**НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет інформаційних технологій**

| | |
|---|---|
| **ПОГОДЖЕНО** | **ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ** |
| **Декан факультету інформаційних технологій** | **Завідувач кафедри економічної кібернетики** |
| _____ **Ігор БОЛБОТ** | _____ **Наталя РОГОЗА** |
| (підпис) | (підпис) |
| "___" _____ 2025 р. | "___" _____ 2025 р. |

# МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

## на тему: «Моделі та методи прогнозування курсу криптовалют»

Спеціальність: 051 «Економіка»

Освітня програма: «Економічна кібернетика»

Орієнтація освітньої програми: освітньо-професійна

**Гарант освітньої програми**

к.е.н., доцент _____ **Людмила ГАЛАЄВА**
(підпис)

**Керівник магістерської кваліфікаційної роботи**

д.е.н., професор _____ **Володимир КРАВЧЕНКО**
(підпис)

**Виконав** _____ **Денис МАРИНЕНКО**
(підпис)

**КИЇВ – 2025**

**НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

**Факультет інформаційних технологій**

**ЗАТВЕРДЖУЮ**
**Завідувач кафедри**
**економічної кібернетики**
к.е.н., доцент _____ **Наталя РОГОЗА**
"____" _____ 2025 року

# З А В Д А Н Н Я

# ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Мариненку Денису Вікторовичу

Спеціальність: 051 «Економіка»

Освітня програма: «Економічна кібернетика»

Орієнтація освітньої програми: освітньо-професійна

Тема магістерської кваліфікаційної роботи: «**Моделі та методи прогнозування курсу криптовалют»**

затверджена наказом від 01.11.2024 р. №1967 «С»

Термін подання завершеної роботи на кафедру: 14.11.2025 р.

Вихідні дані до магістерської кваліфікаційної роботи: дослідження ринку криптовалют, матеріали кафедри

Перелік питань, що підлягають дослідженню:
1. Проаналізувати сучасний стан і тенденції розвитку криптовалютного ринку.
2. Дослідити існуючі моделі та методи прогнозування фінансових часових рядів.
3. Реалізувати програмне забезпечення для проведення експериментів і перевірки моделі

Перелік графічного матеріалу: презентація результатів роботи

**Дата видачі завдання** 15.11.2025 р.

**Керівник магістерської кваліфікаційної роботи** _____ **Володимир КРАВЧЕНКО**
( підпис )

**Завдання прийняв до виконання** _____ **Денис МАРИНЕНКО**
( підпис )

# РЕФЕРАТ

Магістерська кваліфікаційна робота: 55 с., 2 табл., 13 рис., 3 додатки, 36 джерел.

Метою магістерської роботи є розроблення та дослідження моделей і методів прогнозування курсу криптовалют на основі сучасних підходів машинного та глибинного навчання з метою підвищення точності прогнозів і забезпечення аналітичної підтримки прийняття рішень на криптовалютному ринку.

Об’єктом дослідження є процеси прогнозування часових рядів вартості криптовалют на основі історичних та ринкових даних.

Предметом дослідження є моделі, методи та алгоритми прогнозування курсу криптовалют, що базуються на використанні статистичних, машинних і глибинних підходів.

У роботі використовуються методи статистичного аналізу, економетрики, машинного та глибинного навчання, аналізу часових рядів, а також експериментальні методи дослідження моделей прогнозування з використанням реальних ринкових даних.

Розділ 1 «Теоретичні основи прогнозування курсів криптовалют» присвячено аналізу наукових і практичних підходів до прогнозування фінансових часових рядів. Розглянуто сутність криптовалют, їх роль у цифровій економіці та основні фактори впливу на курс (ринкові, технічні, інформаційні, психологічні). Проаналізовано методи прогнозування: від класичних статистичних моделей (ARIMA, VAR) до сучасних алгоритмів машинного та глибинного навчання (Random Forest, LSTM, GRU). Окрему увагу приділено огляду наукових публікацій і порівнянню підходів до прогнозування криптовалют.

У розділі 2 «Розроблення моделі прогнозування курсу криптовалют» сформульовано постановку задачі прогнозування та описано підхід до побудови моделі. Пояснюється вибір методів, описано процес збирання та попередньої обробки даних (очищення, нормалізація, усунення пропусків, підготовка

навчальної вибірки). Розглядається архітектура запропонованої моделі машинного або глибинного навчання (наприклад, LSTM-мережа з кількома шарами). Показано реалізацію алгоритму за допомогою інструментів Python.

У розділі 3 «Експериментальні дослідження та аналіз результатів» подано опис проведених експериментів із використанням реальних даних. Порівнюються результати різних моделей (ARIMA, LSTM, GRU, Hybrid). Для оцінки якості прогнозів застосовано метрики точності: MAE, RMSE, MAPE. Розділ 4 «Практична реалізація моделі» присвячено створенню прикладного програмного рішення для прогнозування курсу криптовалют. Описано архітектуру системи, принципи взаємодії з API криптобірж (CoinGecko, Binance) та реалізацію інтерфейсу користувача. Демонструється робота системи, приклади візуалізації результатів і можливості подальшого розширення (підтримка кількох валют, інтеграція з аналітичними платформами). Подано результати тестування продуктивності та точності, а також рекомендації щодо використання програми у практичній діяльності.

Розроблено програмне забезпечення для автоматизованого прогнозування курсу криптовалют, яке включає модулі збору даних із біржових API (Binance, CoinGecko), попередньої обробки даних, навчання моделей, побудови прогнозів і візуалізації результатів. Реалізовано систему кешування, обробки rate-limit, підтримку Docker-контейнеризації та веб-інтерфейс користувача.

Результати роботи можуть бути використані для створення інтелектуальних фінансових систем, розширення торговельних платформ, оптимізації алгоритмічної торгівлі та подальших досліджень у галузі фінансової аналітики.

Ключові слова: криптовалюта; прогнозування; машинне навчання; глибинне навчання; нейронні мережі; Bitcoin; алгоритмічний трейдинг; фінансове моделювання.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

**API (Application Programming Interface)** – прикладний програмний інтерфейс для отримання даних із зовнішніх сервісів.

**ARIMA (Autoregressive Integrated Moving Average)** – авторегресійна інтегрована модель середнього ковзання.

**BTC** – Bitcoin, перша і найпоширеніша криптовалюта.

**CNN (Convolutional Neural Network)** – згорткова нейронна мережа.

**ETH** – Ethereum, криптовалюта другого покоління з підтримкою смарт-контрактів.

**GRU (Gated Recurrent Unit)** – рекурентна нейронна мережа зі структурою «керованого блоку».

**LSTM (Long Short-Term Memory)** – нейронна мережа довгої короткочасної пам’яті.

**MAE (Mean Absolute Error)** – середня абсолютна похибка.

**MAPE (Mean Absolute Percentage Error)** – середня абсолютна відносна похибка.

**ML (Machine Learning)** – машинне навчання.

**MSE (Mean Squared Error)** – середньоквадратична функція втрат.

**OHLC (Open, High, Low, Close)** – формат біржових даних: відкриття, максимум, мінімум, закриття.

**RMSE (Root Mean Squared Error)** – корінь середньоквадратичної похибки.

**RNN (Recurrent Neural Network)** – рекурентна нейронна мережа.

**SOL** – Solana, високошвидкісна блокчейн-платформа та криптовалюта.

**SQL (Structured Query Language)** – мова структурованих запитів для роботи з базами даних.

**CSV (Comma-Separated Values)** – формат табличних даних із розділенням комами.

**UI (User Interface)** – користувацький інтерфейс.

**VAE (Validation Absolute Error)** – абсолютна похибка на валідаційному наборі.

# ВСТУП

Стрімкий розвиток цифрових технологій і поширення блокчейн-рішень призвели до формування нового фінансового середовища – ринку криптовалют. Криптовалюти стали не лише інструментом обміну та збереження вартості, але й об’єктом інвестиційної діяльності, що характеризується високою волатильністю та непередбачуваністю. Саме тому питання прогнозування курсу криптовалют набуло особливої актуальності як у науковій, так і у прикладній площині.

