконано
5.	Оформлення пояснювальної записки	17.11.2025	виконано
6.	Подання закінченої роботи науковому керівникові	16.12.2025	виконано
7.	Підготовка презентації	18.12.2025	виконано
8.	Перевірка оригінальності тексту	22.12.2025	виконано
9.	Подання роботи на рецензування	23.12.2025	виконано
10.	Попередній захист	23.12.2025	виконано
11.	Подання роботи до екзаменаційної комісії	24.12.2025	виконано

Дата видачі завдання 17 жовтня 2025 р.

Здобувач

(підпис)

Мацюцький В.Р.

Керівник роботи (проекту)

(підпис)

доцент каф. СТ Урняєва І.А.

(посада, прізвище, ініціали)

РЕФЕРАТ

Робота містить: 73 сторінок, 24 рисунків, 2 таблиці, 16 джерел інформації.

ІНФОРМАЦІЙНІ СИСТЕМИ, КРИПТОВАЛЮТИ, МОДЕЛЮВАННЯ,
НЕЙРОМЕРЕЖІ, НОРМАЛІЗАЦІЯ, ПРОГНОЗУВАННЯ, РИНОК, ТОЧНІСТЬ,
ТРЕНУВАННЯ, ВОЛАТИЛЬНІСТЬ, DFD, PYTHON

Об'єктом дослідження є динаміка ринкових цін на криптовалюти.

Предметом дослідження є процеси збору, обробки та прогнозування часових рядів цін на криптовалюти.

Метою дослідження є розробка технології прогнозування динаміки цін криптовалют із застосуванням сучасних інтелектуальних методів аналізу даних та моделей машинного навчання.

Методи дослідження включають аналіз часових рядів, нормалізацію даних, побудову технічних індикаторів, застосування моделей машинного та глибинного навчання. Реалізація базується на Python із використанням сучасних бібліотек для аналізу даних та нейронних мереж.

У результаті визначено найбільш ефективні підходи до підготовки криптовалютних даних та обрано моделі, що забезпечують підвищену точність прогнозування. Новизна роботи полягає у комплексному поєднанні методів фільтрації даних, технічних індикаторів і сучасних архітектур глибинного навчання в одному інструменті.

Розроблена система має модульну структуру, підтримує масштабування, містить механізми оновлення моделей і засоби візуалізації результатів. Роботу можна впроваджувати у фінтех-сервісах, аналітичних платформах і дослідницьких системах.

Отримані результати можуть бути використані у фінансовій аналітиці, для побудови торгових стратегій і навчальних цілей. Економічна ефективність полягає у зменшенні ризиків та підвищенні точності прогнозів, а значимість у створенні сучасного інструментарію для аналізу криптовалютних ринків.

ABSTRACT

The work contains: 72 pages, 24 figures, 2 tables, 16 sources.

INFORMATION SYSTEMS, CRYPTOCURRENCIES, MODELING, NEURAL NETWORKS, NORMALIZATION, FORECASTING, CRYPTOCURRENCY MARKET, ACCURACY, MODEL TRAINING, VOLATILITY, DFD, PYTHON

The object of the study is the dynamics of market prices for cryptocurrencies.

The subject of the study is the processes of collecting, processing and forecasting time series of prices for cryptocurrencies.

The purpose of the study is to develop a technology for forecasting the dynamics of cryptocurrency prices using modern intelligent methods of data analysis and machine learning models.

The research methods include time series analysis, data normalization, construction of technical indicators, application of machine and deep learning models. The implementation is based on Python using modern libraries for data analysis and neural networks.

As a result, the most effective approaches to cryptocurrency data preparation were identified, and models that provide improved forecasting accuracy were selected. The novelty of the work lies in the comprehensive combination of data filtering methods, technical indicators, and modern deep learning architectures within a single tool.

The developed system has a modular structure, supports scalability, includes mechanisms for model updating, and provides tools for result visualization. The system can be implemented in fintech services, analytical platforms, and research systems.

The obtained results can be used in financial analytics, for building trading strategies, and for educational purposes. The economic efficiency lies in risk reduction and improved forecast accuracy, while the significance is in creating a modern toolkit for analyzing cryptocurrency markets.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Інформаційні системи у сфері аналізу динаміки фінансових даних та криптовалютних ринків.....	10
1.2 Огляд існуючих розробок та підходів.....	13
1.3 Аналіз існуючих підходів.....	15
1.4 Постановка задачі дослідження.....	17
2 РОЗРОБКА ВИМОГ ДО РОЗРОБЛЮВАЛЬНОЇ СИСТЕМИ	19
2.1 Аналіз функціональних вимог до системи	19
2.2 Розробка моделі потоків даних системи.....	26
2.3 Розробка системних вимог до системи.....	29
2.4.1 Розробка діаграми варіантів використання системи	30
3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ ПРИ РОЗРОБЦІ СИСТЕМИ.....	33
3.1 Опис архітектури розробленої системи.....	33
3.2 Логічне та фізичне моделювання даних	35
3.3 Створення бази даних на платформі СУБД Postgress	38
3.4. Джерела даних та методи їх отримання	39
3.5. Аналіз та вибір моделей машинного навчання.....	41
3.7. Проектування взаємодії компонентів системи	43
3.4. Реалізація ML сервісу на Python	46
3.5. Реалізація Java бекенду	49
3.6. Реалізація фронтенду.....	52
4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТА ЕФЕКТИВНІСТЬ СИСТЕМИ	55
4.1. Порівняння моделей та їх статистичні результати.....	55
4.1.1. Перелік та характеристика досліджуваних моделей.....	55
4.1.2. Методика оцінювання якості моделей.....	57
4.1.3. Кількісні результати	58
4.1.4. Візуальний аналіз прогнозів	61
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	72
Додаток А	Помилка! Закладку не визначено.
Додаток Б	Помилка! Закладку не визначено.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

ІС – інформаційна система

СУБД – система управління базами даних

Волатильність – міра мінливості ціни активу за певний період

ARIMA – Autoregressive Integrated Moving Average

CNN – Convolutional Neural Networks

DFD – діаграма потоків даних

GARCH – Generalized Autoregressive Conditional Heteroskedasticity

GRU – Gated Recurrent Unit

VAR – Vector Autoregression)

SVR – Методи опорних векторів

LSTM – Long Short-Term Memory

ВСТУП

Швидкий розвиток цифрових фінансових технологій та виникнення децентралізованих валют спричинили формування ринку криптовалют, який характеризується високою динамічністю, мінливістю та значними обсягами транзакцій. У сучасних реаліях криптовалюти стали повноцінним фінансовим інструментом, що активно застосовується інвесторами, компаніями та платіжними системами. Передбачення їхніх вартостей є ключовим завданням фінансової аналітики, адже воно дає змогу знижувати ризики, вдосконалювати торговельні стратегії та збільшувати результативність ухвалення рішень.

Світові тренди у прогнозуванні цін криптовалют полягають у зміщенні від традиційних статистичних моделей до глибоких нейронних мереж, здатних оперувати з нелінійними та нестабільними даними ринку. Головний наголос робиться на комбінуванні часових рядів із технічними показниками, даними мережевої активності, обсягами торгів, метриками блокчейну та суспільними сигналами.

Актуальність роботи зумовлена високою волатильністю ринку та нестабільністю закономірностей, що ускладнює математичне моделювання. Існуючі дослідження показують, що точність прогнозів залежить від якості даних та вибору моделі, проте аналітикам і трейдерам бракує інтегрованих інструментів, що поєднують підготовку даних, тестування моделей і оцінку результатів у єдиній системі.

Метою роботи є розробка методології та програмної системи для комплексного аналізу підходів до прогнозування динаміки криптовалют, визначення найефективніших моделей, дослідження впливу структури та якості даних на точність прогнозів та реалізація інтелектуального модуля, здатного видавати результати разом із відповідною статистикою. Розроблений інструментарій може застосовуватися у фінансових аналітичних платформах, торгових терміналах, освітніх модулях та корпоративних рішеннях з управління ризиками.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи у сфері аналізу динаміки фінансових даних та криптовалютних ринків

Інформаційні системи сьогодні є фундаментальною основою для функціонування будь-яких сучасних організацій, платформ і сервісів, особливо тих, що працюють у цифровому середовищі та оперують великими масивами даних [1]. Їхня структура поєднує бази даних, програмно-аналітичні модулі, серверні потужності, мережеву інфраструктуру та сервіси візуалізації, створюючи цілісну екосистему, здатну забезпечувати збір, обробку, аналіз і передачу інформації. Основою таких систем є бази даних, у яких зберігаються як сирі дані, так і проміжні результати обчислень, зокрема агреговані часові ряди чи аналітичні показники. Вони виконують роль центрального сховища, що забезпечує доступність і цілісність інформації, а також дає змогу інтегрувати зовнішні джерела.

Над базовим рівнем збереження функціонують сервери, які забезпечують виконання аналітичних запитів, роботу моделей передбачення й логіку взаємодії між компонентами. Серверна частина в таких системах часто працює під значним навантаженням, оскільки дані постійно надходять, а обчислення мають виконуватися у режимі наближеному до реального часу [2]. Це надзвичайно суттєво для ринків фінансів, де будь-яка затримка може спричинити втрату цінності даних або неточності передбачення. Розподіл навантаження між серверами дає змогу рівномірно розподіляти ресурси так, аби система лишалась стійкою, навіть коли кількість запитів або обсяг даних стрімко збільшується.

Ринок криптовалют є типовим прикладом середовища, де швидкість, постійний потік інформації та непередбачуваність відіграють вирішальну роль. На відміну від традиційних фондових ринків, де зміни можуть бути більш плавними, криптовалютний сегмент характеризується високою волатильністю та значною децентралізованістю. Це створює додаткові вимоги до інформаційних систем: вони повинні мати стійку інфраструктуру, здатну обробляти дані з численних бірж,

враховувати часові лаги, коригувати технічні розбіжності та забезпечувати максимально точний, актуальний і синхронізований набір даних [3].

З урахуванням специфіки криптовалютних ринків, розподілені інформаційні системи стають ключовою технологічною основою для забезпечення безперервної роботи аналітичних сервісів. Розподілена архітектура дозволяє розміщувати обчислювальні процеси на різних серверах та навіть у різних географічних регіонах. Це підвищує доступність платформ, мінімізує ризик відмови та дає змогу масштабувати обчислювальні ресурси відповідно до потреб аналізу ринку.

Винятковою аспектом подібних систем є можливість оперувати з численними джерелами даних, об'єднуючи їх у єдиний потік, який потребує очищення, сукупності та стандартизації. Для передбачення криптовалют це критично необхідно, адже саме якість вхідних даних зумовлює точність наступних моделей. Система повинна самостійно узгоджувати часові проміжки, вирівнювати різні формати подання OHLCV-рядків, усувати шум та виключати відхилення. На цьому етапі створюється єдиний упорядкований набір даних, придатний для подальшого комп'ютерного дослідження.

Також значущим елементом розгалужених систем є можливість з'єднання блоків машинного навчання без зниження ефективності. Оскільки моделі передбачення навчаються на об'ємних архівних масивах фінансової інформації, їхня обчислювальна складність може бути досить високою. Розподілені системи дають змогу паралельно стартувати етапи навчання, перевірки моделей та актуалізації даних.

У контексті електронної комерції та фінансових платформ інформаційні системи поділяються за своїм призначенням на операційні, управлінські, аналітичні та інтегровані [4]. Криптовалютні біржі та сервіси моніторингу ринку належать до класу спеціалізованих фінансових інформаційних систем, які поєднують елементи електронної комерції з можливостями глибокого аналітичного аналізу. Вони включають інструменти взаємодії з користувачами, підсистеми візуалізації графіків, модулі інтерпретації ринкових сигналів, рекомендаційні блоки та алгоритмічні модулі автоматичного прогнозування.

У таких системах особливу роль відіграють модулі, які відповідають за аналітику і прогнозування. Вони формують фундамент для прийняття рішень як роздрібними трейдерами, так і великими інвесторами. Сюди входять алгоритми класифікації тенденцій, підсистеми оцінювання ризиків, індикаторні блоки, що розраховують технічні показники, та модулі прогнозування, які використовують машинне навчання для визначення майбутніх змін ринкової динаміки [5].

У роботі увага приділяється процесами підготовки даних, визначення оптимальних моделей прогнозування, перевірки їх точності та побудови інформаційної системи, яка дозволяє автоматизувати формування прогнозів і відображення їх статистичних характеристик.

Задача прогнозування криптовалют посідає центральне місце у структурі інтелектуальних фінансових систем, оскільки саме від її точності залежить можливість приймати ефективні рішення на ринку з високим рівнем ризику. Прогнозування в такій системі не може розглядатися як ізольований етап: воно є результатом складного процесу, який включає збір даних, їх очищення, нормалізацію, обчислення додаткових ознак і формування структурованих наборів для машинного аналізу.

Підготовка даних у криптовалютній сфері – це обов'язковий етап, оскільки ринок характеризується нестабільністю, непостійністю і наявністю значної кількості шуму. Застосування методів масштабування, таких як MinMax або StandardScaler, дозволяє привести дані до стабільного діапазону значень, покращуючи роботу моделей та запобігаючи числовим спотворенням. Додаткові технічні індикатори SMA, EMA, RSI, MACD та інші дають змогу глибше описати стан ринку, що позитивно впливає на точність прогнозування [6].

На основі підготовлених даних моделі машинного навчання аналізують динаміку вартості та формують прогноз. У роботі важливо зіставити декілька моделей, з'ясувати, які саме методики (алгоритми) найкраще слугують для передбачення криптовалютних часових послідовностей, а також виміряти їх ефективність на випробувальних даних. Це дає змогу розробити інформаційну систему, яка не тільки видає прогноз, але й пропонує користувачу повну статистику

щодо придатності моделей, зокрема похибки, стійкості та спроможності узагальнювати дані.

1.2 Огляд існуючих розробок та підходів

Дослідження динаміки криптовалют ґрунтується на поєднанні класичних методів аналізу часових рядів та сучасних інтелектуальних технологій, що дозволяють враховувати високу волатильність, нерегулярність та нелінійність поведінки ринку.[4]

Часові ряди являють собою послідовність значень показника, зафіксованих у певні моменти часу. Стосовно криптовалют такими значеннями є ціни, обсяги торгів та інші біржові показники, що трансформуються з окремим часовим періодом. Дослідження часових рядів дає змогу виявляти закономірності зміни даних у часі та застосовувати їх для передбачення майбутніх значень.

Часові ряди криптовалют мають високу нестійкість, значну кількість шуму та стрімкі коливання вартості.

У процесі опрацювання часових рядів використовуються методи приведення до спільного вигляду, усереднення та формування технічних індикаторів, які дозволяють зменшити вплив шуму та окреслити головні тренди. Поєднання традиційних підходів з методами машинного та глибинного навчання забезпечує більш інформативний аналіз часових рядів та поліпшує точність прогнозування руху криптовалют.

На початковому етапі досліджень фінансових часових рядів використовували класичні статистичні моделі. Моделі ARIMA добре працюють для даних зі стабільною сезонністю, проте на ринку криптовалют їх точність знижується через відсутність чітких закономірностей, тому вони найчастіше застосовуються як базовий порівняльний варіант [7]. Моделі GARCH дозволяють враховувати змінну волатильність, що є характерною для криптовалютних цін, ефективно описують раптові стрибки і періоди підвищеної нестабільності, однак не здатні повністю відобразити складну нелінійну структуру ринку [8]. Моделі VAR

використовуються при аналізі взаємозв'язку декількох криптовалют або технічних індикаторів, проте їх адаптивність до хаотичних коливань обмежена. Загалом статистичні моделі досліджуються як еталонні рішення, що дозволяє оцінювати ефективність сучасних алгоритмів машинного навчання.

Увага до методів машинного навчання для прогнозування криптовалют суттєво зріс у 2016–2018 роках [9]. Лінійні та поліноміальні регресійні моделі застосовуються для базових експериментів, проте не враховують складну нелінійність ринку. Моделі на основі дерев рішень, такі як Random Forest та Gradient Boosting, здатні виявляти складні залежності між ознаками, наприклад технічними індикаторами, обсягами торгів чи макроекономічними параметрами. SVR ефективні для виявлення чітких нелінійних залежностей і в багатьох дослідженнях демонструють кращі результати на коротких горизонтах прогнозування, ніж класичні статистичні моделі або прості нейронні мережі. При цьому точність алгоритмів машинного навчання безпосередньо залежить від якості та кількості даних, а також від правильної підготовки ознак, включаючи розрахунок технічних індикаторів, нормалізацію та формування часових рядів.

Найсучасніші дослідження у сфері прогнозування криптовалют зосереджені на застосуванні нейронних мереж і глибокого навчання. Моделі LSTM добре аналізують довгі часові залежності та ефективно працюють з волатильними даними, часто забезпечуючи значно вищу точність порівняно з ARIMA чи SVR. Мережі GRU мають меншу кількість параметрів, що прискорює навчання, і підходять для систем, які працюють у реальному часі. CNN, хоча традиційно використовуються для обробки зображень, можуть трансформувати часові ряди у матриці ознак та виявляти локальні патерни, а сучасні моделі Transformers дозволяють ефективно враховувати довгі залежності без рекурентних структур, що робить їх одними з найефективніших на великих обсягах даних. У практичних дослідженнях 2022–2024 років активно застосовуються гібридні архітектури, які поєднують різні підходи, наприклад CNN-LSTM, Transformer-LSTM або ARIMA-LSTM, що дозволяє поєднувати переваги кількох моделей.

У реальних прикладних системах прогнозування криптовалют часто інтегруються готові алгоритми та аналітичні інструменти. Платформи на кшталт TradingView застосовують власні математичні моделі для оцінки трендів, біржі Binance, Coinbase та Kraken пропонують графіки, технічні індикатори та базові прогнозні модулі, а спеціалізовані сервіси, такі як CoinPredictor, LunarCrush або IntoTheBlock, використовують алгоритми машинного навчання, аналіз соціальних медіа та дані блокчейнів [10]. Зазвичай користувач отримує готовий прогноз, проте не має доступу до вихідних моделей і даних, що стимулює науковців створювати відкриті системи та порівнювати ефективність різних підходів на стандартизованих наборах даних.

Розгляд наявних підходів демонструє, що кожна сукупність методів володіє власними перевагами та недоліками. Статистичні моделі зрозумілі та легкі в застосуванні, але знижують відносну точність у разі значної мінливості. Методи машинного навчання надають гнучкість та врахування заплутаних атрибутів, проте дуже залежать від якості даних і коректного вибору ознак. Нейронні мережі та глибоке навчання дозволяють досягти найвищої точності, однак вимагають значних обсягів даних та процесорних потужностей. Комбіновані (гібридні) моделі зводять разом переваги різнорідних методів, але є важкими у впровадженні. У цьому світлі, дослідження, яке зіставляє результативність різних моделей, розглядає структуру даних та створює систему, що самостійно формує прогнози і показує їх статистику, займає значуще місце у поточних наукових та практичних роботах і відповідає актуальним напрямам у сфері інтелектуального прогнозування фінансових ринків.

1.3 Аналіз існуючих підходів

У ході дослідження планується розроблення та вдосконалення комплексної інформаційної технології прогнозування зміни цін криптовалют, яка поєднуватиме методи збору та обробки даних із сучасними моделями машинного та глибинного навчання. Основою для подальшої розробки слугуватимуть існуючі наукові

підходи, що довели ефективність у близьких задачах, а також напрацювання, які описують специфіку побудови моделей для високоволатильних ринків. Система, що створюється, має не лише відтворювати вже відомі методи, але й адаптувати їх до умов криптовалютного ринку, забезпечуючи підвищену точність, гнучкість та можливість подальшого масштабування.