В умовах швидких змін ринкових тенденцій традиційні економетричні методи не завжди забезпечують належний рівень точності прогнозування. Натомість сучасні технології машинного та глибинного навчання дозволяють враховувати складні нелінійні залежності між факторами та забезпечують гнучкість моделей у динамічних умовах. Застосування таких підходів відкриває нові можливості для побудови інтелектуальних систем прогнозування фінансових часових рядів, зокрема курсів криптовалют.

Актуальність теми зумовлена потребою у створенні ефективних інструментів прогнозування, що дозволяють підвищити точність оцінки ринкових тенденцій та мінімізувати ризики при прийнятті інвестиційних рішень.

Метою магістерської роботи є розроблення та дослідження моделей і методів прогнозування курсу криптовалют на основі сучасних підходів машинного та глибинного навчання з метою підвищення точності прогнозів і забезпечення аналітичної підтримки прийняття рішень на криптовалютному ринку.

Для досягнення поставленої мети необхідно вирішити такі завдання:

Проаналізувати сучасний стан і тенденції розвитку криптовалютного ринку.

Дослідити існуючі моделі та методи прогнозування фінансових часових рядів.

Визначити основні фактори, що впливають на динаміку курсу криптовалют.

Реалізувати програмне забезпечення для проведення експериментів і перевірки моделі.

Провести експериментальні дослідження та оцінити точність прогнозування за допомогою стандартних метрик.

Порівняти результати з існуючими підходами та сформулювати практичні рекомендації.

Об’єктом дослідження є процеси прогнозування часових рядів вартості криптовалют на основі історичних та ринкових даних.

Предметом дослідження є моделі, методи та алгоритми прогнозування курсу криптовалют, що базуються на використанні статистичних, машинних і глибинних підходів.

У роботі використовуються методи статистичного аналізу, економетрики, машинного та глибинного навчання, аналізу часових рядів, а також експериментальні методи дослідження моделей прогнозування з використанням реальних ринкових даних.

Наукова новизна магістерської роботи полягає у розробленні комбінованої моделі прогнозування курсу криптовалют із використанням LSTM, GRU та ARIMA, удосконаленні підходів до підготовки даних шляхом оптимізованої нормалізації та формування лагових ознак, а також у запропонованому адаптивному методі вибору довжини часових вікон. Створено програмну архітектуру з автоматизованою інтеграцією даних із Binance та CoinGecko, включно з механізмами кешування та контролю частоти запитів. Експериментально підтверджено перевагу LSTM над класичними статистичними моделями за точністю прогнозування, а також сформовано узагальнену методику оцінювання ефективності моделей для задач фінансового прогнозування.

# РОЗДІЛ 1
# ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ КУРСІВ КРИПТОВАЛЮТ

## 1.1. Сутність та економічна природа криптовалют

Криптовалюта – це форма цифрової або віртуальної валюти, яка використовує криптографічні технології для забезпечення безпеки транзакцій. Термін складається з двох компонентів: "крипто" відноситься до криптографії, яка забезпечує безпеку інформації користувачів та транзакцій, а "валюта" – це просто засіб обміну [1].

Основними характеристиками криптовалют є:

- Децентралізована природа: Криптовалюти функціонують через комп'ютерну мережу без залежності від центрального органу влади, такого як уряд чи банк. Завдяки децентралізованій природі емісії, криптовалюти не втілюють жодних вимог до держави, кредитної установи чи іншого емітента, і тому, з точки зору банківських регуляторів, не є абсолютно безпечними для використання як засіб обміну [1].
- Використання технології блокчейн: В основі криптовалют лежить технологія блокчейн – розподілений цифровий реєстр, який безпечно зберігає записи в мережі комп'ютерів у прозорий, незмінний спосіб та стійкий до втручання. Блокчейн – це відкрита база даних, яка формує постійний запис транзакцій між сторонами. Кожна транзакція представляє "блок" даних, а ці блоки формують "ланцюг", який неможливо змінити чи підробити [2].
- Економічні функції: З економічної точки зору, важлива відмінність між криптовалютами полягає в їх призначенні. Деякі криптовалюти, через технологічні особливості реалізації та функціональні характеристики, найбільш підходять для використання як засіб заощадження та/або резервний актив (Bitcoin), інші – як засіб платежу (наприклад, Litecoin і Bitcoin Cash), а треті – як інструмент/платформа для реалізації децентралізованих додатків на основі смарт-контрактів (наприклад, Ethereum, Cardano, EOS та TRON).

Криптовалюта не є формою грошей у традиційному розумінні. Щоб вважатися грошима, актив повинен відповідати ключовим характеристикам:

- Широковживаний засіб платежу: Хоча криптовалюти можуть використовуватися для купівлі та продажу товарів, вони не є широковживаними засобами платежу, і опитування показують, що лише невелика частка власників криптовалют регулярно використовує їх для платежів [3].

- Засіб збереження вартості: Великі коливання цін багатьох криптовалют означають, що їхня купівельна спроможність не зберігається з часом, що зменшує їхню ефективність як засобу збереження вартості.

- Одиниця обліку: Криптовалюти не є загальноприйнятим способом вимірювання вартості товарів та послуг. У більшості країн ціни вимірюються у фіатній валюті.

За даними на червень 2023 року, існувало понад 25 000 інших криптовалют на ринку, з яких понад 40 мали ринкову капіталізацію, що перевищує 1 мільярд доларів. Станом на квітень 2025 року, капіталізація ринку криптовалют вже оцінювалася у 2,76 трільйона доларів. У 2025 році ринкова капіталізація криптовалют досягла значної віхи, перевищивши 4 трільйони доларів, що відображає 20% зростання користувачів мобільних гаманців порівняно з попереднім роком [4].

### 1.2. Огляд сучасних тенденцій розвитку криптовалютного ринку

Основними трендами на ринку криптовалют у 2025 році є:

- Інституційне впровадження: 2025 рік позначився підвищенням довіри інституційних інвесторів до криптовалют. За даними Kraken, 92% власників криптовалют у США оптимістично налаштовані щодо потенціалу блокчейну для модернізації економіки США. Ключовими факторами, що сприяють цьому зростанню, є регуляторна ясність, збільшення інституційного впровадження та покращення інфраструктури [5].

- Регуляторні зміни: Історичні схвалення кількох спот Bitcoin ETF у США стали поворотним моментом для ринку. Після отримання схвалення від SEC у

2024 році, Bitcoin спот ETF побачили приплив у 36,4 мільярда доларів з моменту початку торгівлі у січні, тоді як Ethereum спот ETF залучили 2,4 мільярди доларів з липня. Bitcoin ETF від BlackRock зріс до 70 мільярдів доларів притоку швидше, ніж будь-який інший ETF на ринку [6].

- Стейблкойни: Стейблкойни стали критично важливим компонентом криптоекосистеми. Загальна ринкова капіталізація стейблкойнів подвоїлася з 120 мільярдів доларів 18 місяців тому до 250 мільярдів доларів сьогодні, і прогнозується, що вона досягне понад 400 мільярдів доларів до кінця року та 2 трільйонів доларів до 2028 року. Комбінований торговий обсяг USDT та USDC – двох найбільших стейблкойнів за ринковою капіталізацією – досяг 23 трільйонів доларів у 2024 році, що на 90% більше порівняно з попереднім роком [7].

- DeFi та токенізація: Децентралізовані фінанси (DeFi) продовжують розвиватися, хоча з деякими викликами. Приблизно 50 мільярдів доларів було втрачено з ринків DeFi у першому кварталі 2025 року, що означає зниження на 27% з кінця 2024 року. Водночас, токенізація революціонізує такі галузі, як нерухомість та мистецтво, дозволяючи дрібне володіння та покращуючи ліквідність для традиційно неліквідних активів [5]

- AI та криптовалюти: Штучний інтелект (AI) трансформує криптоландшафт, при цьому AI-керовані технології відкривають більшу ефективність та можливості для інновацій. Інтеграція AI з криптовалютним аналізом дозволяє краще прогнозувати та автоматизувати торгові стратегії [5].

У 2025 році прогнозується, що Bitcoin торгуватиметься між 80 440 і 151 200 доларами, з розширеною цільовою ціною від 175 000 до 185 000 доларів. Станом на 26 жовтня 2025 року, ціна Bitcoin становила 114 472,44 долара з ринковою капіталізацією 2,28 трільйона доларів [8].

Протягом 2023 року спостерігалося зростання ринкової капіталізації Bitcoin та зниження реалізованої волатильності. Реалізована волатильність нижче 50% спостерігалася лише у 5% історії Bitcoin. У 2024 році при ціні 60 000 доларів Bitcoin був майже вдвічі менш волатильним порівняно з 2021 роком [9].

Експерти передбачають продовження зростаючого ринку криптовалют протягом 2025 року. Очікується зростання цін Bitcoin та Ethereum, тоді як інші визначні проекти також набиратимуть обертів. Зростання може призупинитися влітку і може призвести до різкіших спадів, хоча з можливостями відновлення восени [10].

Венчурні інвестиції в криптостартапи досягли 4,9 мільярда доларів у першому кварталі 2025 року – найвищого показника за понад 2 роки. Серед 445 інших угод інвестиції зосереджувалися на криптостартапах ранньої стадії. Загальне венчурне фінансування у крипто цього року прогнозується перевищити 18 мільярдів доларів [11].

### 1.3. Фактори, що впливають на курс криптовалют

Ціни криптовалют визначаються складною взаємодією численних факторів, які можна класифікувати на економічні, технічні, соціальні та інформаційні [12].

Економічні фактори:

1) Попит та пропозиція: Як і всі товари та активи, коли пропозиція криптовалюти перевищує попит, ціни падають; коли попит перевищує пропозицію, ціни зростають. Багато криптовалют мають обмежену пропозицію – наприклад, Bitcoin має граничну пропозицію в 21 мільйон монет, що збільшує дефіцит з часом [13].

2) Токеноміка: Пропозиція криптовалюти часто визначається на початку створення валюти, що відомо як токеноміка, яка включає такі фактори, як загальна пропозиція, методи розподілу, механізми консенсусу та протоколи спалювання. Наприклад, Ethereum спочатку не мав обмеження пропозиції та випускав близько 18 мільйонів нових ETH щорічно. Перехід механізму випуску з Proof of Work (PoW) на Proof of Stake (PoS) суттєво зменшив новий випуск, а впровадження механізму спалювання також допомагає підтримувати довгострокову стабільність цін [13].

3) Інфляція та відсоткові ставки: Коли інфляція та відсоткові ставки зростають, інвестори прагнуть захистити свої інвестиції в інших місцях, подалі від продуктів, на які безпосередньо впливають ці фактори. Криптовалюти пропонують альтернативну форму торгівлі активами та інвестицій, тому деякі монети можуть побачити зростання своїх цін у такій ситуації [14].

4) Макроекономічні тенденції: Ширші економічні тенденції впливають на ціни криптовалют. Такі фактори, як інфляція, відсоткові ставки та геополітичні події, можуть впливати на настрої інвесторів та спрямовувати капітал в криптовалюти або виводити його з них. Економічна нестабільність або невизначеність можуть змусити інвесторів шукати альтернативні активи, такі як криптовалюти, як захист проти традиційних ринків [15].

5) Ліквідність: Ліквідність відноситься до того, наскільки легко актив можна купити або продати без значного впливу на його ціну. Висока ліквідність означає стабільні ціни, тоді як низька ліквідність означає більшу волатильність. Менші криптовалюти з нижчими торговими обсягами, як правило, зазнають різкіших коливань цін порівняно з добре встановленими монетами, такими як Bitcoin або Ethereum [12]

6) Халвінг Bitcoin: Халвінг Bitcoin – це подія, коли винагорода за майнінг нових блоків зменшується вдвічі, що відбувається приблизно кожні чотири роки. Історично, халвінг події корелювали з бичачими трендами в ціні Bitcoin, оскільки знижений темп створення нових Bitcoin посилює їх дефіцитність. Наприклад, приблизно через рік після першого халвінгу частка Bitcoin, утримуваного довгостроковими інвесторами (понад 3 роки), зросла приблизно на 73% [16].

Технічні фактори:

1) Регуляторні зміни: Урядові регулювання відіграють значну роль у формуванні довіри інвесторів. Оголошення про юридичні обмеження або заборони можуть призвести до раптових розпродажів, тоді як сприятливі регулювання можуть заохочувати інвестиції та підвищувати ціни. Оскільки регулювання різняться за країнами та все ще розвиваються, невизначеність сприяє триваючій волатильності рухів цін криптовалют [12].

2) Технологічні зміни та оновлення мережі: Великі оновлення блокчейн-мереж – відомі як форки або оновлення – можуть впливати на ціни. Успішне оновлення може покращити масштабованість або безпеку, підвищуючи довіру інвесторів. Невдалі оновлення, що призводять до хард-форків (як Bitcoin проти Bitcoin Cash), можуть створювати невизначеність та розділяти спільноти [12].

3) Безпека мережі: Хешрейт Bitcoin – загальна обчислювальна потужність, що використовується для майнінгу – є індикатором безпеки та здоров'я мережі. Вищий хешрейт означає більш безпечну мережу та може позитивно впливати на ціну [17].

4) Масштабованість та швидкість транзакцій: Технологічні вдосконалення, які покращують швидкість транзакцій та масштабованість блокчейну, можуть збільшити прийняття та впливати на ціни. Наприклад, рішення Layer-2 для Ethereum спрямовані на зниження комісій та покращення швидкості транзакцій [18].

До соціальних та інформаційних факторів, що впливають на курс криптовалют можна віднести:

1) Ринкові настрої: Ринкові настрої відносяться до загального настрою або ставлення інвесторів до конкретної криптовалюти або ринку загалом. Позитивні новини (наприклад, інституційне впровадження або технологічні оновлення) можуть підвищувати ціни. Негативні новини (наприклад, злами бірж або регуляторні придушення) можуть викликати панічний продаж. Соціальні медіа-платформи, такі як Twitter та Reddit, часто посилюють рухи, керовані настроями, швидко поширюючи інформацію – чи то точну, чи ні [12].

2) Страх та жадібність: Вартість будь-якої криптовалюти є спекулятивною, що означає, що люди купують (або продають її) на основі своїх власних думок та почуттів – зокрема, страху та жадібності. Існує навіть індекс страху та жадібності, який допомагає визначити поточні ринкові настрої [19].

3) Новинний фон: Іноді велика новинна подія може швидко змінити ціну Bitcoin і відповідно вплинути на весь ринок [15].