Для досягнення поставленої мети необхідно розробити повноцінний технологічний цикл, який включатиме механізми автоматизованого збору даних, методи їх глибокого очищення, агрегування та формування узгодженого набору ознак, що підходять для різних типів моделей. Окрім цього, потрібно визначити оптимальні архітектури моделей прогнозування, налаштувати систему порівняння їхніх результатів та забезпечити можливість вибору найточнішої моделі для поточного періоду. Важливим елементом розробки є також створення інструментарію, який дозволить здійснювати регулярне оновлення даних, повторне тренування моделей та формування статистики щодо їх точності на різних часових інтервалах.

Підбір, очищення та фільтрація даних ґрунтуватимуться на матеріалах, де детально розкрито специфіку структурування криптовалютних часових рядів, методи усунення шумів, вибору часових лагів, формування технічних індикаторів та інтеграції зовнішніх даних [11]. Саме ці підходи формують основу блоку підготовки даних у розроблюваній системі. Натомість вибір моделей прогнозування та їх порівняння базуватиметься на результатах дослідження Omole та Enke (2024), у якому проведено ґрунтовний аналіз ефективності різних архітектур – від LSTM та GRU до CNN-LSTM, а також оцінено їхню здатність передбачати напрямок руху ринку [12]. Узагальнення цих наукових джерел дозволяє сформулювати концептуальну основу майбутньої системи та визначити ключові компоненти, які слід реалізувати або вдосконалити.

1.4 Постановка задачі дослідження

Задача полягає у розробці технології прогнозування динаміки цін криптовалют із застосуванням сучасних інтелектуальних методів аналізу даних та моделей машинного навчання. Вихідною інформацією для побудови системи є часові ряди криптовалют, що включають відкриття, закриття, максимальні та мінімальні значення, обсяги торгів, а також додаткові технічні індикатори, що обчислюються на їх основі, та зовнішні фактори, які можуть впливати на ринок (наприклад, макроекономічні дані, новини, активність користувачів у соцмережах, дані блокчейнів).

Передбачається створення комплексної технології, яка дозволяє обробляти вхідні дані, підготовлювати їх для подальшого аналізу, вибирати та тестувати моделі прогнозування, а також формувати вихідний результат у вигляді прогнозованих значень цін і статистичних показників ефективності моделей. Технологія включає кілька ключових етапів.

Перший етап передбачає збір та підготовку даних. Він включає очищення та нормалізацію часових рядів, обчислення технічних індикаторів, формування наборів ознак для моделей машинного навчання, а також розподіл даних на навчальну, тестову та валідаційну вибірки. На цьому етапі важливо забезпечити коректне відображення динаміки ринку та врахувати особливості волатильності криптовалют, нерегулярність даних і можливу наявність пропусків.

Другий етап передбачає вибір моделей прогнозування і визначення їх параметрів. Для цього проводиться порівняльний аналіз різних методів, включаючи класичні статистичні підходи, моделі машинного навчання та нейронні мережі глибинного навчання. Кожна обрана модель тестується на підготовлених даних із подальшою оцінкою точності прогнозів, стабільності результатів і здатності моделі адаптуватися до змін ринку.

Третій етап передбачає розробку програмного інструментарію, який автоматизує процес прогнозування. Система повинна здійснювати інтеграцію модулів для обробки даних, запуску моделей прогнозування та візуалізації

результатів у вигляді графіків і статистичних показників. Особлива увага приділяється забезпеченню можливості вибору різних моделей, порівнянню їхніх результатів та аналізу ефективності.

В результаті технологія забезпечує формування прогнозу цін для обраної моделі та надання повної статистики для оцінки точності та надійності прогнозів. В результаті реалізована система дозволяє не лише отримувати прогнози, але й здійснювати порівняльний аналіз моделей, оцінювати вплив структури та якості даних на результати прогнозування і підвищувати ефективність прийняття рішень у сфері криптовалютної торгівлі.

2 РОЗРОБКА ВИМОГ ДО РОЗРОБЛЮВАЛЬНОЇ СИСТЕМИ

2.1 Аналіз функціональних вимог до системи

На основі аналізу предметної області та постановки задачі проведено функціональне моделювання у стандарті IDEF0 розроблюваної системи. Контекстна діаграма наведена на рисунку 2.1.

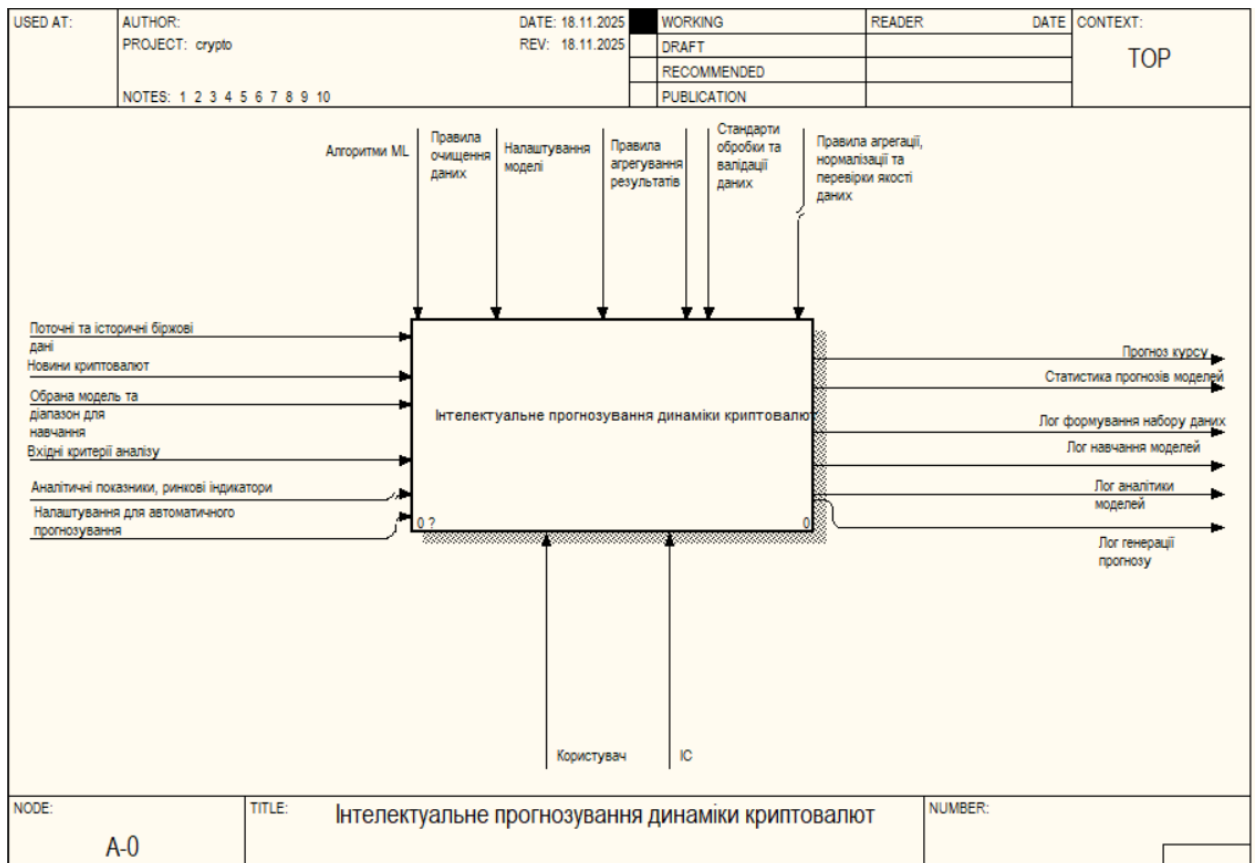


Рисунок 2.1 – Контекстна діаграма

Головним бізнес-процесом розроблюваної системи є «Інтелектуальне прогнозування динаміки криптовалюту». Цей компонент є головним процесом системи та відповідає за керування усіма операціями прогнозування та аналітики ціни криптовалюти. Він забезпечує обробку даних із зовнішніх джерел, підготовку та нормалізацію часових рядів, навчання моделей машинного та глибинного навчання, генерацію прогнозів для обраних криптовалют, а також формування

аналітичних звітів для оцінки точності прогнозів і підтримки прийняття рішень користувачем.

Вхідними даними для головного процесу є «Поточні та історичні біржові дані», «Новини криптовалют», «Аналітичні показники, ринкові індикатори», «Обрана модель та діапазон для навчання», «Вхідні критерії аналізу», «Вхідні критерії аналізу», «Налаштування для автоматичного прогнозування».

Вихідними даними є «Прогноз курсу», «Статистика прогнозів моделей», «Лог формування набору даних», «Лог навчання моделей», «Лог аналітики моделей», «Лог генерації прогнозу».

Ресурси механізму включають в себе ІС та користувач системи.

В якості управління використовуються алгоритми ML, Правила очищення даних, налаштування моделі, правила агрегування результатів, стандарти обробки та валідації даних, правила агрегації, нормалізації та перевірки якості даних.

У результаті декомпозиції головного бізнес-процесу виділено чотири основні підпроцеси: «Збір даних із зовнішніх джерел», «Процесинг та збереження даних», «Прогнозування та аналітична модель», «Аналіз історії прогнозів». Діаграма декомпозиції контекстної діаграми зображена на рис. 2.2.

Процес «Збір даних із зовнішніх джерел» відповідає за отримання необхідних даних для прогнозування. Він включає завантаження історичних котирувань, технічних індикаторів та додаткових даних (наприклад, обсягів торгів, соціальних сигналів). На цьому етапі відбувається первинна валідація та перевірка цілісності даних, а також підготовка їх для подальшої агрегації та формування єдиного набору даних для навчання моделей.

Процес «Процесинг та збереження даних» забезпечує обробку та структурування отриманих даних. Він включає нормалізацію часових рядів, очищення від шумів, формування ознак для моделей, а також збереження агрегованого датасету. Процес повертає готовий до навчання та прогнозування набір даних для подальшої роботи.

Процес «Прогнозування та аналітична модель» реалізує навчання моделей та генерацію прогнозів. Він включає вибір моделі користувачем або автоматичний

підбір вже навчених моделей, налаштування параметрів навчання, тренування моделей на агрегованих даних, а також генерацію прогнозів для обраних криптовалют. Результатом процесу є готова інформація для аналізу та оцінки точності моделей.

Процес «Аналіз історії прогнозів» забезпечує створення статистики, що включає оцінку ефективності моделей. Процес включає обчислення точності прогнозів, порівняння моделей, а також формування статистичних звітів.

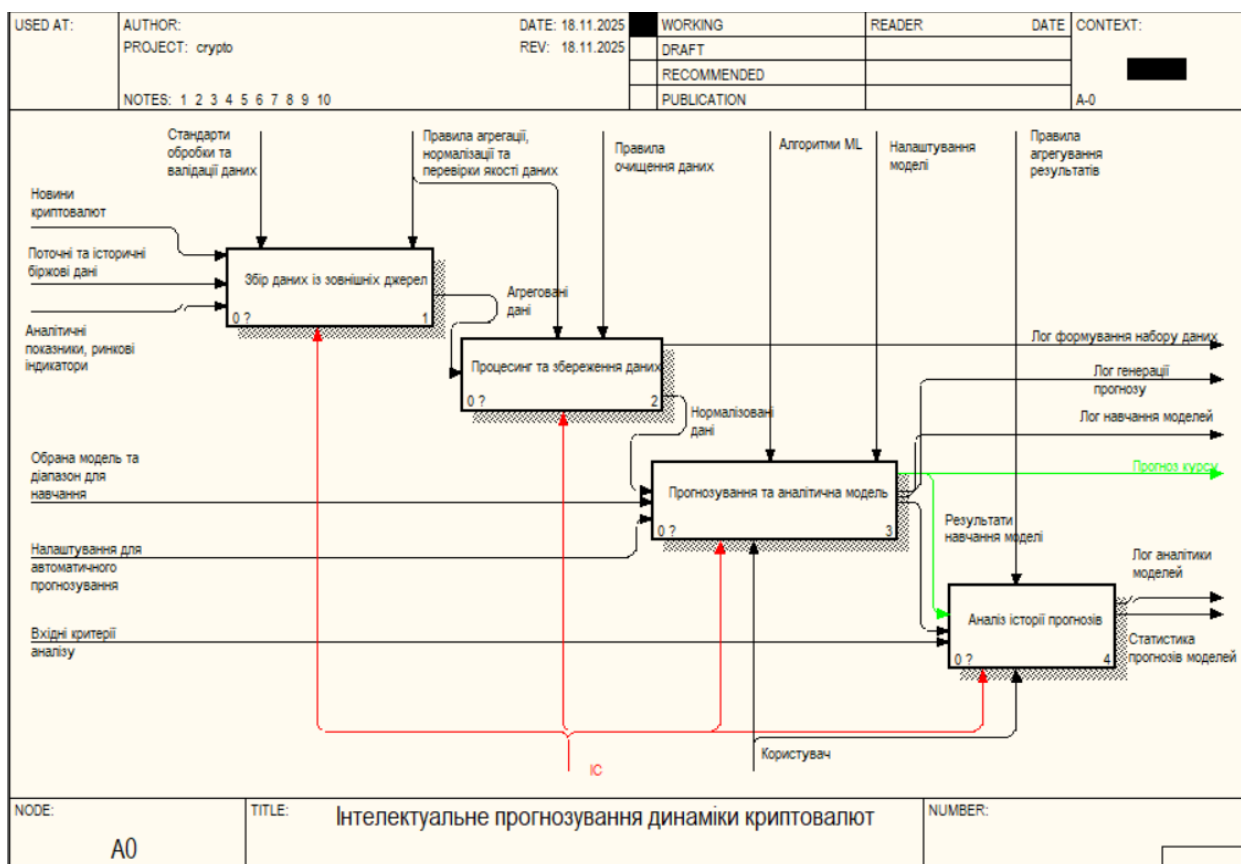


Рисунок 2.2 – Діаграма декомпозиції контекстної діаграми

У результаті декомпозиції процесу «Збір даних із зовнішніх джерел» отримано 2 бізнес функції: «Первинна валідація даних», «Агрегація та формування набору даних». Діаграма декомпозиції процесу «Збір даних із зовнішніх джерел» наведена на рис. 2.3.

У функції «Первинна валідація даних» система приймає дані з різних джерел, включаючи біржі, API та відкриті фінансові бази. Дані перевіряються на цілісність,

коректність форматів, наявність пропусків або аномалій. Після проходження перевірки, дані передаються до етапу «Агрегація та формування набору даних».

У функції «Агрегація та формування набору даних» система об'єднує відфільтровані та перевірені дані у єдиний узгоджений набір, де кожен запис прив'язаний до конкретної дати та часу, типу активу та моделі. Цей набір даних стає основою для подальшого навчання моделей і генерації прогнозів.

У результаті декомпозиції процесу «Процесинг та збереження даних» отримано три бізнес функції: «Перевірка якості даних», «Нормалізація та трансформація» та «Формування фінального датасету». Діаграма декомпозиції процесу «Процесинг та збереження даних» зображена на рисунку 2.4.

У функції «Перевірка якості даних» система оцінює отримані агреговані дані на наявність пропусків, аномалій, дублювань та некоректних значень. Дані, що не відповідають вимогам, позначаються для подальшого очищення або видаляються.

У функції «Нормалізація та трансформація» здійснюється приведення даних до єдиного формату, масштабування числових значень, створення технічних індикаторів та інших ознак, необхідних для моделей прогнозування. Додатково виконується синхронізація часових рядів та інтеграція зовнішніх даних.

У функції «Формування фінального датасету» система об'єднує перевірені та трансформовані дані в єдиний узгоджений набір, прив'язаний до часу, типу активу та інших необхідних параметрів. Цей фінальний датасет стає основою для навчання моделей і подальшого прогнозування динаміки криптовалют.

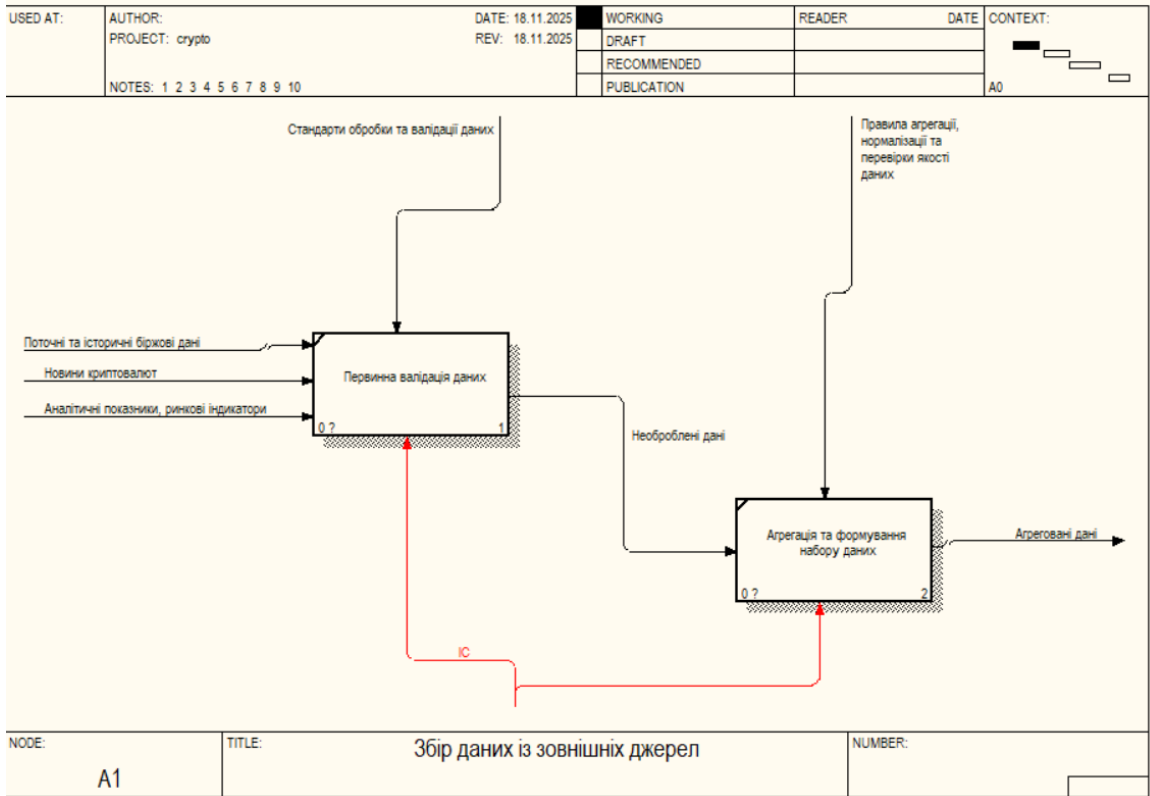


Рисунок 2.3 – Діаграма декомпозиції процесу «Збір даних із зовнішніх джерел»

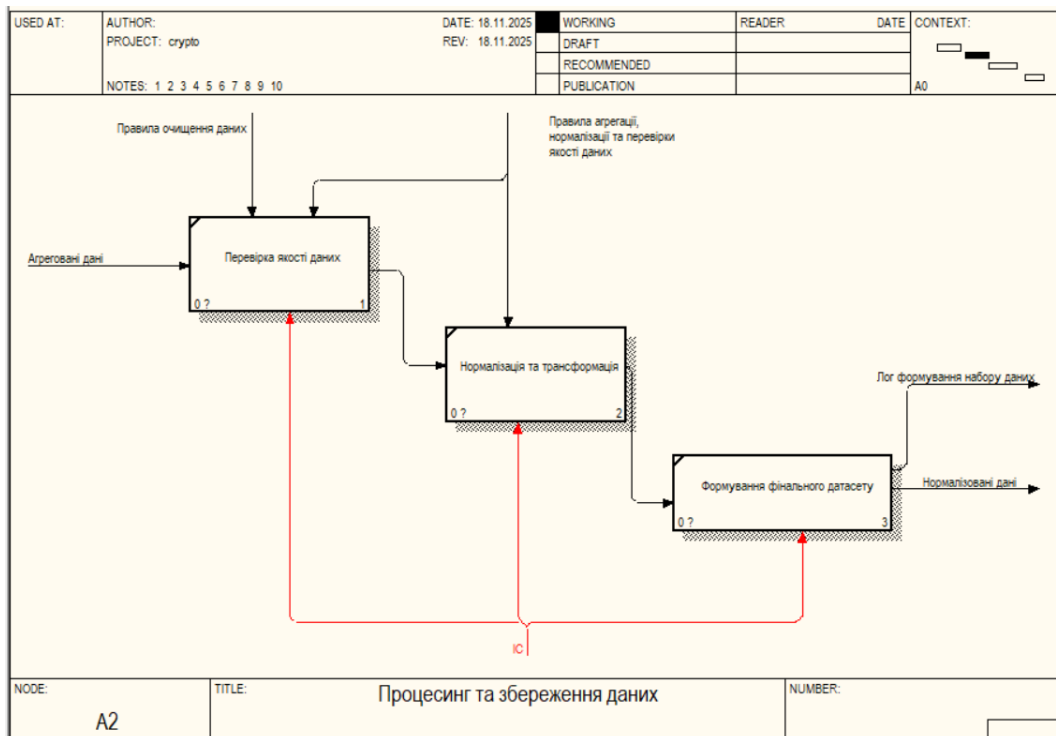


Рисунок 2.4 – Діаграма декомпозиції процесу «Процесинг та збереження даних»

У результаті декомпозиції процесу «Прогнозування та аналітична модель» отримано дві бізнес функції «Навчання моделі» та «Генерація прогнозу». Діаграма

декомпозиції процесу «Прогнозування та аналітична модель» наведено на рисунку 2.5.