4) Активність великих гравців (китів): Крипто "кити" – це особи або установи, які володіють значною кількістю певної криптовалюти (наприклад,

володіння 1000 BTC або більше класифікує когось як кита Bitcoin). Їхні великі холдинги дозволяють їм значно впливати на ринкову динаміку, ціни токенів та навіть стабільність торговельних платформ. Кити використовують різні тактики для маніпулювання криптовалютним ринком, включаючи схеми pump-and-dump, спуфінг, воштрейдинг та створення стін продажу [20].

5) Кореляція з Bitcoin: Для більшості криптовалют важливим фактором впливу на ціну є її кореляція з котируваннями Bitcoin. Практика показує, що коли основний цифровий актив дорожчає або дешевшає, це впливає на курси всіх альткойнів. Згідно з дослідженням, проведеним Binance на початку року, ціни Ethereum, Litecoin, EOS, XRP, Bitcoin Cash та BNB у 70% випадків повторювали рух котирувань Bitcoin [15].

6) Спекуляції та левередж: Багато трейдерів використовують спекуляції – купівлю на основі очікуваних майбутніх змін цін – щоб отримати прибуток від короткострокових рухів. Левередж дозволяє трейдерам брати в борг кошти для збільшення розміру своєї позиції. Хоча це потенційно збільшує прибутки, це також сприяє швидким ринковим коливанням, коли позиції ліквідуються масово [12].

7) Вплив торгового обсягу: Торговий обсяг відіграє ключову роль у визначенні здоров'я та життєздатності криптовалюти. Вищі обсяги зазвичай свідчать про те, що криптовалюта активно торгується та має достатню ліквідність для полегшення транзакцій. Ліквідність є важливою для трейдерів, оскільки вона зменшує прослизання, забезпечуючи виконання ордерів на покупку та продаж за сприятливими цінами [21].

8) Лістинг на біржах: Майже очевидно, але для торгівлі певною криптовалютою вона повинна бути зареєстрована на біржі. Біржі вибирають, якими монетами торгувати, і лістинг на більших біржах може зробити величезну різницю в обсязі. Станом на березень 2025 року, Binance залишається домінуючою спот-біржею з понад 40% частки ринку [22]. У 2024 році топ-10 централізованих бірж (CEX) зареєстрували 17,4 трільйона доларів у спотовому торговому обсязі, що на 140% більше порівняно з 2023 роком. Для порівняння, торговий обсяг децентралізованих бірж (DEX) становить близько 10% від CEX.

### 1.4. Методи прогнозування фінансових часових рядів

Прогнозування цін криптовалют є складним завданням через високу волатильність та нелінійну динаміку ринку. Дослідники та трейдери використовують широкий спектр методів, від класичних статистичних моделей до передових методів глибинного навчання.

**Класичні статистичні методи:**

1) Авторегресійне інтегроване ковзне середнє – ARIMA (Autoregressive Integrated Moving Average) – це одна з найбільш широко використовуваних техник для прогнозування часових рядів. ARIMA означає авторегресійну інтегровану ковзну середню і інтегрує два різні підходи до прогнозування даних часових рядів [23].

Компоненти ARIMA:

Авторегресійна (AR) частина: Показує, що змінна, що розвивається, регресує на свої попередні значення. AR-компонент будує тренд з минулих значень.

Інтегрована (I) частина: Передбачає диференціювання часового ряду, щоб зробити його стаціонарним, що означає, що середнє та дисперсія повинні залишатися постійними протягом певного періоду часу. Диференціювання допомагає моделі підлаштовуватися під дані, а не під шум.

Ковзна середня (MA) частина: Зосереджується на відношенні між спостереженням та залишковою помилкою. MA-компонент оцінює вплив минулих помилок на поточне спостереження.

2) Сезонне авторегресійне інтегроване ковзне середнє SARIMA (Seasonal ARIMA) додає сезонність до ARIMA з сезонними параметрами (P, D, Q) на додаток до несезонних параметрів (p, d, q). Якщо сезонність присутня у вашому часовому ряді, її використання в прогнозі є критичним. Позначення SARIMA: SARIMA(p, d, q)(P, D, Q)m, де m визначає сезонний період (наприклад, 12 для місячних даних або 4 для квартальних даних).

Переваги: SARIMA значно потужніша за ARIMA при прогнозуванні складних просторів даних, що містять цикли. Модель може захоплювати як

несезонні, так і сезонні патерни у ваших даних і будувати модель прогнозування [24].

3) Векторна авторегресія (VAR, Vector Autoregression) – це узагальнення одновимірної авторегресійної моделі для багатовимірних часових рядів. Модель VAR є одним з найуспішніших моделей для аналізу багатовимірних часових рядів. Вона продемонструвала успіх у описі взаємозв'язків та прогнозуванні економічних і фінансових часових рядів, забезпечуючи більш точні прогнози, ніж одновимірні моделі часових рядів та теоретично обґрунтовані моделі одночасних рівнянь [25].

VAR розглядає всі змінні як ендогенні та дозволяє кожній змінній залежати від p лагових значень себе та всіх інших змінних у системі. Модель VAR порядку p можна представити як систему регресій, де вектор yt (що містить N ендогенних змінних) залежить від своїх власних лагових значень та лагових значень інших змінних.

VAR слід використовувати, коли ціль прогнозу залежить від кількох інших змінних, що змінюються в часі, що призводить до багатовимірного часового ряду, і коли взаємодії між ціллю та іншими змінними є простими та лінійними.

**Методи машинного навчання** показали багатообіцяючі результати у прогнозуванні цін криптовалют, демонструючи кращу точність прогнозування порівняно з традиційними параметричними регресійними підходами [26].

**Регресійні моделі**

1) Лінійна регресія: Проста, але ефективна техніка для моделювання взаємозв'язку між залежною змінною (ціною) та однією або кількома незалежними змінними (технічними індикаторами, обсягом тощо). Абрахам та ін. представили метод прогнозування змін цін Bitcoin та Ethereum з використанням даних Twitter та даних Google Trends, використовуючи лінійну модель [27].

2) Метод опорних векторів – Support Vector Machine (SVM): SVM є потужним алгоритмом машинного навчання, який може використовуватися як для класифікації, так і для регресії. SVM працює, знаходячи оптимальну

гіперплощину, яка максимізує межу між різними класами. У прогнозуванні цін криптовалют SVM досяг точності 82% для Bitcoin [28].

**Random Forest** – це метод ансамблевого навчання, який будує кілька дерев рішень під час навчання та виводить клас, що є режимом класів (класифікація) або середнім прогнозом (регресія) окремих дерев. Random Forest виявив майже ідеальну точність у прогнозуванні цін Bitcoin, випереджаючи багато інших моделей [29].

Random Forest менш схильний до перенавчання порівняно з одиночними деревами рішень і може обробляти великі набори даних з вищою розмірністю. Для довгострокових прогнозів цін Bitcoin Random Forest (RF), Gradient Boosting

**Методи глибинного навчання**

Архітектури глибинного навчання – такі як RNN, LSTM та GRU – виділяються як особливо придатні для захоплення часової складності та нелінійності часових рядів криптовалют.

LSTM (Long Short-Term Memory)

LSTM – це тип рекуррентної нейронної мережі (RNN), спеціально розроблений для вирішення проблеми зникаючого градієнта, яка заважає стандартним RNN вивчати довгострокові залежності [30].

LSTM використовує структуру з воротами (вхідні ворота, вихідні ворота та ворота забування), які контролюють потік інформації. Це дозволяє LSTM зберігати інформацію протягом тривалих періодів часу та вибірково забувати нерелевантну інформацію.

LSTM-мережі перевершили інші моделі з 88% точністю для Bitcoin завдяки їхній здатності зберігати часові патерни, які є критичними для прогнозів на крипторинку. Дослідження показало, що запропонована модель на основі RNN-LSTM перевершує інші встановлені методи, про що свідчать нижчі помилки (MSE, MAE, MAPE) та підвищена точність (R-квадрат). Для Ethereum LSTM досяг RMSE 0,01083, MSE 0,00011, $R^2$ 0,80618 [31].