У функції «Навчання моделі» система приймає підготовлений фінальний датасет. Користувач може обрати нову модель для навчання з необхідними параметрами. Після налаштування параметрів алгоритму виконується тренування моделі на історичних даних. Результатом цього процесу є навчені моделі, готові до генерації прогнозів.

У функції «Генерація прогнозу» система використовує навчені моделі для створення прогнозів динаміки криптовалют. Прогнози можуть генеруватися за запитом користувача з обраними параметрами або автоматично у задані часові інтервали за визначеними моделями. Результати включають прогнозовані значення, що далі передається на аналіз та формування аналітичної інформації.

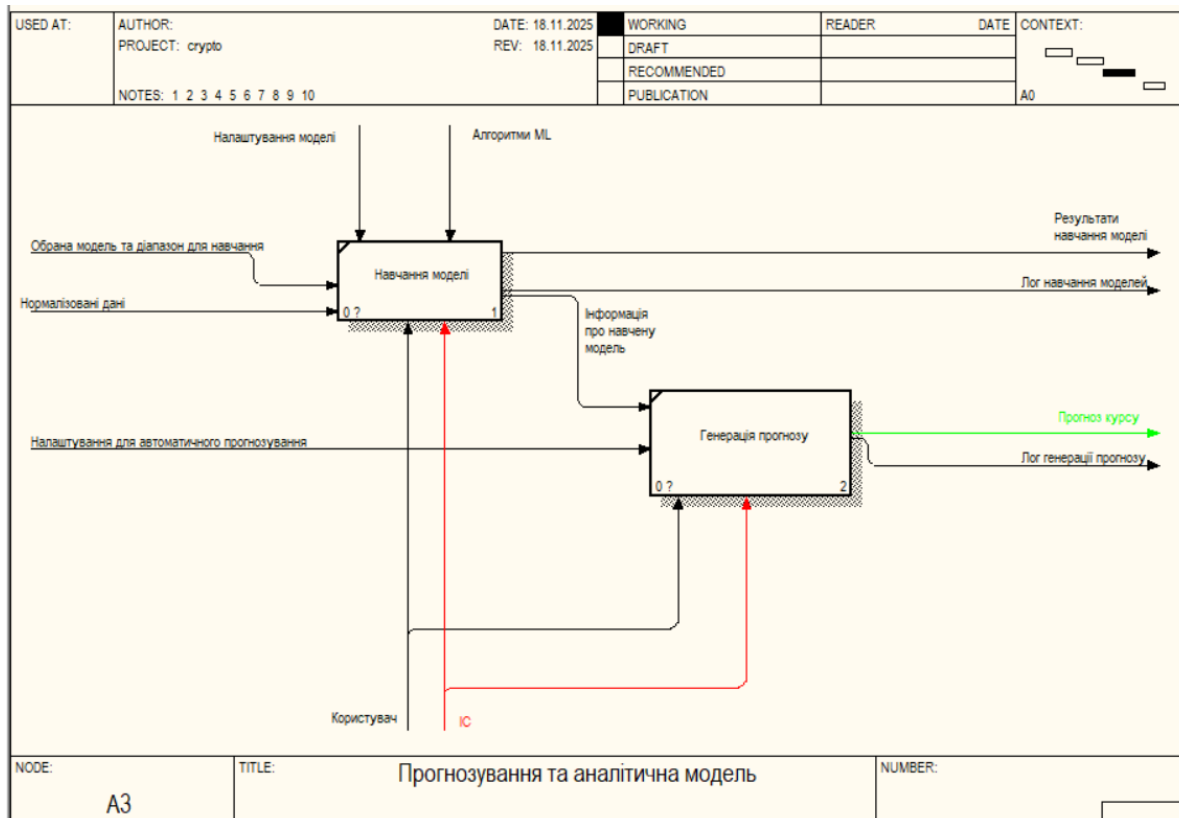


Рисунок 2.5 – Діаграма декомпозиції процесу «Прогнозування та аналітична модель»

У результаті декомпозиції процесу «Аналіз історії прогнозів» отримано бізнес функції: «Обчислення точності прогнозів», «Побудова статистики та аналітики». Діаграма декомпозиції процесу «Аналіз історії прогнозів» зображена на рисунку 2.6.

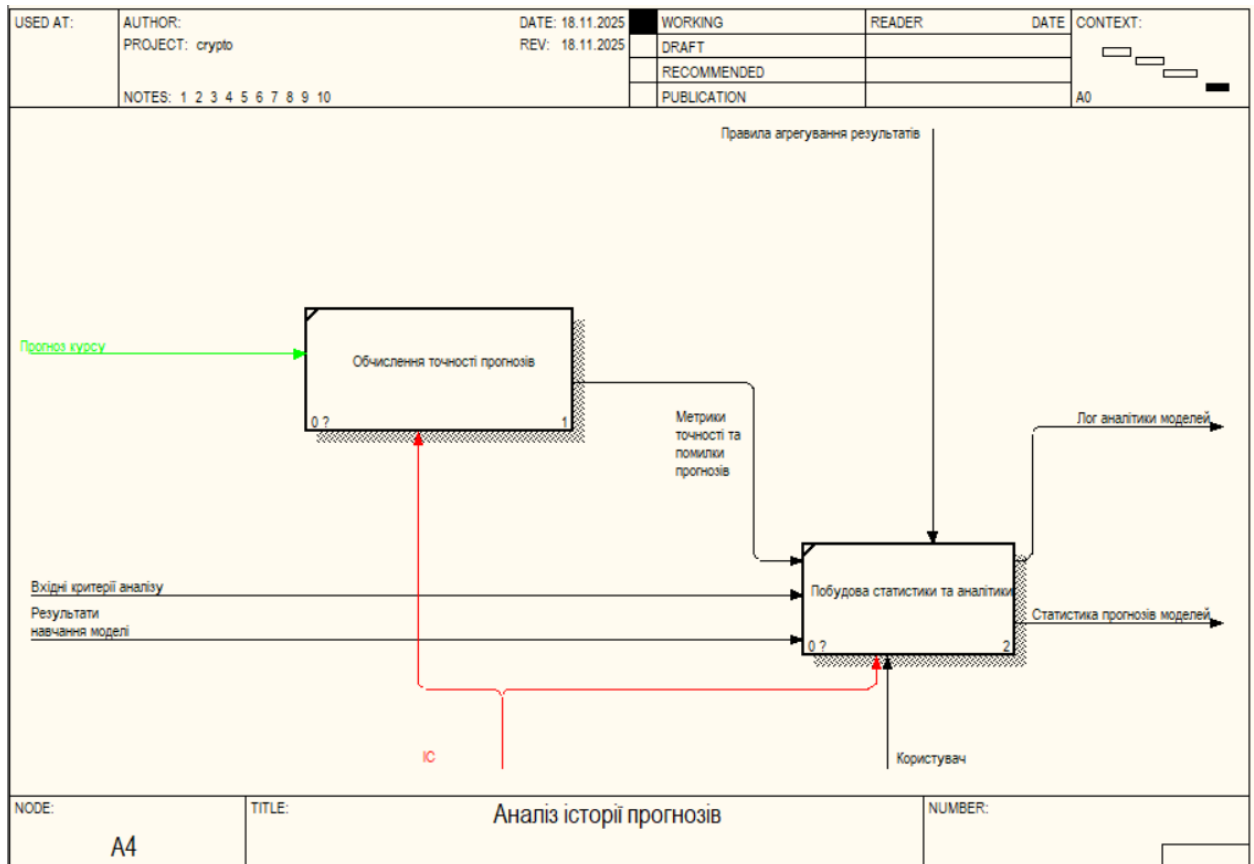


Рисунок 2.6 – Діаграма декомпозиції процесу «Аналіз історії прогнозів»

У функції «Обчислення точності прогнозів» система порівнює отримані прогнозні значення з фактичними даними ринку криптовалют. На основі заданих критеріїв оцінки точності, таких як середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE) або процентна точність, визначаються показники ефективності кожної моделі. Результатом цього процесу є числові оцінки точності прогнозів, які слугують для подальшого аналізу.

У функції «Побудова статистики та аналітики» система формує аналітичні дані на основі обчислених показників точності. Це включає визначення найефективніших моделей для конкретних криптоактивів, аналіз історії прогнозів, оцінку стабільності моделей та візуалізацію результатів для зручного сприйняття

користувачем. Отримані аналітичні матеріали дозволяють приймати обґрунтовані рішення щодо вибору моделі та налаштувань прогнозування для майбутніх періодів.

Діаграма дерева вузлів наведена на рисунку 2.7. На діаграмі наведено процеси системи у ієрархічному вигляді, де системою є «Інтелектуальне прогнозування динаміки криптовалют».

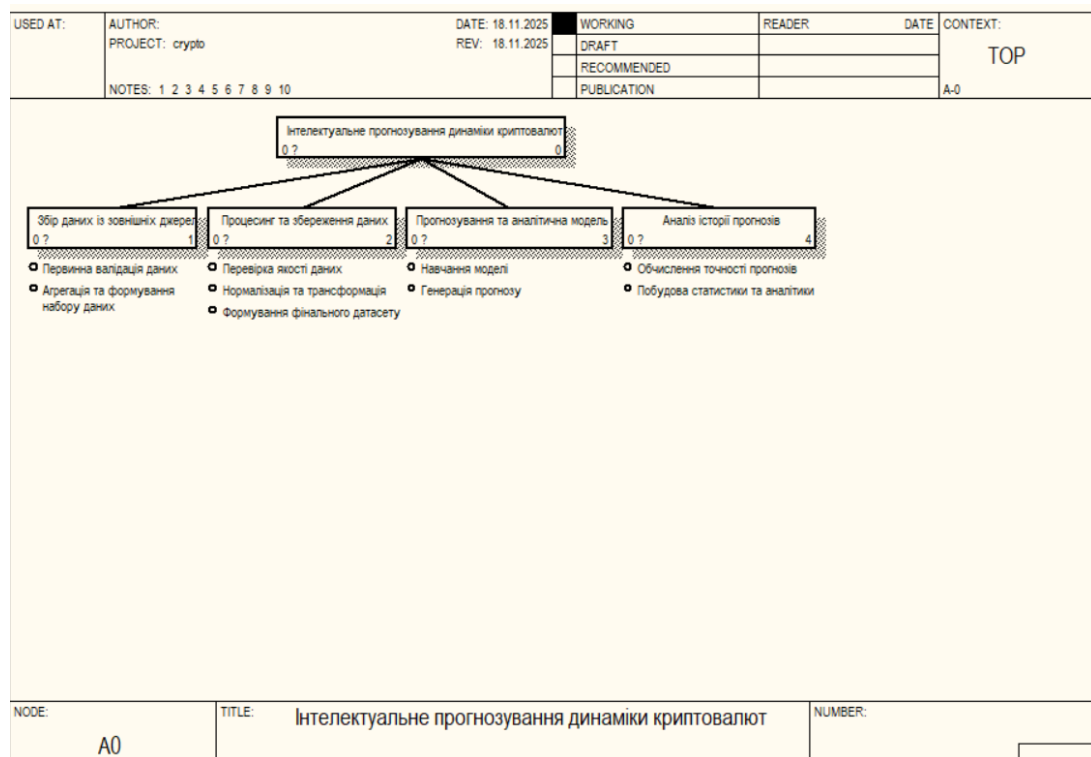


Рисунок 2.7 – Діаграма дерева вузлів

2.2 Розробка моделі потоків даних системи

Для відображення загальної логіки функціонування інформаційної системи було побудовано DFD-діаграму контекстного рівня. Діаграма наведена на рис. 2.8. Вона відображає систему як узагальнений процес та показує її взаємодію із зовнішніми сутностями.

Головним елементом діаграми є процес «Інтелектуальне прогнозування динаміки криптовалют», який реалізує повний цикл обробки даних, навчання

моделей машинного навчання та формування прогнозів. Зовнішніми сутностями виступають Binance API, Telegram API та Користувач.

Від сутності Binance API до центрального процесу надходить потік даних «Історичні біржові дані», який містить інформацію про котирування криптовалют у форматі OHLCV та інші торгові параметри. Ці дані використовуються для аналізу ринкової динаміки та формування навчальних вибірок.

Від сутності Telegram API надходить потік «Новини та повідомлення», що включає текстові дані з обраних каналів. Цей інформаційний потік використовується для проведення аналізу тональності та формування додаткових ознак, які враховують інформаційний вплив соціальних мереж на ринок.

Від Користувача до системи передаються потоки «Вибір параметрів аналізу», «Запит на навчання моделі» та «Запит на прогнозування», які визначають актив, часовий інтервал, модель машинного навчання та часовий діапазон даних. У відповідь система передає користувачеві потоки «Прогноз курсу», «Статистика прогнозів», «Лог навчання моделей» та «Результати аналізу», що забезпечує інтерактивну взаємодію з інформаційною системою.

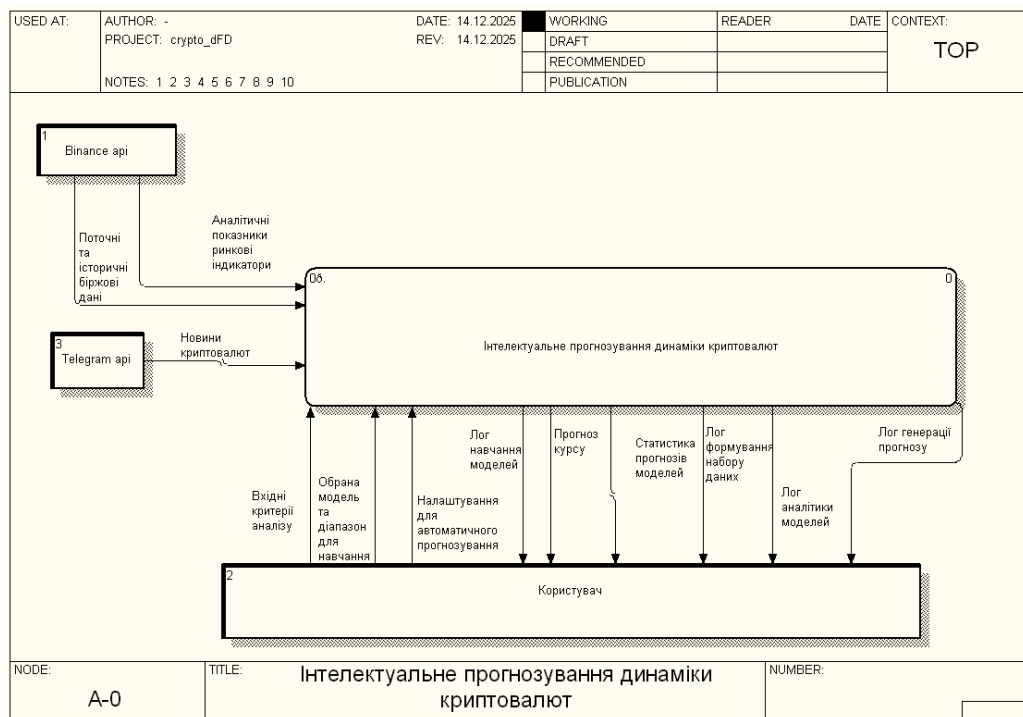


Рисунок 2.8 – Контекстна діаграма DFD

У результаті проведення декомпозиції головного бізнес-процесу виділено чотири основні бізнес-процеси, які відображено на DFD-діаграмі рівня A0, наведеній на рис. 2.9.

Першим бізнес-процесом є «Збір даних із зовнішніх джерел». До цього процесу надходять потоки «Історичні біржові дані» від Binance API та «Новини та повідомлення» від Telegram API. У межах даного процесу здійснюється отримання, первинна перевірка та агрегація даних, після чого формується потік «Агреговані дані», який передається до наступного процесу.

Другим бізнес-процесом є «Обробка та збереження даних». На вхід цього процесу надходять агреговані дані, які проходять нормалізацію, фільтрацію та синхронізацію за часовими інтервалами. Результатом роботи процесу є «Нормалізовані дані», які зберігаються у відповідному сховищі та використовуються для навчання моделей машинного навчання.

Третій бізнес-процес «Прогнозування та аналітика моделей» є ключовим у системі. До нього надходять «Нормалізовані дані», а також потік «Параметри навчання та прогнозування» від користувача. У межах цього процесу здійснюється навчання моделей машинного навчання, генерація прогнозів, а також обчислення метрик якості. Вихідними потоками є «Результати навчання моделей», «Прогнози курсу», «Логи генерації прогнозу» та «Статистика прогнозів».

Четвертим бізнес-процесом є «Аналіз і контроль прогнозів». До нього надходять результати прогнозування, які використовуються для оцінки точності моделей після настання прогнозованого часу. У межах цього процесу формується «Аналіз коректності прогнозів» та «Статистика якості моделей», які передаються користувачеві для подальшого аналізу.

Вихідні потоки з усіх бізнес-процесів спрямовуються до сутності Користувач, що дозволяє переглядати результати прогнозування, історію моделей і аналітичні показники

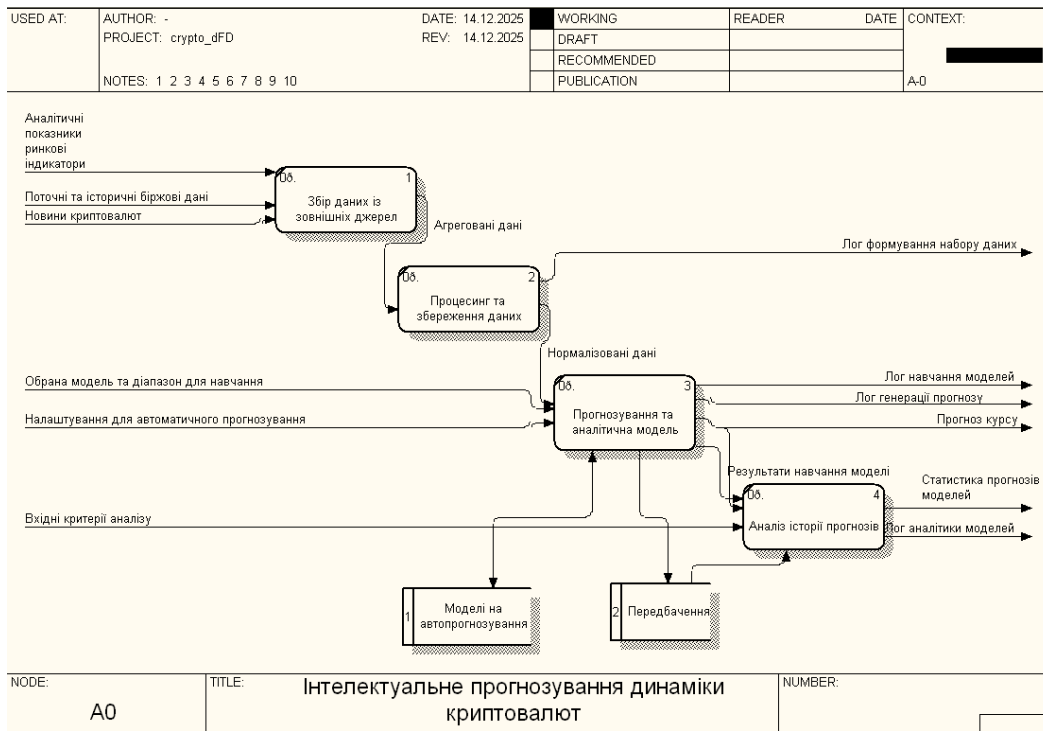


Рисунок 2.9 – Декомпозиція контекстної діаграми DFD

2.3 Розробка системних вимог до системи

Під час формування системних вимог для платформи прогнозування курсу криптовалют було обрано багаторівневу архітектуру, яка забезпечує розподіл логіки між клієнтським інтерфейсом, основним сервером та сервісами машинного навчання.

Клієнтська частина реалізована на React, що дозволяє створювати інтерактивний і адаптивний інтерфейс для сучасних веб-браузерів на різних операційних системах, включаючи мобільні пристрої. Інтерфейс забезпечує можливість перегляду історичних даних, вибору криптовалют для аналізу, запуску процесу навчання моделей, генерації прогнозів та відображення статистики точності моделей.

Основний сервер виконано на Java з використанням Spring Framework, який відповідає за організацію REST API, маршрутизацію запитів між клієнтом та сервісами машинного навчання, управління сесіями користувачів, логування результатів виконання операцій та організацію черги завдань для модулів

машинного навчання. Для спрощення розробки та підтримки коду застосовується Lombok, що автоматизує генерацію допоміжних конструкцій, таких як геттери та сеттери.

Сервер взаємодіє з базою даних PostgreSQL, яка забезпечує надійне збереження даних та їх структуровану організацію.

Сервіси машинного навчання реалізовані на Python із застосуванням FastAPI, що дозволяє швидко обробляти HTTP-запити від Java-сервера та повертати результати обчислень у форматі JSON. На цих сервісах реалізовані модулі для навчання моделей, генерації прогнозів та оцінки їх точності з використанням сучасних бібліотек машинного та глибинного навчання, таких як TensorFlow, PyTorch та scikit-learn, що дозволяє будувати LSTM, GRU, CNN-LSTM та Random Forest для аналізу часових рядів криптовалют.

Система інтегрує готові сервіси, зокрема API бірж криптовалют для отримання ринкових даних у реальному часі та бібліотеки для обробки даних і візуалізації, водночас розробляючи власні модулі збору та агрегації даних, первинної обробки, нормалізації та видалення шумів, а також алгоритми навчання моделей, генерації прогнозів та формування аналітичної статистики.

Контроль версій коду реалізований через Git, що дозволяє інтегрувати нові функції без ризику втрати стабільності системи.

Система спроектована для масштабування та подальшого розширення функціоналу, включаючи інтеграцію додаткових моделей машинного навчання, нових джерел даних та аналітичних інструментів, що забезпечує високу точність прогнозів та ефективну роботу аналітичної платформи у середовищі високоволатильного ринку криптовалют.

2.4.1 Розробка діаграми варіантів використання системи

Діаграма варіантів використання призначена для графічного подання функціональних можливостей інформаційної системи та взаємодії користувачів із

нею. Вона дозволяє визначити основні сценарії використання системи, акторів та їх ролі, а також межі відповідальності системи.

На основі аналізу вимог розроблено діаграму варіантів використання системи прогнозування, яка відображає взаємодію користувачів із функціональними модулями системи(рис. 2.10).

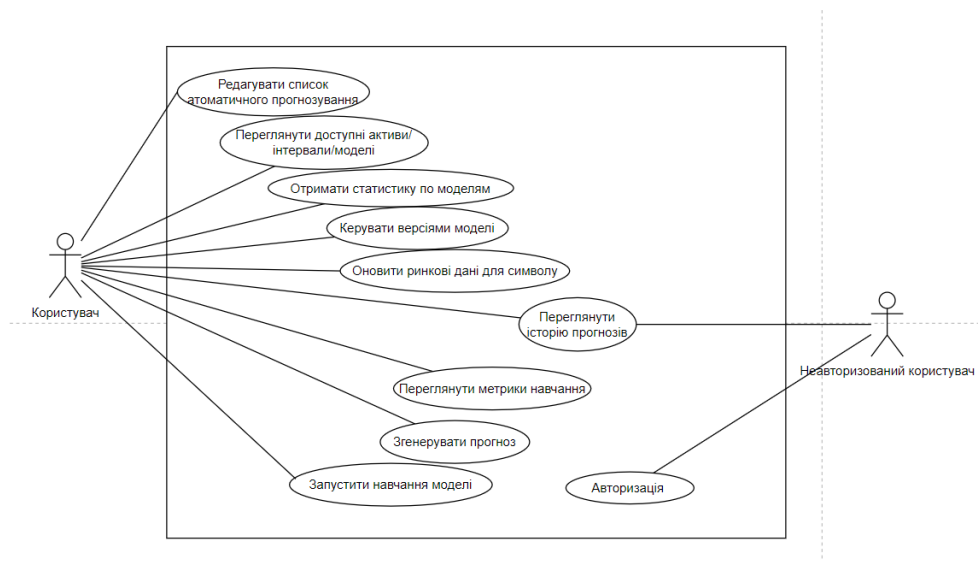


Рисунок 2.10 – Діаграма варіантів використання системи

У межах системи виділено два типи акторів: користувач та неавторизований користувач. Користувачем є авторизований учасник системи, який має повний доступ до її функціональних можливостей і може здійснювати управління моделями прогнозування, аналізувати дані та формувати прогнози. Неавторизований користувач, у свою чергу, не пройшов процедуру автентифікації та має обмежений доступ до функціоналу системи.

Авторизований користувач взаємодіє з системою з метою повноцінної роботи з моделями прогнозування. Він має можливість редагувати список доступних прогнозів, переглядати активні інтерфейси моделей, отримувати статистичну інформацію щодо їхньої роботи, а також керувати версіями моделей шляхом їх оновлення, активації або деактивації. Крім того, користувач може оновлювати дані, які використовуються для аналізу та навчання моделей, переглядати навчальні набори даних і історію раніше згенерованих прогнозів. Важливою функцією є можливість запуску процесу навчання або перенавчання

моделей, а також безпосередня генерація прогнозів на основі обраної моделі та вхідних даних.

Неавторизований користувач має доступ лише до базової функції системи авторизації, яка передбачає введення облікових даних з метою отримання розширеного доступу. Після успішного проходження процедури авторизації користувач переходить до ролі авторизованого користувача та отримує можливість користуватися повним функціоналом системи.

3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ ПРИ РОЗРОБЦІ СИСТЕМИ

3.1 Опис архітектури розробленої системи

Для розробки системи прогнозування курсу криптовалют обрана триланкова архітектура, яка включає презентаційний рівень, логічний рівень та рівень даних, з додатковими сервісами для машинного навчання. Такий підхід забезпечує чітке розділення відповідальностей, спрощує інтеграцію нових компонентів, дозволяє незалежно масштабувати будь-який рівень і підвищує надійність системи.

Презентаційний рівень, або фронтенд, реалізовано на React, що дозволяє створювати динамічний, інтуїтивно зрозумілий веб-інтерфейс. Користувачі можуть переглядати графіки динаміки криптовалют, обирати активи для прогнозування, запускати тренування моделей і отримувати результати прогнозів у режимі реального часу. Взаємодія цього рівня з логічним рівнем здійснюється через REST API, що забезпечує передачу запитів і отримання відповідей у форматі JSON, включно з параметрами моделей, обраними користувачем, та результатами прогнозів.

Логічний рівень реалізований на Java із використанням Spring Boot, що забезпечує основну бізнес-логіку системи, маршрутизацію запитів, управління управління сесіями та чергами завдань для ML-сервісів, що відповідає підходам до побудови розподілених інформаційно-аналітичних систем підтримки прийняття рішень [13, 14]. Використання Lombok дозволяє автоматично генерувати стандартні методи, зменшуючи шаблонний код і підвищуючи підтримуваність. У межах логічного рівня реалізовані алгоритми валідації даних, управління навчанням моделей, обчислення точності прогнозів, а також порівняння результатів різних моделей для вибору оптимальної архітектури на поточний період. Для взаємодії з Python-сервісами ML застосовуються REST-запити, що передають підготовлені набори даних та параметри моделей, а отримані прогнози повертаються назад у логічний рівень для подальшого оброблення та відображення користувачу.

Рівень машинного навчання реалізований як окремі Python-сервіси з FastAPI, які обробляють обчислювально-інтенсивні завдання: тренування нейронних мереж, прогнозування динаміки криптовалют, генерацію серій прогнозів та обчислення метрик точності [13]. Ці сервіси працюють незалежно від Java-сервера, що дозволяє масштабувати обчислення горизонтально, додавати нові моделі або оновлювати існуючі без порушення роботи основної бізнес-логіки. Для навчання моделей застосовуються бібліотеки TensorFlow, PyTorch та scikit-learn, а також алгоритми попередньої обробки даних: нормалізація, фільтрація шумів, формування часових лагів і технічних індикаторів [15].

Рівень даних забезпечує зберігання та організацію всіх вхідних та оброблених даних у PostgreSQL. База підтримує складні SQL-запити та ACID-транзакції, що гарантує цілісність даних[16]. Схема зберігання організована так, щоб агреговані набори даних, історія прогнозів та статистичні показники моделей були доступні для логічного рівня та сервісів машинного навчання без дублювання даних.

Система використовує готові сервіси для збору ринкових даних із відкритих API криптовалютних бірж, а власноруч розроблені модулі відповідають за первинну валідацію, нормалізацію, трансформацію, формування фінального датасету, а також навчання і порівняння моделей. Архітектура дозволяє додавати нові моделі та алгоритми, інтегрувати додаткові джерела даних і розширювати аналітичні функції без порушення роботи інших рівнів.

Триланкова архітектура підвищує безпеку, оскільки користувачі отримують доступ до даних лише через логічний рівень, що знижує ризик несанкціонованого доступу до бази. Вона також спрощує масштабування і повторне використання модулів, дозволяючи незалежно модернізувати фронтенд, серверну логіку та обчислювальні сервіси ML. Схема архітектури наведена на рисунку 3.1, де показані взаємодії між рівнями та потоками даних для прогнозування та аналітики.

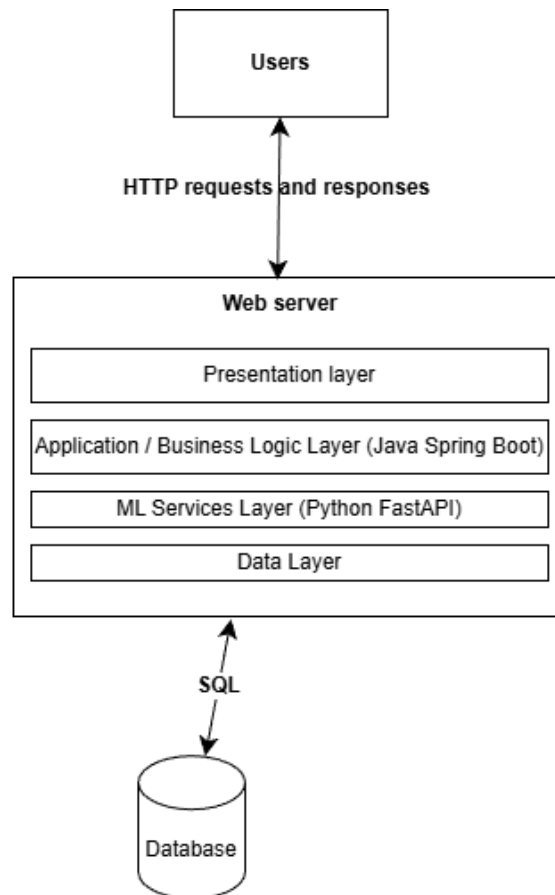


Рисунок 3.1 – Схема трирівневої архітектури

3.2 Логічне та фізичне моделювання даних

Розроблено логічну модель даних у формі ER-діаграми з використанням CASE-засобу AllFusion Data Modeler відповідно до нотації IDEF1X. Створена модель відповідає визначеним функціональним вимогам, оскільки охоплює зберігання даних про користувачів, механізм розмежування доступу на основі ролей, опис і керування моделями автоматичного прогнозування, а також фіксацію результатів прогнозів для подальшої оцінки їх точності.

Логічна модель даних, представлена на рисунку 3.2, включає основні сутності, їх атрибути та взаємозв'язки, що дає змогу формалізувати структуру інформаційних ресурсів і правила їх використання в межах системи. На основі цієї логічної моделі за допомогою CASE-засобу AllFusion Data Modeler було побудовано фізичну модель даних (рис. 3.3), у якій уточнено структуру таблиць,

типи даних, первинні та зовнішні ключі, а також обмеження цілісності, необхідні для подальшої реалізації в системі управління базами даних.

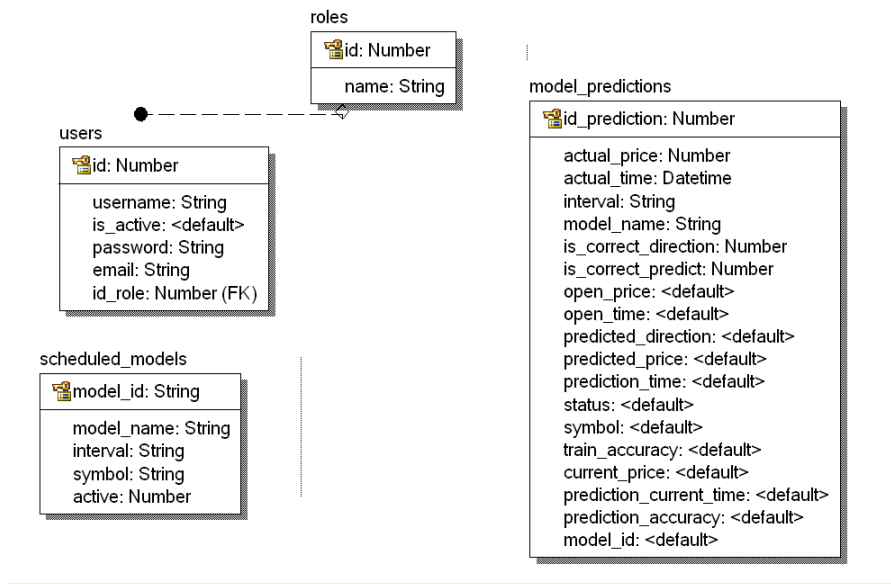


Рисунок 3.2 – Логічна модель даних

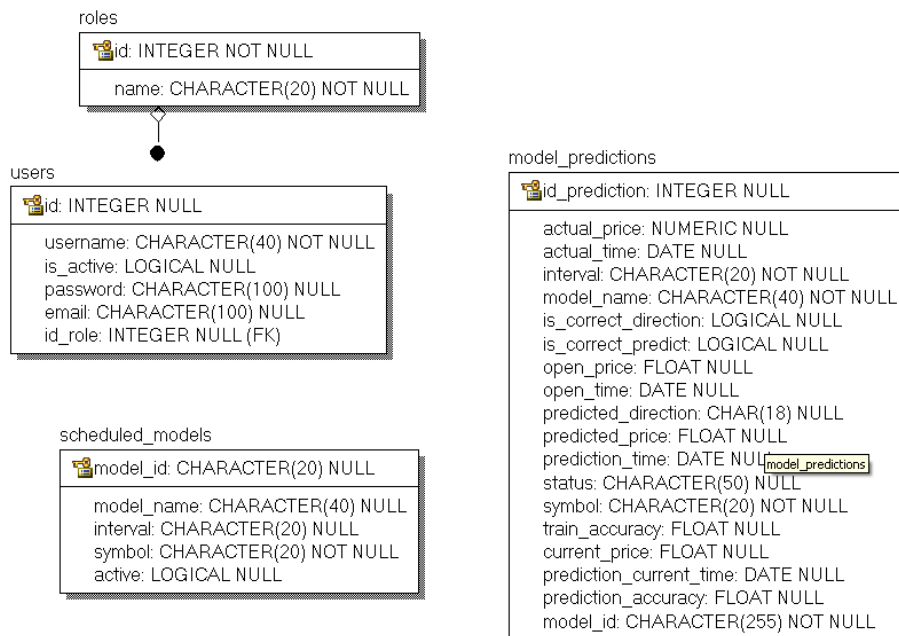


Рисунок 3.3 – Фізична модель даних

Основними сутностями моделі є `roles`, `users`, `scheduled_models` та `model_predictions`. Сутність `roles` призначена для реалізації ролей і прав доступу в системі та містить унікальний ідентифікатор ролі `id` (`Number`) як первинний ключ і

назву ролі `name` (String), що використовується під час авторизації/розмежування доступу. Сутність `users` забезпечує збереження інформації про користувачів і містить первинний ключ `id` (Number), атрибут `username` (String) для ідентифікації користувача в системі, `email` (String) для контактних даних і/або входу, `password` (String) для збереження облікових даних (у реалізації доцільно зберігати не пароль у відкритому вигляді, а його хеш), а також ознаку активності `is_active` для блокування/відновлення доступу без видалення запису. Для зв'язку з роллю використовується зовнішній ключ `id_role` (Number, FK), який посилається на `roles.id` і забезпечує відношення «одна роль багато користувачів», тобто кожен користувач має рівно одну роль, а одна роль може бути призначена багатьом користувачам.

Сутність `scheduled_models` описує перелік моделей, які система може використовувати в автоматичному режимі прогнозування, та параметри їх запуску. Первинним ключем є `model_id` (String), що однозначно ідентифікує модель у системі. Атрибут `model_name` (String) зберігає для користувача назву моделі, `interval` (String) визначає часовий інтервал/таймфрейм прогнозу, `symbol` (String) задає символ, а `active` (Number) використовується як прапорець, що вказує, чи дозволено системі автоматично запускати цю модель у плановому режимі.

Сутність `model_predictions` призначена для збереження всіх результатів прогнозування та значень, які з'являються пізніше і дозволяють оцінити якість моделі. Вона містить первинний ключ `id_prediction` (Number), який ідентифікує запис прогнозу. Для фіксації параметрів прогнозу зберігаються `model_name` (String), `interval` (String) та `symbol`, що дозволяє швидко групувати історію прогнозів за моделями/інтервалами/інструментами навіть без додаткових приєднань. Поля `open_price` і `open_time` відображають початкові умови на момент старту прогнозу, а `predicted_price`, `predicted_direction` і `prediction_time` фіксують прогнозоване значення, напрям (наприклад, зростання/падіння) та час формування прогнозу. Після того як стає відомим фактичний результат, система заповнює `actual_price` (Number) і `actual_time` (Datetime), що дозволяє порівняти прогноз із реальністю. Для автоматизованої валідації якості передбачені поля

is_correct_predict (Number) та is_correct_direction (Number), які можуть зберігати результат перевірки (наприклад, 1/0) для коректності прогнозованого значення та напряму відповідно. Також зберігаються метрики якості train_accuracy як характеристика точності у відсотках моделі на навчанні та prediction_accuracy як оцінка точності конкретного прогнозу або агрегованої точності на періоді. Поля current_price і prediction_current_time використовуються для збереження ціни та часу на момент сворення прогнозу. Атрибут status використовується для зберігання статусу прогнозу, створено, очікує перевірки.