GRU (Gated Recurrent Unit)

GRU – це варіант LSTM, який має більш просту архітектуру з лише двома воротами (ворота оновлення та ворота скидання). GRU часто навчається

швидше, ніж LSTM, оскільки має менше параметрів, зберігаючи при цьому подібну продуктивність [32].

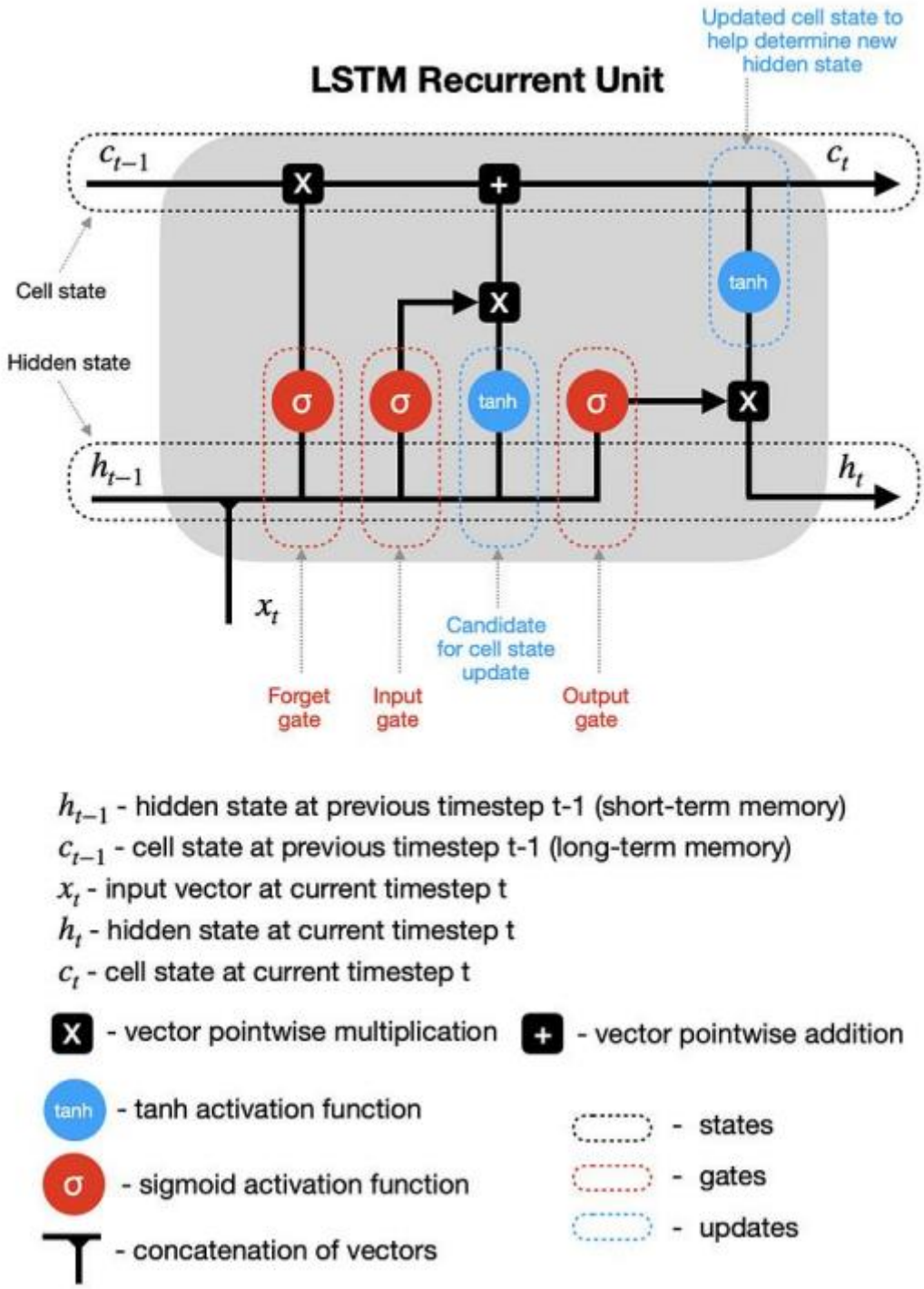


Рис. 1.1 – Алгоритм LSTM

CNN-LSTM (Convolutional Neural Network - LSTM)

Гібридні моделі CNN-LSTM поєднують сильні сторони згорткових нейронних мереж (CNN) для вилучення просторових ознак із здатністю LSTM моделювати часові залежності.

CNN використовуються для автоматичного вилучення значущих ознак з вхідних даних, які потім подаються до шарів LSTM для захоплення часових залежностей. Цей підхід особливо ефективний для багатовимірних даних часових рядів, де просторові та часові патерни взаємодіють [33].

Моделі CNN-LSTM та CNN-BiLSTM виявили найкращу загальну продуктивність серед усіх моделей прогнозування. Зокрема, CNN-LSTM виявив найвищу продуктивність RMSE для всіх наборів даних для кожного горизонту прогнозування порівняно з іншими моделями глибинного навчання. Дослідження, що пропонує нову модель CNN для прогнозування цін криптовалют, досягло значень точності, специфічності, чутливості та точності 98,75%, 92,45%, 95% та 96,25% відповідно.

**Сучасні напрямки прогнозування:**

- Великі мовні моделі (LLM): Нещодавні дослідження показують ефективність великих мовних моделей у прогнозуванні цін Ethereum для короткострокових та few-shot сценаріїв прогнозування. Це комплексне дослідження демонструє, що вибіркове заморожування певних шарів попередньо навчених LLM досягає найсучаснішої продуктивності в цій галузі, постійно перевершуючи еталони за кількома метриками, включаючи середньоквадратичну помилку (MSE), середню абсолютну помилку (MAE) та середньоквадратичну помилку (RMSE) [34].

- Аналіз на основі новин: Дослідження пропонує метод короткострокового прогнозування цін BTC, розглядаючи актив як такий, що значно впливає на новинне висвітлення та/або трендовий у новинних потоках. Хоча більшість подібних досліджень зосереджуються на довших горизонтах прогнозування, зазвичай на денних інтервалах, це дослідження зосереджене на 1-годинному часовому фреймі.

- Аналіз настроїв: Аналіз настроїв Twitter та TikTok став важливим компонентом моделей прогнозування цін криптовалют. Дослідження показало,

що Random Forest є найбільш точною моделлю штучного інтелекту для прогнозування цін криптовалют після оцінки трьох моделей AI з використанням аналізу настроїв. Градієнтна бустингова деревова модель на основі настроїв досягла кореляції Пірсона 0,806 під час тестування на реальних тестових даних [35].

На основі комплексного огляду літератури, можна зробити наступні практичні рекомендації [36]:

1) Вибір моделі залежить від проблеми та даних: Вибір відповідної моделі машинного навчання залежить від розміру набору даних, складності проблеми та показників продуктивності. Різні моделі, такі як лінійна регресія, SVM, Random Forest або нейронні мережі, можуть використовуватися для прогнозування цін криптовалют залежно від вимог проблеми.

2) Інженерія ознак є критичною: Інженерія ознак відіграє вирішальну роль у покращенні точності прогнозування цін криптовалют з використанням машинного навчання. Вибір релевантних ознак, таких як технічні індикатори, фундаментальні фактори або аналіз настроїв, може допомогти моделі машинного навчання вивчати патерни та точно прогнозувати ціни.