3.3 Створення бази даних на платформі СУБД PostgreSQL

Для зберігання та обробки даних розроблюваної інформаційної системи обрано СУБД PostgreSQL. PostgreSQL є надійною реляційною СУБД із відкритим вихідним кодом, яка забезпечує високу стабільність роботи, підтримку стандарту SQL та розвинені механізми забезпечення цілісності даних. Для системи прогнозування критично важливими є коректність збереження історії прогнозів, відстеження значень, а також можливість виконувати вибірки для аналітики точності, тому використання транзакцій, обмежень цілісності та зовнішніх ключів є обов'язковим. PostgreSQL підтримує повноцінну роботу з транзакціями, що гарантує узгодженість даних під час одночасного доступу різних користувачів або фонових процесів, які автоматично формують прогнози.

PostgreSQL добре підходить, бо в системі присутні як довідникові та облікові дані, так і часові/історичні записи. Для таких наборів характерні часті операції додавання нових записів, збереження історії без втрат, а також необхідність швидкого пошуку та групування за моделями, інтервалами й часовими мітками.

Проектування структури бази даних виконано на основі попередньо розробленої логічної моделі. На етапі фізичного проектування визначено таблиці, типи даних, первинні та зовнішні ключі, а також обмеження, що підтримують цілісність і коректність інформації. Реалізована схема бази даних забезпечує збереження облікових записів користувачів та їх ролей доступу, конфігурацій

моделей для автоматичного запуску прогнозування, а також накопичення історії прогнозів із подальшою можливістю оцінювання точності.

Візуалізацію та перевірку структури створеної бази даних виконано за допомогою програмного додатка DbVisualizer, який дозволяє переглядати зв'язки між таблицями, їх атрибути, ключі та обмеження, а також формувати ER-представлення схеми на основі реальної структури PostgreSQL. Отримана схема бази даних наведена на рис. 3.4 та демонструє взаємозв'язки між таблицями користувачів і ролей, таблицею опису моделей та таблицею прогнозів, що накопичуються автоматично в процесі роботи системи.

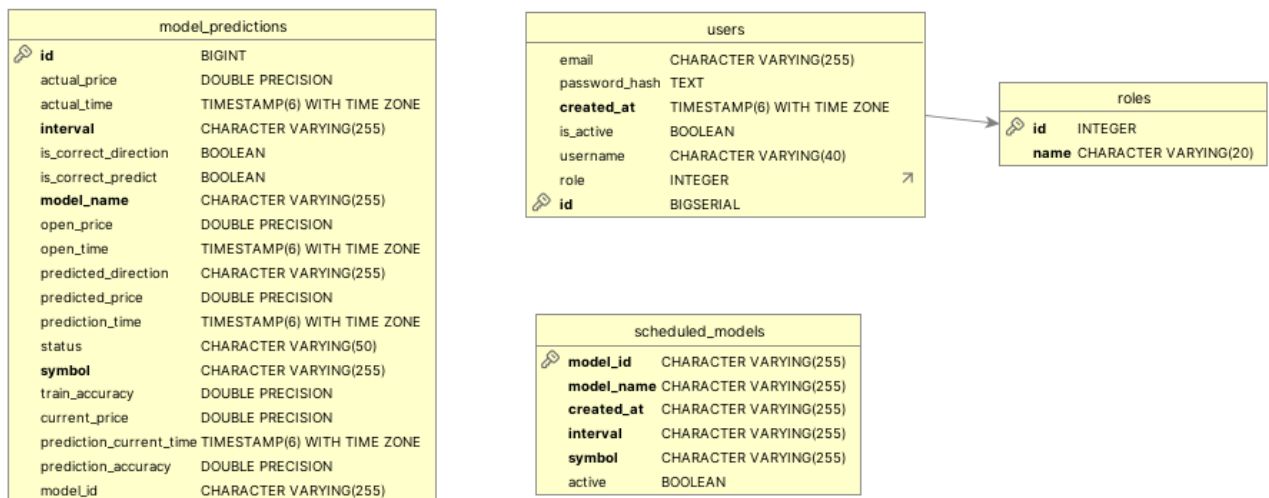


Рисунок 3.4 – Схема даних реалізованої бази даних

3.4. Джерела даних та методи їх отримання

Для задач передбачення фінансових ринків та цінових рухів активів звичайно застосовується комбінація різних типів даних, що відображають як ринкову динаміку, так і зовнішній інформаційний вплив. У подібних системах можуть використовуватися числові часові послідовності, котрі описують зміну вартостей, обсягів торгів та інтенсивності ринкової діяльності, а також дані, що характеризують поведінкові та інформаційні аспекти, зокрема відгуки суб'єктів ринку на новини та публічні оголошення.

У розробленій інформаційній системі для пророкування цінових рухів криптовалют застосовується з'єднання кількісних фінансових даних та неструктурованої текстової інформації із соціальних мереж. Такий спосіб дає змогу враховувати як об'єктивні ринкові індикатори, так і інформаційне тло, що створюється навколо криптовалютних активів у цифровому просторі.

Дані біржових котирувань добуваються з відкритого програмного інтерфейсу криптовалютної біржі Binance, яка є однією з найбільших платформ у світі. Для кожного фінансового інструменту система вивантажує історичні часові послідовності у форматі OHLCV, які містять час початку й завершення свічки, початкову вартість, найвищу та найнижчу ціни за період, кінцеву вартість, а також обсяг торгів. Отримання інформації відбувається з огляду на обмеження API біржі через поетапне завантаження часових відрізків, що дає змогу сформувати повний історичний масив даних за будь-який обраний відрізок часу.

На підставі початкових біржових відомостей у системі додатково обраховуються сукупні показники за рухомим 24-годинним вікном, зокрема обсяг торгів, найвищі та найнижчі вартості, кількість угод та відносний зсув ціни. Вказані метрики дозволяють описати поточний стан ринку, рівень його активності та мінливості, а також використовуються як додаткові ознаки для тренування моделей машинного навчання.

Окрім числових фінансових індикаторів, у системі застосовуються дані із соціальних мереж, зокрема дописи з відкритих каналів платформи Telegram. Збирання текстових даних відбувається автоматизовано за допомогою спеціалізованого клієнта, який забезпечує доступ до історії дописів у заданому часовому інтервалі. Для кожного активу створюється перелік ключових слів, що відповідають його найменуванню та поширеним варіантам згадування, після чого виконується відсіювання дописів, у яких згадується відповідна криптовалюта.

З метою оцінки інформаційного впливу соціальних мереж на ринкову поведінку активів у системі впроваджено підхід до аналізу тональності дописів, що ґрунтується на словниковому методі. Для цього застосовуються заздалегідь сформовані словники слів та висловів, які асоціюються з позитивним чи

негативним очікуванням щодо ринку. Кожному допису надається числова оцінка спрямування, яка відображає перевагу позитивних чи негативних сигналів, після чого здобутки сумуються за сталими часовими відрізками. Таким чином утворюється часова послідовність показників настрою ринку, яку можна узгодити з біржовими даними.

Передбачено автоматизований механізм отримання та оновлення даних із зовнішніх джерел з певною періодичністю. Оновлення відбувається відповідно до заданих часових інтервалів, що дає змогу зберігати актуальність інформації та узгоджувати дані з реальним становищем ринку.

Отримані дані проходять кілька етапів первинного опрацювання, спрямованих на підвищення їхньої якості та придатності до подальшого аналізу. На цьому етапі виконується приведення часових позначок до єдиного формату та часової зони, усунення прогалів і невірних значень, а також приведення числових індикаторів до єдиного масштабу. Для часових послідовностей додатково виконується агрегація даних відповідно до обраних інтервалів, що дає змогу узгодити дані з різних джерел та забезпечити їхнє коректне поєднання.

3.5. Аналіз та вибір моделей машинного навчання

Задача прогнозування фінансових часових рядів належить до класу складних прикладних задач машинного навчання, оскільки такі дані характеризуються високою волатильністю, наявністю шуму, нелінійними залежностями та залежністю від попередніх значень. У зв'язку з цим у сучасних інформаційних системах застосовуються різні напрями моделей машинного навчання, кожен з яких має власні переваги та обмеження залежно від типу даних і поставленої задачі.

Класичні алгоритми машинного навчання, зокрема ансамблеві методи на основі дерев рішень, добре підходять для роботи з табличними даними, що містять числові та категоріальні ознаки. Такі моделі ефективно виявляють нелінійні залежності між ознаками, стійкі до викидів та не потребують складної попередньої нормалізації даних.

Нейронні мережі, орієнтовані на роботу з послідовностями, зокрема рекурентні архітектури, широко застосовуються для аналізу часових рядів, оскільки здатні враховувати довгострокові та короткострокові часові залежності. Моделі типу LSTM дозволяють зберігати інформацію про попередні стани системи та ефективно працюють у середовищі, де важливу роль відіграє контекст попередніх спостережень. Водночас згорткові нейронні мережі можуть використовуватися для автоматичного виділення локальних патернів і структур у часових послідовностях, що є корисним для виявлення повторюваних ринкових формацій.

Окремий напрям становлять гібридні моделі, які поєднують переваги різних підходів. Такі архітектури дозволяють одночасно використовувати здатність нейронних мереж до аналізу часових залежностей та високу точність ансамблевих методів при роботі з агрегованими ознаками. Використання гібридних моделей є доцільним у випадках, коли система оперує різномірними даними та потребує підвищеної стабільності прогнозу.

У межах розробленої інформаційної системи реалізовано декілька моделей машинного навчання, кожна з яких орієнтована на вирішення окремих аспектів задачі прогнозування.

Моделі на основі градієнтного бустингу, зокрема XGBoost, LightGBM та CatBoost, використовуються для прогнозування на основі числових фінансових ознак і технічних індикаторів. Ці моделі відзначаються високою швидкістю навчання, можливістю ефективною роботи з великими наборами даних та стабільними результатами навіть за наявності шуму в даних. CatBoost додатково демонструє підвищену ефективність у випадках, коли в даних присутні складні нелінійні взаємозв'язки між ознаками.

Для аналізу часових залежностей у системі застосовуються рекурентні нейронні мережі типу LSTM, а також згорткові нейронні мережі одомірної архітектури. LSTM-моделі використовуються для врахування послідовної природи фінансових даних і прогнозування майбутніх значень на основі попередніх станів ринку. Одновимірні згорткові нейронні мережі застосовуються для виділення

локальних патернів у часових рядах, що дозволяє підвищити чутливість моделі до короткострокових змін.

Окрім цього, у системі реалізовано гібридні архітектури, зокрема поєднання згорткових та рекурентних нейронних мереж, а також комбінації моделей градієнтного бустингу з LSTM. Такі моделі використовуються для підвищення якості прогнозу шляхом поєднання різних типів ознак та механізмів обробки даних.

Для пришвидшення навчання нейронних мереж та обробки великих обсягів даних у системі передбачена можливість використання графічних процесорів. Застосування GPU-обчислень дозволяє значно скоротити час навчання складних моделей і забезпечує масштабованість системи при роботі з високочастотними часовими рядами.

3.7. Проектування взаємодії компонентів системи

Інформаційна система побудована за принципом розподілу відповідальності між окремими компонентами, кожен з яких виконує чітко визначені функції. Такий підхід дозволяє забезпечити гнучкість архітектури, спростити супровід системи та створити передумови для її масштабування. У межах системи виділено серверну частину, що відповідає за бізнес-логіку та управління даними, окремий сервіс машинного навчання, який реалізує обчислювальні операції прогнозування, а також клієнтську частину, що забезпечує взаємодію з користувачем.

Python ML-сервіс використовується для збору даних, попередньої обробки, навчання моделей машинного навчання та генерації прогнозів. Java-бекенд виконує роль центрального керуючого компонента, який відповідає за авторизацію користувачів, управління запитами, збереження результатів та координацію взаємодії між фронтендом і ML-сервісом. Такий поділ дозволяє ізолювати обчислювально навантажені задачі машинного навчання від бізнес-логіки системи та забезпечити стабільну роботу всіх компонентів.

Взаємодія між Python ML-сервісом та Java-бекендом реалізована на основі REST API, що забезпечує стандартизований обмін даними між компонентами

системи. Обмін інформацією здійснюється у форматі JSON, що спрощує серіалізацію даних і забезпечує сумісність між різними мовами програмування.

Python ML-сервіс надає набір HTTP-ендпоінтів, через які Java-бекенд ініціює основні операції, пов'язані з життєвим циклом моделей машинного навчання. Реалізовано ендпоінти для запуску навчання моделей на історичних даних, виконання прогнозів на основі актуальних часових рядів, а також отримання метрик якості та метаданих моделей. Запити на навчання та прогнозування можуть виконуватися як у синхронному режимі, так і з використанням асинхронних механізмів, що дозволяє уникнути блокування основних потоків обробки запитів у Java-бекенді.

Асинхронна обробка використовується насамперед для ресурсоємних операцій, таких як навчання моделей на великих обсягах даних. У цьому випадку Java-бекенд ініціює відповідний запит до Python ML-сервісу та отримує результат після завершення обчислень. Синхронні запити застосовуються для швидких операцій, наприклад отримання прогнозу або метаданих уже навчених моделей.

Фронтенд інформаційної системи взаємодіє з серверною частиною виключно через Java-бекенд, який виступає єдиною точкою доступу до бізнес-логіки та даних. Це дозволяє централізувати контроль доступу, забезпечити безпеку та уникнути прямої взаємодії клієнтської частини з ML-сервісом.

Фронтенд звертається до Java-бекенду через набір REST-маршрутів для отримання списків доступних моделей, запуску процесів навчання, перегляду історії прогнозів та отримання результатів прогнозування. Для зменшення навантаження на сервер та підвищення швидкодії системи передбачено кешування результатів прогнозів, що дозволяє повторно використовувати вже обчислені значення без необхідності повторного виклику ML-сервісу.

Проектування системи з урахуванням вимог до продуктивності та масштабованості є важливим аспектом її практичного застосування. З огляду на обчислювальну складність моделей машинного навчання та потенційне зростання кількості користувачів, у системі реалізовано архітектурні рішення, що дозволяють

ефективно розподіляти навантаження та масштабувати окремі компоненти незалежно один від одного.

Розроблена система побудована на основі мікросервісної архітектури з чітким розподілом функціональних ролей між окремими технологічними компонентами. Центральним елементом серверної частини є Spring-базований керуючий сервер, який виконує функції координації, маршрутизації запитів і балансування навантаження між декількома екземплярами Java-бекенду. Такий підхід дозволяє масштабувати бізнес-логіку системи незалежно від інших компонентів і забезпечує стабільну обробку великої кількості клієнтських запитів.

Java-бекенд реалізований із використанням технологій Spring Boot та REST-архітектури і відповідає за управління моделями, авторизацію користувачів, збереження результатів прогнозування та взаємодію з сервісами машинного навчання. Кожен Java-сервер працює як незалежний обробник запитів і за необхідності може звертатися до відповідних Python ML-сервісів для виконання обчислювально складних операцій прогнозування. Розподіл навантаження між Java-екземплярами здійснюється автоматично за допомогою балансувальника, що дозволяє рівномірно використовувати ресурси системи.

Python ML-сервіси функціонують як окремі спеціалізовані обчислювальні вузли, призначені для навчання моделей та генерації прогнозів. Кожен Python-сервер після запуску проходить процедуру реєстрації в системі, під час якої передає метадані про підтримувані торгові інструменти, доступні моделі машинного навчання та характеристики обчислювальних ресурсів. Завдяки цьому система динамічно формує перелік доступних ML-сервісів і може використовувати їх для виконання запитів користувачів без жорсткої прив'язки до конкретного екземпляра сервісу.

У разі ініціації запиту на навчання моделі або отримання прогнозу Java-бекенд звертається до списку зареєстрованих Python ML-серверів і виконує маршрутизацію запиту до одного або декількох сервісів, які підтримують відповідний торговий інструмент і модель. Це дозволяє реалізувати гнучку систему вибору обчислювального вузла та забезпечує можливість паралельної роботи з

кількома моделями або активами одночасно. Отримані від Python-сервісів результати агрегуються Java-бекендом і передаються клієнтській частині у стандартизованому форматі.

Масштабування Python ML-сервісів здійснюється за рахунок горизонтального розгортання нових екземплярів, які автоматично реєструються в системі після запуску.

3.4. Реалізація ML сервісу на Python

Python ML-сервіс реалізований як окремий самостійний компонент системи, побудований за модульним принципом, що забезпечує гнучкість, розширюваність та можливість масштабування залежно від кількості активів і моделей, які використовуються у системі. Архітектура сервісу орієнтована на підтримку декількох типів моделей машинного навчання та різних торгових символів без необхідності внесення змін у базову логіку роботи сервісу.

Основою ML-сервісу є чітке розділення відповідальностей між модулями. Окремі компоненти відповідають за отримання даних, їх збереження, обробку, навчання моделей, виконання прогнозів та керування версіями моделей.

Сервіс підтримує динамічне підключення моделей машинного навчання. Кожна модель реалізується як окремий Python-модуль, який повинен реалізовувати стандартизований інтерфейс навчання, прогнозування та збереження стану. Завдяки цьому до системи можна додавати нові типи моделей без змін у керуючому коді сервісу. Python ML-сервер може обслуговувати довільну кількість моделей та торгових символів, обмежену лише обчислювальними ресурсами.

Для кожної моделі використовується версіонування, що дозволяє зберігати історію навчань, повторно використовувати попередні версії та виконувати донавчання на нових даних. Навчені моделі та їх метадані зберігаються у файловій системі у структурованому вигляді, що спрощує їх перенесення, резервне копіювання та масштабування між серверами.

ML-сервіс спроектований таким чином, що може бути розгорнутий у кількох екземплярах. Кожен Python-сервер може бути спеціалізований на певному наборі торгових символів або моделей. Під час запуску сервіс може реєструвати доступні моделі та підтримувані активи, після чого Java-бекенд використовує цю інформацію для маршрутизації запитів. Це дозволяє динамічно балансувати навантаження та масштабувати систему горизонтально шляхом додавання нових ML-вузлів.

ML-сервіс реалізовано на базі FastAPI як набір модулів, що формують цілісний пайплайн: оновлення даних, підготовка датасету, навчання/прогноз, збереження версій моделей. Точкою входу є FastAPI-застосунок, який підключає два основні роутери: модуль обробки даних та модуль метаданих. Під час старту сервісу виконується `startup`-подія, в якій запускаються фонові задачі оновлення даних. Це дозволяє сервісу працювати не тільки у режимі “запит-відповідь”, а й підтримувати актуальний стан датасетів автоматично, без ручного втручання користувача. На рис. 3.5 наведено схему компонентів ML-сервісу пайтон.

Функціонально сервіс поділяється на три рівні. Перший рівень це API-шар, який містить ендпоінти `/load_data`, `/train`, `/predict`, `/retrain` і набір службових методів `/available_models`, `/model_vers`, `/model_vers_info`, `/intervals`, `/delete_model`. API-шар не містить ML-логіки, а лише виконує валідацію параметрів (символ, інтервал, діапазон дат, назва моделі), завантажує необхідні дані з CSV та передає їх на сервісний рівень.

Другий рівень це шар сервісів і підсистем даних, який відповідає за формування датасету. Компонент `BinanceDataProvider` інкапсулює отримання історичних свічок. Компонент `CSVStorageService` відповідає за файлове збереження даних у структурі `datasets/<SYMBOL>/data/<INTERVAL>.csv`. `DataUpdaterService` реалізує інкрементальне оновлення: він визначає останню доступну свічку за `open_time`, після чого завантажує тільки нові дані. Такий режим суттєво зменшує трафік і час оновлення при тривалому використанні сервісу, оскільки не потребує повторного отримання повної історії.

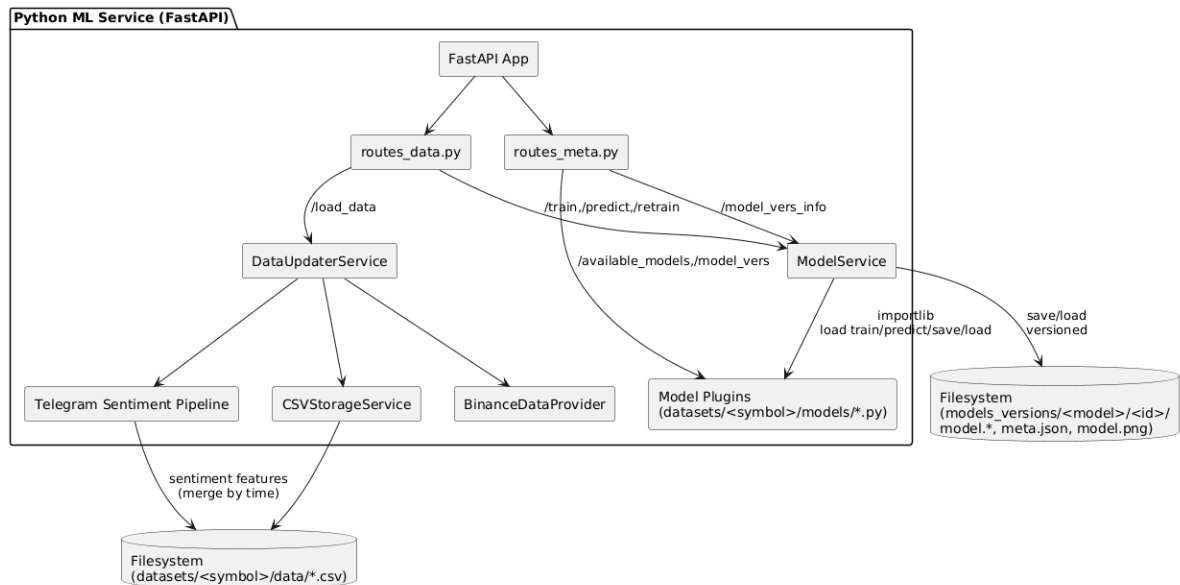


Рисунок 3.5 – Схема компонентів ML-сервісу

Паралельно з оновленням ринкових даних виконується окрема задача збору інформаційних сигналів із Telegram. Під час оновлення використовується словник ключових слів, прив’язаний до конкретного торгового символу. Наприклад, для BTCUSDT цк “btc”, “bitcoin” тощо. Далі зібрані повідомлення проходять спрощену NLP-обробку: визначається напрям інформаційного сигналу за словниками “позитивних” і “негативних” маркерів, після чого формується агрегована оцінка `sentiment_score` у часових проміжках.

Третій рівень модуль керування моделями (`ModelService`), який забезпечує динамічне підключення алгоритмів і управління їх версіями. Вибір конкретної моделі відбувається через імпорт Python-модуля за шаблоном `datasets.<SYMBOL>.models.<model_name>`, що дозволяє додавати нові моделі у вигляді окремих файлів без змін у роутерах або в `ModelService`. Фактично кожна модель виступає як “плагін”, який реалізує стандартні операції `train()`, `predict()`, `save()` та `load()`, а за наявності `rdate()` для донавчання.

Для оптимізації роботи використовується кеш завантажених моделей у пам’яті (`_model_cache`). Якщо модель уже була завантажена, повторні запити прогнозування не потребують читання файлів із диска, що зменшує затримки для

частих викликів `/predict`. Збереження моделі реалізовано як версіонована структура у файлової системі: для кожного поєднання символ-модель-версія створюється окрема директорія, у якій зберігаються файл моделі, метадані і службові артефакти, зокрема `model.png` з графіком тренування. Метадані включають часовий діапазон, інтервал, кількість навчальних прикладів, тривалість навчання і значення точності, що дозволяє відслідковувати еволюцію моделей і порівнювати версії між собою.

Окремо реалізовано сценарій повторного навчання `/retrain`, який створює новий `model_id` та формує нову версію моделі. Якщо конкретний модуль моделі підтримує `update()`, виконується донавчання поверх уже існуючого стану. Якщо ні, то модель перевчається заново, але зберігається як нова версія. Завдяки цьому ML-сервіс може масштабуватися не тільки горизонтально, а й логічно через збільшення кількості підтримуваних символів і моделей, які підключаються як незалежні модулі.

Реалізовано механізм реєстрації ML-серверів у “центральному” сервісі керування. Кожен Python-вузол під час запуску надсилає набір своїх можливостей (підтримувані символи та моделі, а також унікальний ключ доступу). Центральний компонент зберігає відповідність “символ–модель–сервер” і надалі використовує її для маршрутизації запитів. Такий підхід дозволяє додавати нові ML-вузли без змін у клієнтській частині та забезпечує кероване масштабування при зростанні навантаження.

3.5. Реалізація Java бекенду

Java-бекенд реалізовано як центральний керуючий компонент системи, який об’єднує користувацьку логіку, керування доступом, взаємодію з ML-сервісами та довготривале збереження результатів у PostgreSQL. На відміну від Python-компонента, який фокусується на обчисленнях та генерації прогнозу, Java-сервер відповідає за життєвий цикл прогнозів: ініціацію запиту, збереження результатів, планову генерацію прогнозів, а також подальшу валідацію прогнозів на основі фактичної ціни в момент настання прогнозованого часу. Це дозволяє будувати

історію передбачень, формувати статистику ефективності моделей і забезпечувати зручне відображення даних у клієнтській частині.

Архітектурно застосунок має класичну структуру Spring Boot із поділом на рівні controllers, services, repositories, entities, що забезпечує зрозумілу підтримку та масштабування бізнес-логіки [16]. Контролери реалізують API-рівень і приймають HTTP-запити, сервіси інкапсулюють бізнес-правила та інтеграцію з Python, репозиторії реалізують доступ до PostgreSQL через Spring Data JPA, а сутності відповідають за мапінг даних у таблиці. На рис. 3.6 наведено схему компонентів Java-сервісу.

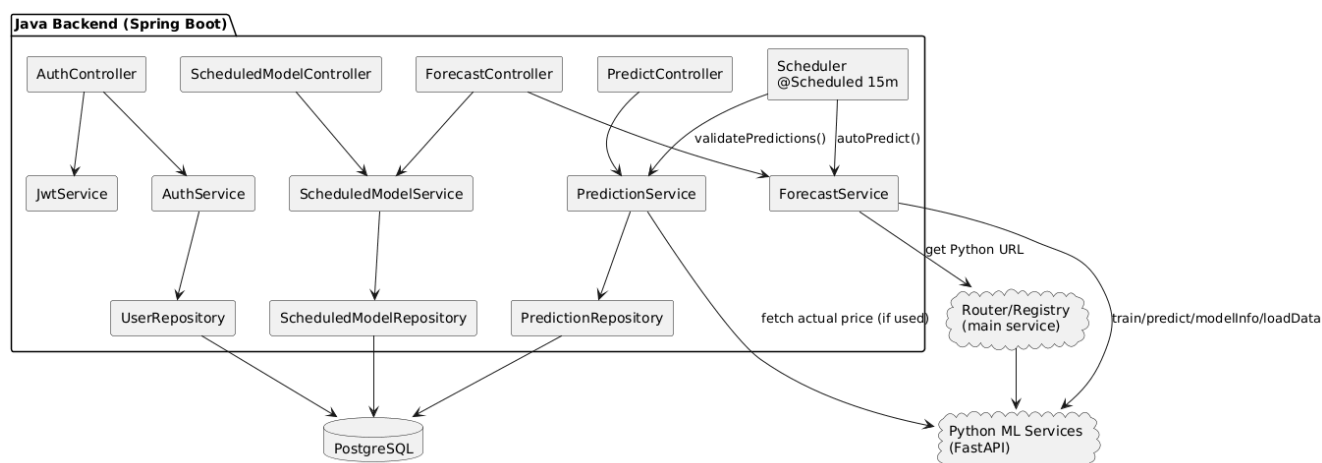


Рисунок 3.6 – Схема компонентів Java-сервісу

На рівні конфігурації безпеки використовується підхід із JWT-фільтром (наприклад `JwtAuthFilter`), який перехоплює запити, перевіряє токен та встановлює контекст безпеки. Таким чином, доступ до операцій прогнозування та перегляду історії прогнозів може бути обмежений лише авторизованими користувачами, тоді як `/auth/**` залишається відкритим для первинного входу.

Для взаємодії з ML-функціональністю використовується контролер `ForecastController`, який надає набір API-методів для операцій над моделями та даними. Через цей контролер реалізовано отримання доступних інтервалів (`/intervals`), ініціацію завантаження/оновлення даних (`/load_data`), отримання списку доступних моделей прогнозування (`/forecast_models`), отримання інформації про

конкретну версію моделі (/getModelInfo), запуск навчання (/train_model) та прогнозування (/predict_model). Окремо реалізовано видалення версії моделі (/delete_model), де після успішного видалення моделі також видається відповідний запис у таблиці запланованих моделей для автопрогнозування.

Задача Java-серверу сформулювати запит до ML-сервісу, передати параметри, отримати відповідь та зберегти її в PostgreSQL у вигляді запису про прогноз.

Для роботи саме з історією прогнозів та фільтрацією реалізовано PredictController з базовим шляхом /api/predictions. Ендпоінт get_predictions дозволяє отримувати прогнозні записи за статусом і додатковими фільтрами: за символом, початковою та кінцевою датою. Це забезпечує готовий механізм під побудову статистики й аналітики на фронтенді, оскільки дані можна вибирати як за станом прогнозу, так і за часовими діапазонами.

Додатково в PredictController реалізовано ендпоінт, який повертає набір торгових сесій у форматі UTC. Це є допоміжним інструментом для аналітики, що дозволяє прив'язувати прогнози до часових зон та торгових активностей ринку.

У системі реалізовано окрему підсистему “автопрогнозування”, яка базується на таблиці scheduled_models. Вона описує, які саме моделі повинні генерувати прогноз регулярно. Для керування цими записами реалізовано ScheduledModelController з маршрутом /scheduled-models, який підтримує операції отримання всіх запланованих моделей, додавання нової моделі в автопрогноз та видалення зі списку. Це дозволяє користувачу або адміністративній логіці вмикати/вимикати автоматичні прогнози без редагування конфігурацій або коду.

Реалізовано дві задачі, що виконуються кожні 15 хвилин і працюють у транзакційному режимі. Перша задача періодично читає таблицю scheduled_models і формує запити до Python ML-сервісів для генерації нових прогнозів на основі актуальної свічки. Отримані результати зберігаються у таблицю прогнозів із початковим статусом PENDING. Друга задача періодично переглядає всі записи зі статусом PENDING, перевіряє чи настав прогнозований момент часу, отримує фактичну ціну і виконує валідацію: обчислюється коректність напрямку,

коректність прогнозованого значення та інтегральний показник точності. Після цього запис оновлюється і переводиться в статус `VALIDATED`.

Такий підхід дозволяє системі автоматично накопичувати статистику якості моделей у на практиці, а не лише на даних для тестування.

Python-сервер при старті передає на головний сервіс перелік доступних моделей та підтримуваних символів і отримує ключ доступу, який зберігається в системі та використовується для контролю взаємодії між компонентами.

PostgreSQL використовується як єдине джерело для історії прогнозів і налаштувань автопрогнозування. У базі зберігаються як параметри прогнозу, так і результати валідації.

3.6. Реалізація фронтенду

Фронтенд системи реалізовано як односторінковий веб-застосунок, розроблений на базі React 18 із використанням TypeScript. Для запуску та збірки використано Vite, що забезпечує швидкий dev-сервер, мінімальний час гарячого оновлення та оптимізовану production-збірку. Інтерфейс орієнтований на взаємодію з бекендом через HTTP-запити та на відображення динаміки криптовалют у вигляді графіків і історії прогнозів, при цьому логіка користувацького доступу реалізована через JWT-автентифікацію.

Архітектура клієнтської частини побудована за модульним принципом. У каталозі `src/pages` зосереджені сторінки `Home`, `ForecastPage`, `ModelsManager`, `PredictStats`, `LoginPage`, `RegisterPage`, `ProfilePage`, які відповідають маршрутам застосунку. У каталозі `src/components` винесені повторно використовувані UI-блоки та функціональні секції сторінок. Для сторінки прогнозування сформовано окремий набір компонентів у `components/forecastPage`, де виділено модулі керування прогнозами (`predictManage`), відображення історії (`ForecastHistory`), секції запуску навчання/завантаження даних (`TrainSection`, `LoadDataSection`), а також компоненти графіків (`CandlestickChart`, `LinePriceChart`). Така декомпозиція

зменшує зв'язність коду, спрощує підтримку і дозволяє поступово розширювати функціональність без переписування сторінок повністю.

Маршрутизація реалізована через `react-router-dom`. У файлі `App.tsx` визначено основні маршрути: головна сторінка (`/`), сторінки авторизації та реєстрації (`/login`, `/register`), модуль прогнозування (`/forecast`), модуль статистики (`/stats`), керування моделями та автопрогнозуванням (`/models_config`), а також сторінка профілю (`/profile`). Додатково застосовано механізм захисту приватних маршрутів через компонент `ProtectedRoute`: якщо токен у стані автентифікації відсутній, користувач перенаправляється на сторінку входу. Це дозволяє відокремити відкриті сценарії (ознайомлення та вхід) від функцій, які потребують авторизації (перегляд профілю й робота з прогнозами/моделями).

Стан авторизації керується через `Zustand` (`useAuthStore`). Після виконання входу або реєстрації фронтенд отримує JWT-токен від бекенду і зберігає його у глобальному стані, після чого цей токен використовується для доступу до захищених ендпоінтів. Водночас логіка авторизації на клієнті реалізована так, щоб не дублювати серверні обмеження: фронтенд лише керує UX-поведінкою, а остаточний контроль доступу здійснюється бекендом.

Взаємодія з бекендом виконується через `axios`, а для керування запитами, повторними завантаженнями та актуальністю даних використовується `tanstack/react-query`. Такий підхід дає можливість стандартизувати отримання даних, централізовано обробляти помилки та підвантажувати оновлення без ручного керування життєвим циклом запитів. В інтерфейсі це проявляється у тому, що після створення прогнозів або змін у конфігурації моделей дані на сторінках оновлюються автоматично в межах повторного запиту або інвалідації кешу.

Сторінка `Home` має інформативний характер і використовується як початковий екран: вона містить загальний опис системи, її призначення та навігацію до основних модулів. Основна прикладна робота виконується на сторінці `ForecastPage`. У цьому модулі користувач може переглядати історію прогнозів, бачити щойно згенеровані прогнози, а також аналізувати динаміку ціни через графіки. Для візуалізації біржових даних застосовуються бібліотеки `react-financial-`

charts і recharts, які дозволяють будувати як свічкові графіки, так і лінійні представлення, зручні для порівняння значень та результатів прогнозування.

Модуль ModelsManager використовується як центр керування моделями машинного навчання. На цій сторінці користувач може отримати перелік доступних моделей, переглянути інформацію про модель/версію, ініціювати створення прогнозу для обраної моделі, а також додати модель до списку автопрогнозування. Таким чином фронтенд відображає логіку, реалізовану на бекенді через таблицю `scheduled_models`: інтерфейс дозволяє керувати тим, які моделі повинні запускатися регулярно без ручного втручання.

Сторінка PredictStats виконує функцію формування аналітичного модуля для оцінювання якості прогнозування. Вона відображає статистику як для конкретної моделі з урахуванням усіх її версій і метрик навчання, так і узагальнену статистику по всіх моделях одночасно. При цьому буде враховуватися не лише результат тренування на історичному розбитті, а і практична точність прогнозів після їх валідації на реальних даних, що зберігаються у PostgreSQL.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТА ЕФЕКТИВНІСТЬ СИСТЕМИ

4.1. Порівняння моделей та їх статистичні результати

У межах дослідження реалізовано та проаналізовано декілька моделей машинного навчання та глибокого навчання, орієнтованих на прогнозування короткострокової динаміки криптовалют. Метою експериментів є порівняння точності прогнозування, стабільності результатів та обчислювальної ефективності моделей на історичних даних.

4.1.1. Перелік та характеристика досліджуваних моделей

Для проведення експериментального дослідження використано декілька моделей машинного навчання та глибокого навчання, які відрізняються за принципами побудови, способом обробки даних та обчислювальною складністю.

У межах дослідження використано моделі: XGBoost, LightGBM, CatBoost, LSTM, 1D CNN, CNN LSTM, XGBoost LSTM, GRU.

XGBoost є ансамблевим методом градієнтного бустингу над деревами рішень, який широко застосовується для роботи з табличними даними. Модель добре обробляє технічні індикатори та агреговані числові ознаки, є стійкою до шуму та забезпечує високу швидкість навчання за відносно невеликих обчислювальних витрат [17].

LightGBM є оптимізованим варіантом градієнтного бустингу, орієнтованим на прискорене навчання та ефективну роботу з великими наборами даних [18]. Завдяки використанню алгоритмів на основі гістограм модель демонструє високу продуктивність та добре масштабується при збільшенні обсягу навчальних даних.

CatBoost є моделлю градієнтного бустингу, розробленою з акцентом на зменшення переобучення та коректну роботу з нелінійними залежностями.

Особливістю цієї моделі є стабільність результатів та здатність ефективно працювати з комплексними взаємозв'язками між ознаками [19].

LSTM є рекурентною нейронною мережею довготривалої пам'яті, спеціально призначеною для аналізу часових рядів. Дана модель здатна враховувати довгострокові залежності у фінансових даних, що є важливим для прогнозування ринкової динаміки з урахуванням попередніх значень [20].

1D CNN є згортковою нейронною мережею, адаптованою для роботи з одномірними часовими послідовностями. Модель дозволяє автоматично виділяти локальні патерни та короткострокові закономірності у фінансових даних, що може покращувати якість прогнозування на коротких інтервалах [21].

CNN LSTM є гібридною моделлю, яка поєднує переваги згорткових та рекурентних нейронних мереж. Згорткова частина використовується для автоматичного виділення інформативних ознак, тоді як LSTM компонент відповідає за аналіз часової динаміки та довготривалих залежностей.

XGBoost LSTM є комбінованим підходом, що поєднує класичні методи машинного навчання з глибинними нейронними мережами. У межах цієї моделі статистичні та технічні ознаки обробляються за допомогою XGBoost, після чого результати використовуються як вхідні дані для LSTM, що дозволяє одночасно враховувати як структурні характеристики даних, так і їх часову динаміку [22].

GRU є підходом до моделювання часових рядів, що базується на глибинних рекурентних нейронних мережах. У цій моделі попередньо оброблені статистичні та технічні ознаки масштабуються та подаються у вигляді послідовностей на вхід GRU-мережі. Такий підхід дозволяє враховувати як поточний стан ринку, так і його динаміку у часі. GRU ефективно вловлює часові залежності та використовує контекст кількох попередніх кроків для точного прогнозування майбутніх значень. Завдяки меншій кількості параметрів у порівнянні з LSTM, GRU забезпечує швидше навчання та кращу узагальнювальну здатність, що особливо важливо при роботі з фінансовими даними у реальному часі [23].

4.1.2. Методика оцінювання якості моделей

Для оцінювання якості побудованих моделей прогнозування цінних рядів використовувалося поєднання класичних регресійних метрик та спеціалізованих показників, що краще відображають особливості часової динаміки фінансових даних. Такий підхід забезпечує комплексне уявлення як про абсолютну точність прогнозу, так і про спроможність моделей правильно передбачати напрям зміни ціни.

На першому етапі розраховувалися традиційні регресійні метрики, зокрема MAE, що характеризує середню абсолютну похибку прогнозу, RMSE, який додатково підсилює вплив великих відхилень, та MAPE відносна похибка у відсотках. Хоча ці показники дозволяють оцінити загальну точність моделі, вони не повністю відображають поведінку фінансових часових рядів, де велике значення має правильність прогнозування напрямку руху.

З огляду на це було застосовано додаткові спеціалізовані метрики. Direction Accuracy характеризує частку випадків, коли модель коректно вгадує напрям зміни ціни, що є критично важливим для сценаріїв застосування у торгових системах. Додатково оцінювалася точність близькості прогнозу до фактичного значення у межах заданого відсоткового допуску, що визначалося через порівняння абсолютної похибки з допустимим діапазоном відхилень. Для уникнення проблеми нестабільності відносних метрик при малих значеннях ціни використовувався симетричний показник помилки sMAPE, на основі якого формувалася метрика sMAPE Accuracy, що інтерпретується як рівень точності у відсотках [24].