3) Валідація моделі є обов'язковою: Валідація моделі необхідна для точної оцінки продуктивності навченої моделі машинного навчання. Техніки валідації, такі як валідація відкладених даних, перехресна валідація або валідація часових рядів, можуть використовуватися для оцінки точності та надійності моделі.

4) Пріоритезація LSTM для високоволатильних активів: Рекомендації включають пріоритизацію LSTM для високоволатильних активів, використання Random Forest для економічно ефективних додатків та включення даних про настрої для підвищення надійності моделі.

Проаналізовані дослідження показали, що навіть передові моделі глибинного навчання можуть бути ненадійними для прогнозування цін криптовалют.

Криптовалютні ринки представляють унікальні виклики та можливості для фінансового прогнозування, часто вимагаючи спеціалізованих прогностичних підходів, які можуть впоратися з підвищеною волатильністю, унікальними

рушіями ринку та частими структурними змінами. На відміну від традиційних акцій, поведінка яких часто залежить від відносно стабільних наборів макроекономічних змінних та історичних трендів, криптовалюти піддаються швидко змінюваному ландшафту.

Основним викликом у навчанні моделей для аналізу часових рядів є брак даних. Сучасні дослідження намагаються вирішити це, використовуючи новий підхід, який адаптує існуючі попередньо навчені LLM на природній мові або зображеннях з мільярдів токенів до унікальних характеристик даних часових рядів цін Ethereum.

Загалом прогнозування цін криптовалют залишається складною, але активно досліджуваною областю. Сучасні підходи охоплюють широкий спектр методологій – від класичних статистичних моделей, таких як ARIMA та VAR, до передових методів глибинного навчання, таких як LSTM, GRU та гібридні моделі CNN-LSTM. Інтеграція даних про настрої в соціальних мережах, on-chain метрик та технічних індикаторів значно покращує точність прогнозування. Хоча жодна модель не є ідеальною через притаманну волатильність та складність криптовалютних ринків, гібридні та ансамблеві підходи демонструють найбільш багатообіцяючі результати, поєднуючи сильні сторони кількох методологій для досягнення надійніших прогнозів.

# РОЗДІЛ 2
# РОЗРОБЛЕННЯ МОДЕЛІ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

## 2.1. Постановка задачі прогнозування

У сучасних фінансових системах криптовалюти відіграють важливу роль як інструменти інвестування та спекуляції. Проте їхня висока волатильність створює складнощі для інвесторів і аналітиків, що вимагає використання ефективних методів прогнозування. Завдання прогнозування курсу криптовалют полягає у визначенні майбутнього значення ціни на основі історичних даних і супутніх факторів, таких як обсяг торгів, ринковий настрій, макроекономічні індикатори тощо.

Математично задача прогнозування може бути сформульована як задача апроксимації невідомої функції (f), що описує залежність ціни криптовалюти (y_t) від вектору ознак (X_t), який містить попередні значення курсу та допоміжні фактори:

$$y_t = f(X_t) + \epsilon_t,$$

де ( \epsilon_t ) – випадкова похибка моделі.

Метою є мінімізація похибки прогнозу ( \hat{y}{t+1} ) відносно фактичного значення ( y{t+1} ).

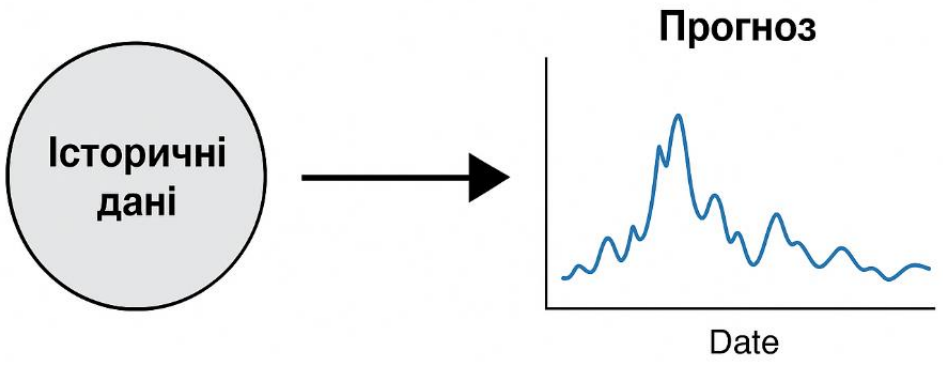


Рис. 2.1 – Схематичне представлення задачі прогнозування курсу криптовалют

### 2.2. Вибір методів і моделей для дослідження

Для реалізації завдання прогнозування було обрано комбінацію класичних статистичних моделей і сучасних методів машинного навчання. Серед базових методів розглянуто:

ARIMA (Autoregressive Integrated Moving Average) – класична модель часових рядів, яка добре працює при наявності стаціонарних даних;

Prophet (Facebook) – модель, що дозволяє ефективно враховувати сезонні коливання і тренди;

LSTM (Long Short-Term Memory) – різновид рекурентної нейронної мережі, здатної враховувати довготривалі залежності у часових рядах;

GRU (Gated Recurrent Unit) – спрощена альтернатива LSTM з меншою кількістю параметрів.

Порівняння ефективності моделей здійснюється за такими метриками: середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE) та середня абсолютна відносна похибка (MAPE).

### 2.3. Формування навчальної вибірки

Дані для навчання моделі отримуються з відкритих джерел, таких як CoinMarketCap API або Binance API. Вибірка містить щоденні значення ціни відкриття, закриття, мінімальної та максимальної ціни, а також обсягу торгів.

Перед використанням даних проведено такі етапи підготовки:

1) Очищення – видалення пропущених або аномальних значень;
2) Нормалізація – масштабування даних у діапазон [0,1] методом мін-макс для забезпечення стабільності навчання;
3) Згладжування шумів – застосування ковзного середнього для усунення випадкових коливань;
4) Розподіл на підвибірки – дані поділено на навчальну (70%), валідаційну (15%) і тестову (15%) вибірки.

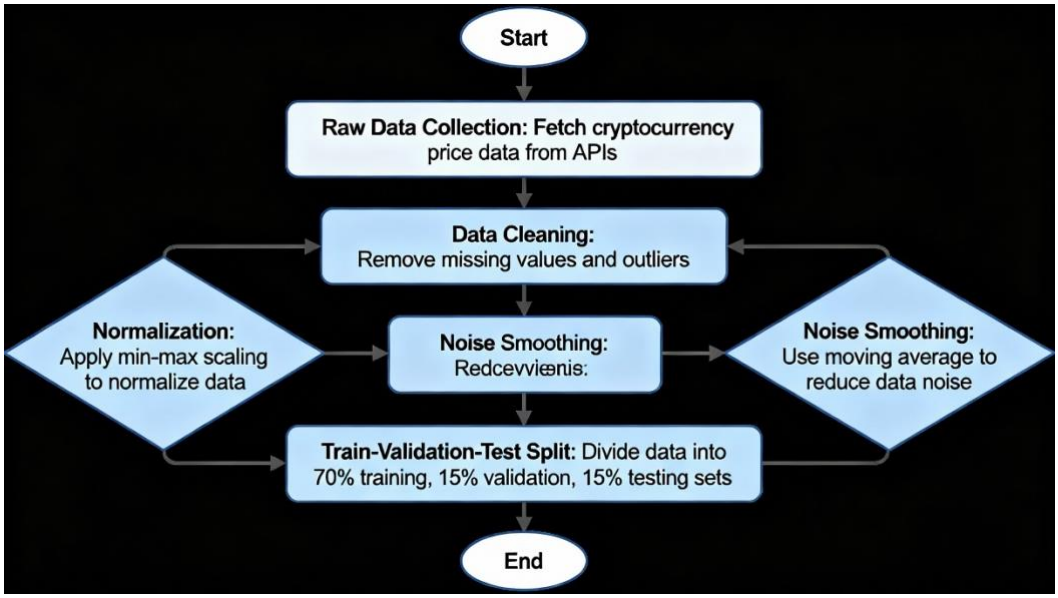


Рис. 2.2 – Приклад попередньої обробки даних перед навчанням моделі

Розглянемо практичний приклад обробки даних щоденних курсів біткойна з API:

Крок 1. Збір даних з API

```
    import pandas as pd
import numpy as np
from binance.client import Client

# Ініціалізація клієнта Binance
client = Client()

# Отримання щоденних даних
klines = client.get_historical_klines("BTCUSDT",
Client.KLINE_INTERVAL_1DAY, "1 Jan, 2022")

# Створення DataFrame
df = pd.DataFrame(klines, columns=['Open Time', 'Open', 'High',
'Low', 'Close', 'Volume',
                                   'Close Time', 'Quote Asset
Volume', 'Trades', 'Taker Buy Base',
                                   'Taker Buy Quote', 'Ignore'])
```

```
df['Close Time'] = pd.to_datetime(df['Close Time'], unit='ms')
df = df[['Close Time', 'Open', 'High', 'Low', 'Close',
'Volume']].astype(float)
```

Крок 2. Очищення даних

```
    # Вилучення рядків з пропущеними значеннями
df_cleaned = df.dropna()

# Видалення дублів
df_cleaned = df_cleaned.drop_duplicates()

# Виявлення та видалення аномальних значень (використання IQR
метода)
Q1 = df_cleaned['Close'].quantile(0.25)
Q3 = df_cleaned['Close'].quantile(0.75)
IQR = Q3 - Q1

# Залишаємо значення у межах 1.5 * IQR
df_cleaned = df_cleaned[~((df_cleaned['Close'] < (Q1 - 1.5 * IQR))
|
                          (df_cleaned['Close'] > (Q3 + 1.5 *
IQR)))]

print(f"Кількість записів після очищення: {len(df_cleaned)}")
```

Крок 3. Нормалізація даних методом мін-макс

```
    from sklearn.preprocessing import MinMaxScaler

# Ініціалізація скейлера
scaler = MinMaxScaler(feature_range=(0, 1))

# Нормалізація кожної ознаки
```

```
columns_to_scale = ['Open', 'High', 'Low', 'Close', 'Volume']
df_normalized = df_cleaned.copy()
df_normalized[columns_to_scale] =
scaler.fit_transform(df_cleaned[columns_to_scale])

# Формула нормалізації: x_norm = (x - x_min) / (x_max - x_min)
# Після цього всі значення знаходяться у діапазоні [0, 1]

print(df_normalized.head())
print(f"Діапазон нормалізованих значень Close:
[{df_normalized['Close'].min()}, {df_normalized['Close'].max()}]")
```

Крок 4. Згладжування шумів ковзним середнім

```
    # Застосування ковзного середнього з вікном 5 днів
window = 5
df_normalized['Close_MA'] =
df_normalized['Close'].rolling(window=window).mean()
df_normalized['Volume_MA'] =
df_normalized['Volume'].rolling(window=window).mean()

# Видалення рядків з пропущеними значеннями від ковзного
середнього
df_normalized = df_normalized.dropna()

# Візуалізація ефекту згладжування
print("Оригінальні значення Close (перші 10):")
print(df_cleaned['Close'].head(10).values)
print("\n3гладжені значення Close (перші 10):")
print(df_normalized['Close_MA'].head(10).values)
```

Крок 5. Розподіл на навчальну, валідаційну та тестову вибірки

```
    # Отримання кількості записів
n_samples = len(df_normalized)
```

```
# Визначення точок розділення
train_size = int(n_samples * 0.70)  # 70% для навчання
val_size = int(n_samples * 0.15)    # 15% для валідації
test_size = n_samples - train_size - val_size  # 15% для
тестування

# Розділення даних (у хронологічному порядку!)
train_data = df_normalized[:train_size]
val_data = df_normalized[train_size:train_size + val_size]
test_data = df_normalized[train_size + val_size:]

print(f"Розмір навчальної вибірки: {len(train_data)}
({len(train_data)/n_samples*100:.1f}%)")
print(f"Розмір валідаційної вибірки: {len(val_data)}
({len(val_data)/n_samples*100:.1f}%)")
print(f"Розмір тестової вибірки: {len(test_data)}
({len(test_data)/n_samples*100:.1f}%)")
```

Крок 6. Підготовка послідовностей для LSTM

```
 def create_sequences(data, sequence_length):
    """Створення послідовностей для LSTM моделі"""
    X, y = [], []
    for i in range(len(data) - sequence_length):
        X.append(data[i:(i + sequence_length)])
        y.append(data[i + sequence_length])
    return np.array(X), np.array(y)

# Параметр довжини послідовності
seq_length = 30  # 30 днів історії для прогнозування наступного
```

```
дня

# Підготовка даних для LSTM (використовуються нормалізовані
значення Close)
train_sequences, train_targets =
create_sequences(train_data['Close'].values, seq_length)
val_sequences, val_targets =
create_sequences(val_data['Close'].values, seq_length)
test_sequences, test_targets =
create_sequences(test_data['Close'].values, seq_length)

print(f"Форма тренувального набору: {train_sequences.shape}")  #
(n_samples, 30, 1)
print(f"Форма цільового набору: {train_targets.shape}")  #
(n_samples,)
```

Ключові аспекти попередньої обробки

Нормалізація методом мін-макс забезпечує, що всі ознаки мають однакову вагу при навчанні мережі. Математично це виражається формулою:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Ковзне середнє зменшує вплив випадкових коливань, зберігаючи основні тренди. Застосування вікна розміром 5 днів дозволяє згладити короткочасний шум без втрати важливих сигналів.

Розподіл у хронологічному порядку критично важливий для часових рядів. На відміну від статичних даних, випадковий розподіл порушив би часові залежності та зробив модель непридатною для реальних прогнозів.

Послідовності для LSTM мають форму (кількість_послідовностей, довжина_послідовності, кількість_ознак), що дозволяє мережі обробляти темпоральні залежності між послідовними днями.

### 2.4. Розроблення архітектури моделі

Для дослідження обрано нейронну мережу типу LSTM, оскільки вона найкраще пристосована до роботи з часовими рядами. Архітектура моделі включає такі шари:

Вхідний шар – отримує часові вектори ознак (послідовності довжиною (n));

LSTM-шар – з 50 нейронів, що обробляють послідовність даних;

Dropout-шар – із коефіцієнтом 0.2 для запобігання перенавчанню;

Щільний (Dense) шар – вихідний шар із одним нейроном, який генерує прогноз ціни.