Узагальнення різних аспектів точності здійснювалося за допомогою інтегрального показника Final Score, який є середнім значенням Direction Accuracy, Close Accuracy та sMAPE Accuracy. Така форма дозволяє оцінити модель не лише з погляду числової помилки, а й з точки зору поведінкової відповідності ринку, що робить можливим коректне порівняння між собою різних типів моделей як бустингових, так і нейромережевих, незалежно від їхньої складності чи архітектури.

Ураховуючи те, що набір доступних ринкових ознак може змінюватися, а деякі додаткові 24-годинні параметри інколи відсутні, система оцінювання була побудована таким чином, щоб автоматично адаптуватися до структури вхідних даних. Моделі використовували лише ті ознаки, які фактично присутні у датасеті, а відсутні фічі ігнорувалися без впливу на процес навчання чи прогнозування. Це забезпечило універсальність методики та її стійкість до змін у джерелах даних.

Процедура оцінювання була однаковою для всіх моделей: дані поділялися на навчальну та тестову частини з дотриманням часової послідовності, навчання відбувалося виключно на навчальному відрізку, а оцінювання на тестовому. Додатково будувалася графічна інтерпретація результатів прогнозування, що давало змогу візуально оцінити відповідність між фактичними та прогнозованими значеннями й визначити характерні відхилення моделей.

4.1.3. Кількісні результати

Для оцінювання ефективності розробленої системи прогнозування було проведено серію експериментів з навчання та тестування моделей машинного і глибинного навчання на різних часових інтервалах. З метою аналізу як короткострокових, так і середньострокових прогнозів використовувалися два набори даних з різною гранулярністю.

У першому експерименті моделі навчалися на історичних даних тривалістю один рік з інтервалом у одну годину, що дозволяє дослідити поведінку моделей на довгих часових відрізках та оцінити їх здатність виявляти загальні тренди ринку. Такий підхід є доцільним для середньо- та довгострокового прогнозування динаміки криптовалют.

У другому експерименті навчання проводилося на даних за три місяці з інтервалом у 15 хвилин, що забезпечує вищу деталізацію часових рядів та дає змогу проаналізувати ефективність моделей у короткостроковому прогнозуванні. Цей сценарій є особливо актуальним для задач оперативного аналізу ринку та побудови торгових стратегій.

Оцінювання здійснювалося за фінальною інтегральною оцінкою, середньою абсолютною похибкою (MAE), середньоквадратичною похибкою (RMSE), відсотковою похибкою (MAPE), симетричною середньою абсолютною відсотковою похибкою (sMAPE), кількістю навчальних зразків та часом навчання моделей.

У таблиці 4.1 наведено кількісні результати експериментів для моделей XGBoost, LightGBM, CatBoost, 1D CNN, CNN-LSTM, XGBoost-LSTM та GRU на таймфреймі 1h терміном в 1 рік.

Таблиця 4.1 – Результати експериментів навчання моделей на 1h таймфреймом

Модель	MAE	RMS E	MAPE (%)	Smape	К. записів	Точність напрямку(%)	Фінальна оцінка(%)	Час навчання (с)
1D CNN	1289.19	1729.05	1.30	98.70	8774	49.43	65.79	65
CatBoost	516.24	714.44	0.53	99.47	8774	55.88	80.5	10,4
CNN-LSTM	3040.92	3780.93	3.26	96.81	8774	49.43	55.88	141.68
gruModel	1771.14	2024.49	1.79	98.23	8774	46.86	56.29	254.6
LightGBM	587.39	825.26	0.596	99.40	8774	51.88	77.65	1.8
XGBoost	470.27	647.77	0.47	99.53	8774	49.34	79.14	64.7
XGBoost LSTM	1505.89	1762.14	1.59	98.42	8774	71.02	69.47	205.836

Аналіз значень похибок показує, що найменші значення MAE, RMSE та MAPE продемонструвала модель XGBoost, що свідчить про її високу точність у відтворенні значень цін. Близькі за якістю результати також показала модель CatBoost, яка додатково характеризується найменшим часом навчання серед усіх розглянутих моделей. Модель LightGBM поступається за точністю, але забезпечує найшвидше навчання.

Нейронні моделі 1D CNN, CNN-LSTM та GRU демонструють вищі значення похибок у порівнянні з ансамблевими методами. Це може бути пов'язано з обмеженим обсягом даних та складністю архітектур, які потребують більшої кількості навчальних прикладів для ефективного узагальнення. Але також потрібно

звернути увагу, що гібридна модель XGBoost-LSTM показала найвищу точність визначення напрямку зміни ціни, що свідчить про її потенціал у задачах аналізу трендів.

Час навчання моделей суттєво відрізняється залежно від використаного підходу. Ансамблеві методи, зокрема CatBoost і LightGBM, навчаються значно швидше, тоді як нейронні моделі потребують у десятки разів більше часу. При цьому збільшення часу навчання не завжди призводить до покращення точності прогнозу.

У таблиці 4.2 наведено кількісні результати експериментів для моделей на таймфреймі 15m терміном в 3 місяці.

Таблиця 4.2 – Результати експериментів навчання моделей на 15m таймфреймом

Модель	MAE	RMS E	MAPE (%)	Smape	К. записів	Точність напрямку(%)	Фінальна оцінка(%)	Час навчання (с)
1D CNN	1137.37	1447.49	1.27	98.72	8737	49.57	66.04	50.0
CatBoost	254.44	350.90	0.28	99.71	8737	50.37	82.50	5.57
CNN-LSTM	1714.92	2126.74	1.94	98.09	8737	48.14	60.16	119.18
gruModel	1824.48	2158.38	2.009	97.96	8737	46.48	57.00	318.02
LightGBM	263.6	384.193	0.29	99.70	8737	37.76	78.10	78.10
XGBoost	470.27	647.77	0.47	99.53	8737	49.34	79.142	64.74
XGBoost LSTM	624.12	784.2	0.70	99.3	8737	68.27	80.68	239.1

На основі отриманих результатів можна зробити висновок, що збільшення часової деталізації позитивно впливає на якість прогнозування для більшості моделей. Усі моделі навчалися на однаковій кількості записів, що забезпечує коректність порівняння отриманих показників.

Найменші значення похибок MAE, RMSE та MAPE продемонструвала модель CatBoost, що свідчить про її високу точність прогнозування на короткострокових часових інтервалах. Близькі результати також отримано для моделі LightGBM, яка забезпечує порівняно високу фінальну оцінку при помірному

часі навчання. Модель XGBoost демонструє дещо гірші значення похибок, проте зберігає стабільну якість прогнозу та прийнятну швидкість навчання.

Нейронні моделі, зокрема 1D CNN, CNN-LSTM та GRU, характеризуються вищими значеннями похибок та значно більшим часом навчання. Це свідчить про те, що за даних умов експерименту їхні переваги у моделюванні складних нелінійних залежностей не реалізуються повною мірою. Водночас гібридна модель XGBoost-LSTM демонструє підвищену точність визначення напрямку зміни ціни, що може бути корисним для задач аналізу ринкових трендів.

Час навчання моделей суттєво відрізняється залежно від обраного підходу. Ансамблеві методи, зокрема CatBoost та LightGBM, забезпечують найкраще співвідношення між точністю прогнозування та обчислювальними витратами, тоді як нейронні та гібридні моделі потребують значно більших ресурсів.

Для прогнозування криптовалютних часових рядів з таймфреймом 15 хвилин найбільш доцільним є використання градієнтних бустингових моделей. Вони забезпечують високу точність прогнозу при відносно невеликому часі навчання, що робить їх ефективними для практичного використання в системах оперативного аналізу ринку.

4.1.4. Візуальний аналіз прогнозів

У ході аналізу кількісних результатів експериментів навчання моделей визначено, що для подальшого аналізу доцільно обрати обмежену кількість найбільш репрезентативних моделей. Для детального порівняння прогнозів обрано моделі CatBoost, LightGBM та XGBoost, які продемонстрували найкраще співвідношення між точністю прогнозування та обчислювальними витратами.

Окрім цього, до подальшого аналізу включено гібридну модель XGBoost+LSTM, яка показала підвищену точність визначення напрямку зміни ціни та високі значення інтегральної оцінки. Використання цієї моделі дозволяє оцінити доцільність поєднання ансамблевих методів із нейронними мережами для прогнозування криптовалютних часових рядів.

Візуальний аналіз прогнозів передбачає порівняння фактичних значень цін із прогнозованими значеннями для обраних моделей, що дозволяє оцінити їхню здатність відтворювати загальні тренди, локальні коливання та різкі зміни ринку. Отримані графіки слугуватимуть додатковим інструментом оцінювання якості моделей та підтвердженням результатів кількісного аналізу.

Переходимо до аналізу моделі CatBoost. На рисунку 4.1 наведено графік актуальної і прогнозованої ціни під час навчання моделі CatBoost на таймфреймі 1h.

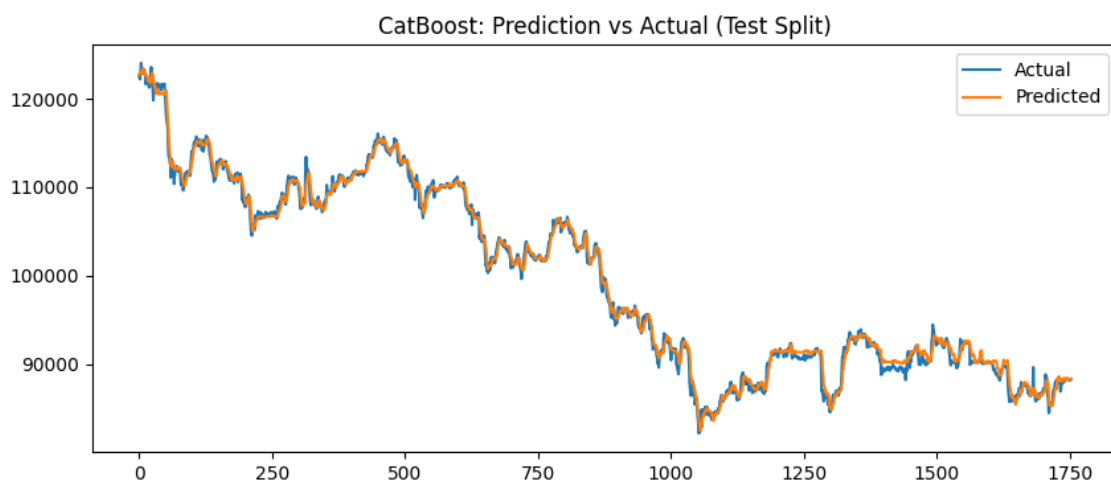


Рисунок 4.1 – Графік актуальної і прогнозованої ціни під час навчання моделі CatBoost на таймфреймі 1h

За результатами аналізу графіку прогнозів моделі CatBoost на таймфреймі 1h можна зробити висновок, що модель достатньо точно відтворює загальну динаміку ринку та основні трендові рухи ціни. Прогнозована крива майже повністю повторює форму фактичних значень, що свідчить про здатність моделі ефективно виявляти середньострокові закономірності у часовому ряді.

Найменші відхилення між актуальними та прогнозованими значеннями спостерігаються на відрізках із відносно плавною динамікою ціни. У періоди різких коливань та підвищеної волатильності модель дещо згладжує екстремальні значення, однак зберігає правильний напрямок руху, що підтверджується достатньо високими показниками точності напрямку.

Загалом отриманий графік підтверджує результати кількісного аналізу та демонструє, що модель CatBoost є ефективною для прогнозування криптовалютних часових рядів на таймфреймі 1 година, забезпечуючи баланс між точністю прогнозу та стабільністю результатів. На рисунку 4.2 наведено графік актуальної і прогнозованої ціни під час навчання моделі CatBoost на таймфреймі 15m.

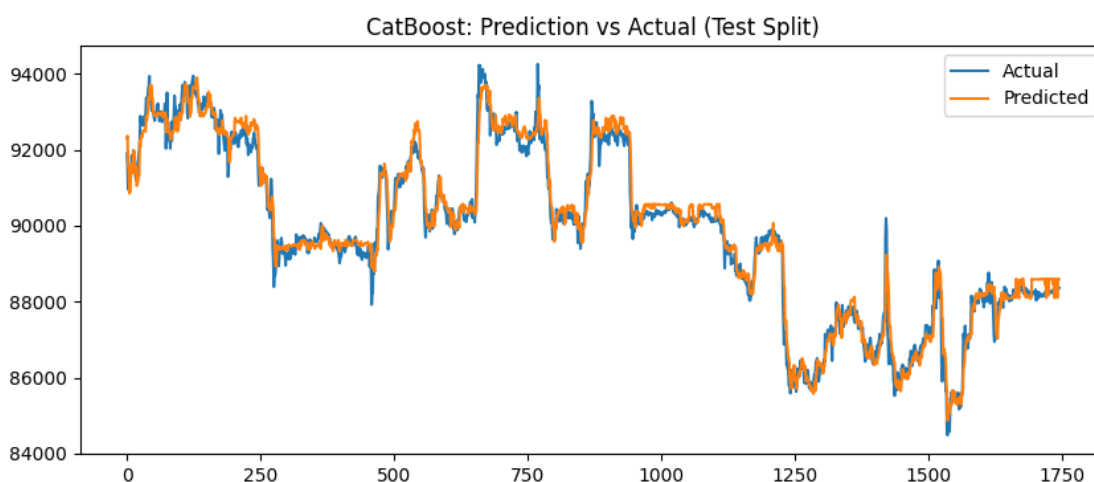


Рисунок 4.2 – Графік актуальної і прогнозованої ціни під час навчання моделі CatBoost на таймфреймі 15m

За результатами аналізу графіку прогнозів моделі CatBoost на таймфреймі 15 хвилин можна зробити висновок, що модель ефективно відтворює короткострокові коливання ціни та здатна реагувати на швидкі зміни ринкової динаміки. Прогнозована крива в більшості випадків щільно прилягає до фактичних значень, що свідчить про високу точність моделі на детальніших часових інтервалах.

На відміну від таймфрейму 1h, на інтервалі 15m більш чітко проявляються локальні коливання та короткочасні імпульси ціни. У цих умовах модель CatBoost демонструє здатність коректно відстежувати напрямок руху та рівні цін, хоча в окремих моментах спостерігається незначне згладжування різких піків і спадів.

Аналіз підтверджує, що використання таймфрейму 15 хвилин дозволяє підвищити чутливість моделі до короткострокових змін ринку, а модель CatBoost

залишається стабільною та точною навіть за умов підвищеної волатильності. Отримані результати узгоджуються з кількісними показниками якості та підтверджують доцільність застосування цієї моделі для оперативного аналізу криптовалютних ринків.

Переходимо до аналізу моделі LightGBM. На рисунку 4.3 наведено графік актуальної і прогнозованої ціни під час навчання моделі LightGBM на таймфреймі 1h.

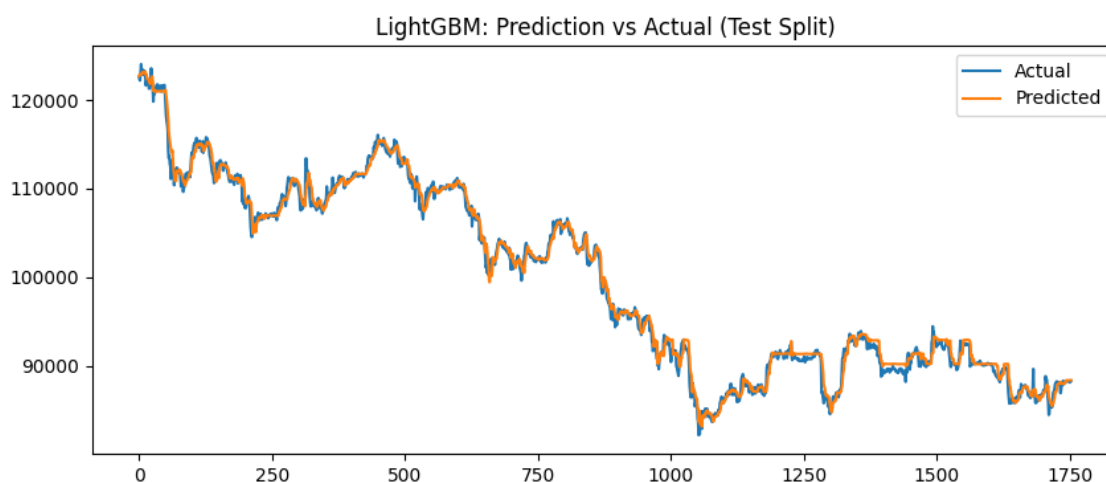


Рисунок 4.3 – Графік актуальної і прогнозованої ціни під час навчання моделі LightGBM на таймфреймі 1h

За результатами аналізу можна зробити висновок, що модель LightGBM достатньо точно відтворює загальний тренд зміни ціни криптовалюти на досліджуваному часовому інтервалі. Прогнозована крива у більшості випадків повторює форму фактичних значень, що свідчить про здатність моделі ефективно виявляти основні закономірності середньострокової динаміки ринку.

Найменші відхилення між актуальними та прогнозованими значеннями спостерігаються на відрізках із плавною зміною ціни. У періоди різких коливань та підвищеної волатильності модель LightGBM демонструє незначне згладжування пікових значень, однак при цьому зберігає правильний напрямок руху ціни. Це узгоджується з високими показниками симетричної середньої абсолютної відсоткової похибки та інтегральної оцінки, отриманими в кількісному аналізі. На

рисунку 4.4 наведено графік актуальної і прогнозованої ціни під час навчання моделі LightGBM на таймфреймі 15m.

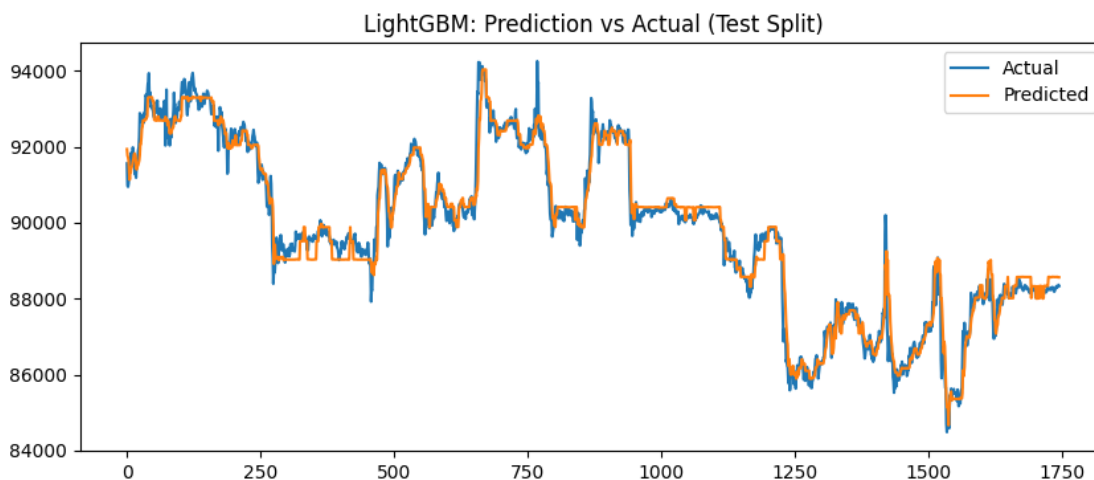


Рисунок 4.4 – Графік актуальної і прогнозованої ціни під час навчання моделі LightGBM на таймфреймі 15m

За результатами аналізу можна зробити висновок, що модель LightGBM добре адаптується до більш детальної часової роздільності та здатна ефективно відтворювати короткострокові коливання ринку. Прогнозована крива в більшості випадків тісно слідує за фактичними значеннями, що свідчить про високу точність моделі на даному таймфреймі.

Особливо добре модель відображає ділянки з помірною волатильністю та послідовними локальними трендами. У періоди різких стрибків ціни та високої волатильності спостерігається незначне згладжування екстремальних значень, однак загальний напрямок руху ціни зберігається коректно. Це підтверджує доцільність використання моделі LightGBM для задач короткострокового прогнозування.

Переходимо до аналізу моделі XGBoost. На рисунку 4.5 наведено графік актуальної і прогнозованої ціни під час навчання моделі XGBoost на таймфреймі 1h.

За результатами аналізу можна зробити висновок, що модель XGBoost демонструє високу здатність до відтворення загальної динаміки криптовалютного

ринку. Прогнозована крива майже повністю повторює форму фактичних значень, що свідчить про ефективне виявлення середньострокових трендів та основних закономірностей часових рядів.

Найменші відхилення між актуальними та прогнозованими значеннями спостерігаються на відрізках зі стабільною та помірно волатильною динамікою. У періоди різких змін ціни модель дещо згладжує пікові значення, однак зберігає правильний напрямок руху ринку, що узгоджується з високими показниками точності напрямку та низькими значеннями похибок, отриманими в кількісному аналізі.

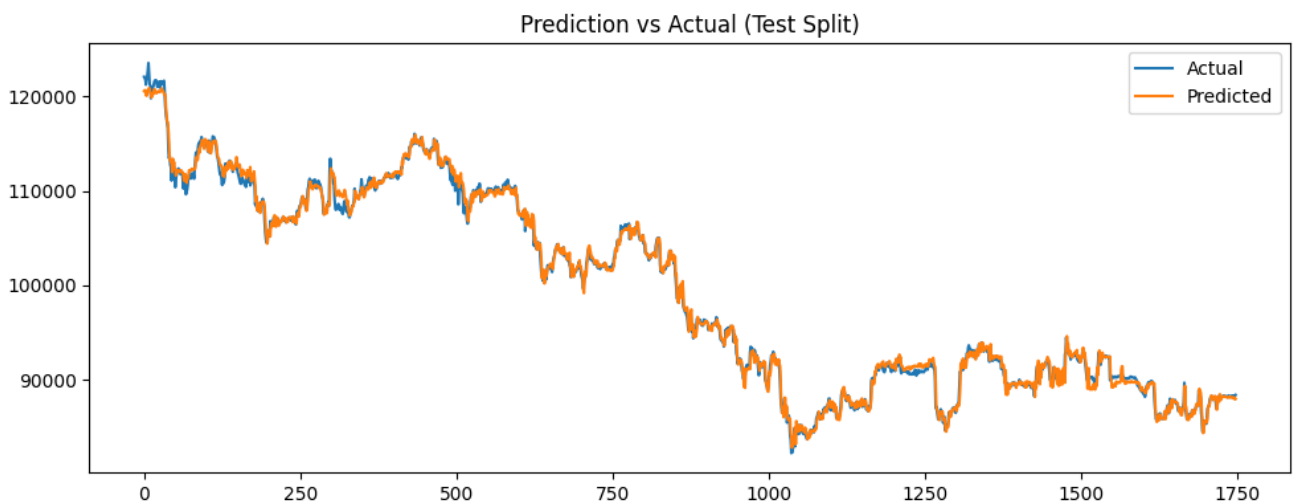


Рисунок 4.5 – Графік актуальної і прогнозованої ціни під час навчання моделі XGBoost на таймфреймі 1h

На рисунку 4.6 наведено графік актуальної і прогнозованої ціни під час навчання моделі XGBoost на таймфреймі 15m.

Аналіз графіка показує, що при переході до меншого таймфрейму модель XGBoost зберігає здатність точно відтворювати загальну динаміку цінового руху, однак прогноз стає більш чутливим до короткострокових коливань ринку. Прогнозована крива в більшості випадків добре узгоджується з фактичними значеннями, що підтверджує ефективність моделі для задач високочастотного прогнозування.

У періоди різких змін ціни та підвищеної волатильності спостерігаються локальні відхилення між прогнозованими та реальними значеннями, що є характерним для коротких таймфреймів. Водночас модель коректно визначає напрямок руху ціни та швидко адаптується до зміни ринкових умов, що підтверджується високими показниками точності напрямку та фінальної оцінки, отриманими під час кількісного аналізу.

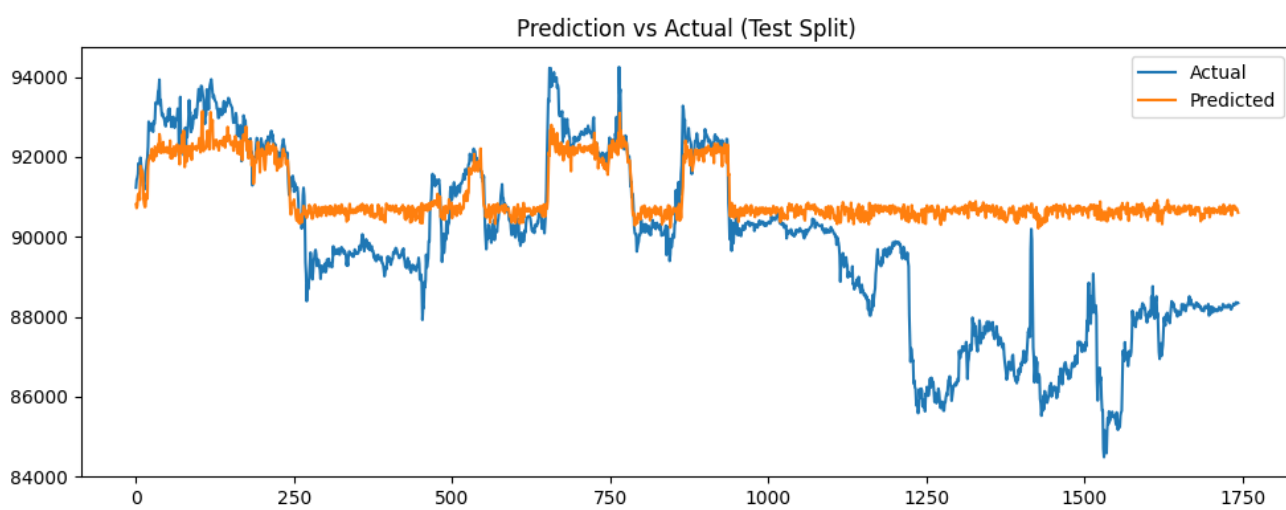


Рисунок 4.6 – Графік актуальної і прогнозованої ціни під час навчання моделі XGBoost на таймфреймі 15m

Переходимо до аналізу моделі XGBoost+LSTM. На рисунку 4.7 наведено графік актуальної і прогнозованої ціни під час навчання моделі XGBoost+LSTM на таймфреймі 1h.

З графіка видно, що модель поєднує переваги двох підходів: XGBoost ефективно відображає локальні коливання та короткострокові залежності, тоді як LSTM краще захоплює загальний тренд та довгострокову структуру часових рядів. У результаті ансамблевий прогноз демонструє більш згладжену поведінку та в більшості випадків знаходиться ближче до фактичних значень порівняно з окремими прогнозами базових моделей.

Особливо помітною є перевага ансамблю у періодах різких змін ринкової динаміки, де окремі моделі можуть давати запізнення або локальні похибки.

Комбінований прогноз зменшує амплітуду помилок та забезпечує більш стабільне відтворення напрямку руху ціни.

Візуальний аналіз підтверджує, що модель XGBoost+LSTM забезпечує підвищену точність та стійкість прогнозування на таймфреймі 1 година, що узгоджується з результатами кількісного аналізу та обґрунтовує доцільність використання ансамблевих підходів у задачах прогнозування криптовалютних часових рядів.

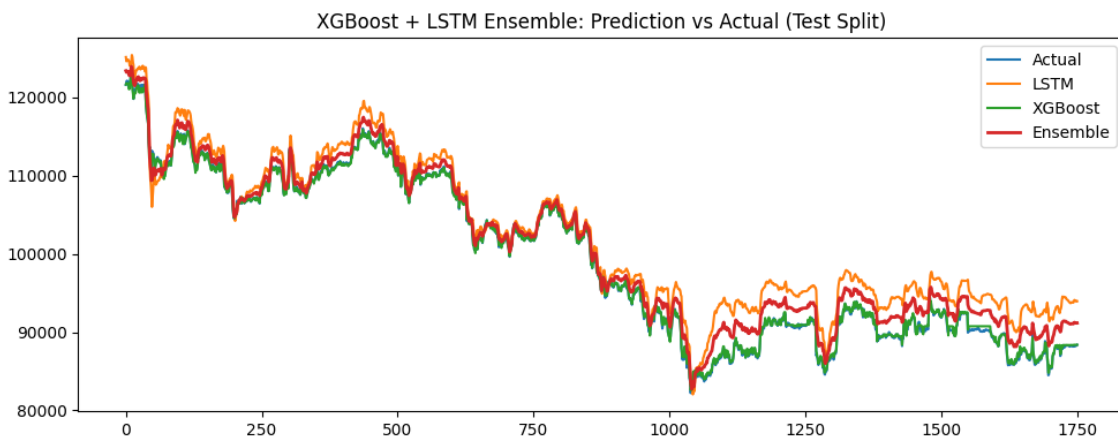


Рисунок 4.7 – Графік актуальної і прогнозованої ціни під час навчання моделі XGBoost+LSTM на таймфреймі 1h

На рисунку 4.7 наведено графік актуальної і прогнозованої ціни під час навчання моделі XGBoost+LSTM на таймфреймі 15m.

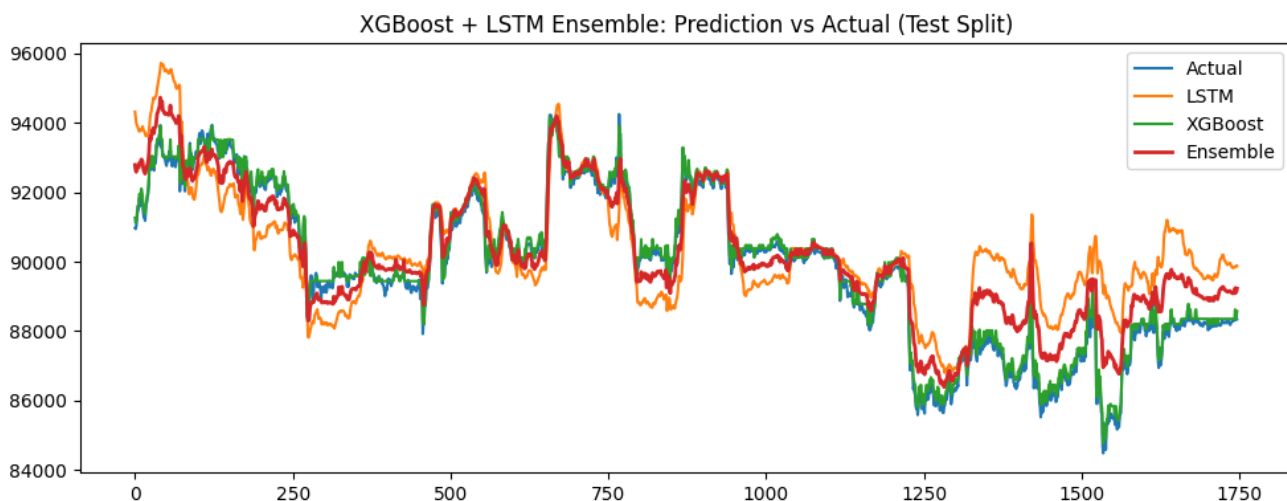


Рисунок 4.8 – Графік актуальної і прогнозованої ціни під час навчання моделі XGBoost+LSTM на таймфреймі 15m

З аналізу графіка видно, що при меншому таймфреймі модель стикається з більшою кількістю короткострокових коливань і ринкового шуму. Незважаючи на це, ансамблевий прогноз (лінія Ensemble) зберігає здатність коректно відтворювати загальний напрямок руху ціни та згладжувати локальні різкі зміни, характерні для окремих прогнозів базових моделей.

LSTM-компонента демонструє підвищену чутливість до локальних коливань, що призводить до деяких відхилень у пікових ділянках, тоді як XGBoost більш стабільно реагує на короткострокові зміни. Об'єднання цих підходів у межах ансамблю дозволяє зменшити вплив шуму та підвищити стабільність прогнозу порівняно з використанням кожної моделі окремо.

Отже, аналіз підтверджує, що модель XGBoost + LSTM на таймфреймі 15m забезпечує збалансований компроміс між чутливістю до швидких змін ринку та стійкістю прогнозування, що робить її доцільною для короткострокового аналізу криптовалютних часових рядів.

За результатами кількісного та візуального аналізу встановлено, що ансамблеві градієнтні моделі CatBoost, LightGBM та XGBoost демонструють найкраще співвідношення між точністю прогнозування та обчислювальними витратами. Завдяки високій швидкодії та здатності коректно відтворювати короткострокові коливання ринку ці моделі є найбільш доцільними для оперативного прогнозування на малих таймфреймах (15m–1h) та використання в режимі близькому до реального часу.

Водночас гібридна модель XGBoost + LSTM показала вищу стабільність і точність при прогнозуванні загального тренду та напрямку зміни ціни, особливо на більших часових інтервалах. Поєднання ансамблевих методів із рекурентною нейронною мережею дозволяє краще враховувати довгострокові залежності часових рядів і зменшувати вплив ринкового шуму, що робить цю модель більш придатною для довгострокового, але точнішого прогнозування.

ВИСНОВКИ

Під час виконання роботи проведено дослідження сучасних методів прогнозування на фінансових та криптовалютних ринках, що включало аналіз наявних моделей, методів обробки даних та інструментарію. Розроблено методологію підбору та підготовки даних, включно з їх фільтрацією, нормалізацією та формуванням наборів ознак для підвищення точності прогнозних моделей.

Розроблена програмна система забезпечує комплексну роботу з криптовалютними даними, включаючи збір, обробку, тестування моделей та генерацію прогнозів з виведенням відповідної статистики. Функціональна структура системи була описана за методологією IDEF0, що дозволило чітко визначити послідовність обробки даних та взаємодію компонентів. Реалізація включає три ключові ланки: фронтенд, Java бекенд для логістичного сервісу та моделі прогнозування на Python, що забезпечує гнучкість, масштабованість і інтеграцію з існуючими платформами.

Проведене дослідження показало, що інтеграція різних підходів і моделей дозволяє підвищити точність прогнозування динаміки цін криптовалют, а створена система може слугувати базою для подальшого розвитку інтелектуальних сервісів аналітики та підтримки прийняття рішень.

Результати роботи можуть бути використані у фінансових аналітичних платформах, торгових терміналах, навчальних та дослідницьких системах, а також у корпоративних рішеннях для управління ризиками. Розробка забезпечує можливість подальшого вдосконалення методів прогнозування, інтеграції додаткових джерел даних і реалізації нових інструментів аналізу для практичних і навчальних цілей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chen M., Mao S., Liu Y. Big data: A survey // *Mobile Networks and Applications*. – 2014. – Vol. 19, No. 2. – P. 171–209. – <https://doi.org/10.1007/s11036-013-0489-0>
2. Tian X., Zhao Y., Chen L., Wang S. Latency-critical big data computing in finance // *Journal of Big Data*. – 2015. – Vol. 2. – Article 14. – <https://doi.org/10.1186/s40537-015-0025-7>
3. Ahmed M. S., El-Masry A. A., Al-Maghyreh A. I., Kumar S. Cryptocurrency volatility: A review, synthesis, and research agenda // *Finance Research Letters*. – 2024. – Available online 5 July 2024. – <https://doi.org/10.1016/j.ribaf.2024.102472>
4. Laudon K. C., Laudon J. P. *Management Information Systems: Managing the Digital Firm*. 16th ed. Pearson Education, 2020. 768 p.
5. Sezer O. B., Gudelek M. U., Ozbayoglu A. M. (2020) *Financial time series forecasting with deep learning: A systematic literature review* // *Applied Soft Computing*. – 2020. – Vol. 90. – 106181. DOI: <https://doi.org/10.1016/j.asoc.2020.106181>
6. Katsiampa P. An empirical investigation of volatility dynamics in the cryptocurrency market // *Research in International Business and Finance*. 2019. Vol. 50. P. 322–335.
7. Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. *Time Series Analysis: Forecasting and Control*. 5th ed. Hoboken : Wiley, 2015. 712 p
8. Engle R. F. Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation // *Econometrica*. 1982. Vol. 50, No. 4. P. 987–1007.
9. Sezer O. B., Gudelek M. U., Ozbayoglu A. M. Financial time series forecasting with deep learning: A systematic literature review // *Applied Soft Computing*. – 2020. – Vol. 90. – 106181. – <https://doi.org/10.1016/j.asoc.2020.106181>

10. TradingView. *TradingView Charting Library Documentation*. 2024. – Режим доступу: <https://www.tradingview.com> (дата звернення: 06.12.2025).

11. Мацюцький В., Урняєва І. Підходи до збору та обробки даних для побудови моделей прогнозування на криптовалютних ринках. Collection of Scientific Papers with the Proceedings of the 2nd International Scientific and Practical Conference «Scientific Research: Emerging Theories and Practical Breakthroughs» (November 17-19, 2025, Edinburgh, Scotland). European Open Science Space, 2025. 123-126 p. URL: <https://www.eoss-conf.com/en/archive/scientific-research-emerging-theories-and-practical-breakthroughs-17-11-25/>

12. Omole, O., & Enke, D. (2024). *Deep learning for Bitcoin price direction prediction: Models and trading strategies empirically compared*. Financial Innovation, 10, Article 117. <https://doi.org/10.1186/s40854-024-00643-1>

13. Bouteska, A., Abedin, M. Z., Hajek, P., & Yuan, K. (2023). *Cryptocurrency price forecasting – A comparative analysis of ensemble learning and deep learning methods*. International Review of Financial Analysis, 89, 103055. <https://doi.org/10.1016/j.irfa.2023.103055>

14. Grebennik I., Reshetnik V., Ovezgeldyyev A., Ivanov V., Urniaieva I. (2019) Strategy of Effective Decision-Making in Planning and Elimination of Consequences of Emergency Situations In: Murayama Y., Velev D., Zlateva P. (eds) Information Technology in Disaster Risk Reduction. ITDRR 2018. IFIP Advances in Information and Communication Technology. Springer, Cham Scopus

15. Gautam M. Crypto price prediction using LSTM+XGBoost. arXiv:2506.22055 [Preprint]. 2025. URL: <https://arxiv.org/abs/2506.22055>.

16. Stonebraker M. The implementation of POSTGRES / Michael Stonebraker, Lawrence Rowe, Michael Hirohama. – 2nd ed. – [S. l.] : IEEE, 1990.

17. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. – 2016. – P. 785–794. – <https://doi.org/10.1145/2939672.2939785>

18. Zhang Z., Liu Y., Wu J. Financial forecasting using LightGBM with large-scale datasets // *Expert Systems with Applications*. – 2021. – Vol. 168. – <https://doi.org/10.1016/j.eswa.2020.114256>
19. Dorogush A. V., Ershov V., Gulin A. CatBoost for gradient boosting on decision trees // *arXiv preprint*. – 2018. – <https://arxiv.org/abs/1810.11363>
20. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions // *European Journal of Operational Research*. – 2018. – Vol. 270, No. 2. – P. 654–669. – <https://doi.org/10.1016/j.ejor.2017.11.054>
21. Sezer O. B., Gudelek M. U., Ozbayoglu A. M. Financial time series forecasting with deep learning: A systematic literature review // *Applied Soft Computing*. – 2020. – Vol. 90. – <https://doi.org/10.1016/j.asoc.2020.106181>
22. Mak G. *Spring Recipes: A Problem-Solution Approach* / Gary Mak, Daniel Rubio, Josh Long. – 2nd ed. – [S. l.] : Apress, 2010. – 1148 p.
23. Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation // *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, 2014. P. 1724–1734.
24. Leitch G., Tanner J. E. Economic forecast evaluation: profits versus the conventional error measures // *American Economic Review*. 1991. Vol. 81, No. 3. P. 580–590.