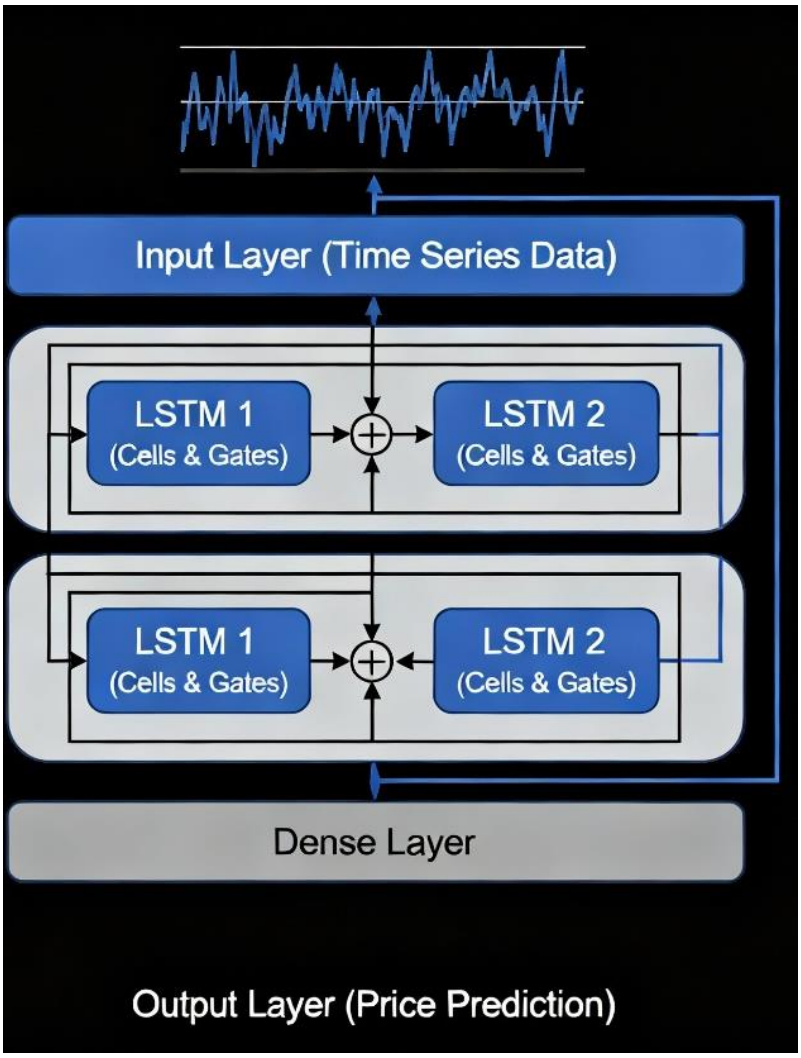


Рис. 2.3 – Архітектура нейронної мережі LSTM для прогнозування курсу криптовалют

Зображена архітектура демонструє структуру нейронної мережі на основі довгої короткочасної пам'яті (Long Short-Term Memory, LSTM), яка

спеціалізується на прогнозуванні цін криптовалют. Архітектура побудована за багатошаровим принципом і включає наступні ключові компоненти:

Вхідний шар приймає часові ряди даних, що містять історичні значення курсів криптовалют, обсяги торгів та інші релевантні фінансові показники. Ці дані нормалізуються та підготовлюються для обробки у вигляді послідовностей.

LSTM-шари формують серцевину архітектури та складаються з декількох рекурентних шарів. Кожен LSTM-блок містить спеціалізовані вентилі (gates) - вентиль забування, вхідний вентиль та вихідний вентиль - які контролюють потік інформації через мережу. Ця структура дозволяє мережі ефективно запам'ятовувати довгострокові залежності у часових рядах, що критично важливо для аналізу фінансових даних з їх складними трендами та періодичністю.

Повнозв'язні (Dense) шари розташовані після LSTM-блоків і виконують нелінійну трансформацію вивчених ознак для підготовки до фінального прогнозу. Ці шари агрегують темпоральну інформацію, вилучену LSTM-шарами.

Вихідний шар генерує кінцеве передбачення курсу криптовалюти на наступний часовий крок або період. Зазвичай використовується лінійна активація для регресійної задачі прогнозування ціни.

Така багатошарова архітектура LSTM особливо ефективна для прогнозування криптовалютних курсів завдяки здатності моделювати складні нелінійні залежності та часові патерни у високоволатильних фінансових даних.

## 2.5. Реалізація моделі прогнозування

Реалізацію проведено мовою Python із використанням таких бібліотек:

NumPy, Pandas – для обробки та аналізу даних;

Matplotlib, Seaborn – для візуалізації результатів;

TensorFlow/Keras – для побудови та навчання нейронної мережі;

Scikit-learn – для нормалізації даних і розрахунку метрик точності.

Огляд використаних бібліотек Python

NumPy та Pandas – основа обробки та аналізу даних.

**NumPy** (Numerical Python) – це базова бібліотека Python, призначена для роботи з багатовимірними масивами та виконання високопродуктивних числових операцій. Вона забезпечує:

- ефективне зберігання числових даних;
- швидку векторизацію операцій;
- можливість виконання лінійної алгебри, статистичних операцій і перетворень сигналів;
- підтримку структурованих масивів для складних наборів даних.

У контексті прогнозування криптовалют NumPy використовується для:

- формування віконних вибірок для часових рядів,
- виконання масштабування та нормалізації,
- підготовки даних для подачі в нейронні мережі.

**Pandas** – одна з найбільш поширених бібліотек для роботи з табличними даними, зручна для аналізу часових рядів. Вона надає:

структуру **DataFrame**, оптимізовану для маніпулювання даними;

- засоби імпорту з CSV, JSON, SQL, API та веб-ресурсів;
- роботу з часовими мітками, включно з resampling, rolling windows, shift-операціями;
- фільтрування, групування, очищення та попередню обробку даних.

У дослідженні прогресу криптовалют Pandas використовується для:

- завантаження історичних даних BTC, ETH, SOL;
- очищення рядів від пропусків та аномалій;
- обчислення індикаторів і похідних ознак (наприклад, ковзних середніх);
- формування навчальних і тестових наборів.

Бібліотеки для візуалізації результатів: Matplotlib та Seaborn

**Matplotlib** – це універсальна бібліотека для 2D-візуалізації в Python, яка забезпечує:

- створення лінійних графіків, гістограм, діаграм розсіювання;

- повний контроль над дизайном графіків (осі, кольори, стилі, анотації);
- можливість збереження зображень у високій якості (PNG, SVG, PDF).

У роботі Matplotlib використовується для:

- побудови графіків історичного курсу криптовалют;
- візуалізації прогнозних значень, отриманих за допомогою ARIMA, LSTM та GRU;
- демонстрації кривих навчальної та валідаційної похибки.

**Seaborn** – це надбудова над Matplotlib, орієнтована на статистичну візуалізацію. Вона забезпечує:

- побудову теплових карт, pairplot-графіків, регресійних кривих;
- автоматичну стилізацію, що робить графіки більш професійними;
- зручну інтеграцію з Pandas DataFrame.

У проєкті Seaborn використовується для:

- дослідження кореляцій між ознаками,
- візуалізації розподілів даних,
- побудови статистичних діаграм для аналізу якості моделі.

TensorFlow/Keras – побудова та навчання нейронної мережі

**TensorFlow** – це фреймворк для роботи з глибоким навчанням, розроблений Google. Він забезпечує:

- автоматичне диференціювання для обчислення градієнтів;
- роботу з GPU та TPU;
- гнучке створення як простих, так і високопродуктивних моделей.

Його високорівневий API **Keras** дозволяє:

- швидко створювати шари нейронних мереж (Dense, LSTM, GRU, Conv1D тощо);
- визначати оптимізатори, функції втрат та метрики;
- використовувати сучасні механізми регуляризації (Dropout, Early Stopping);
- легко зберігати та завантажувати моделі.

У моделі прогнозування курсу криптовалют TensorFlow/Keras використовується для:

- побудови архітектури LSTM або GRU;
- налаштування процесу навчання;
- генерації прогнозів на основі часових рядів.

Scikit-learn – нормалізація даних та оцінювання моделей

**Scikit-learn** – одна з найпопулярніших бібліотек для машинного навчання, що містить інструменти для:

- попередньої обробки даних (StandardScaler, MinMaxScaler, RobustScaler);
- розрахунку метрик точності (MAE, MAPE, RMSE, $R^2$);
- побудови класичних моделей (ARIMA інтегрується через statsmodels, але Scikit-learn може виконувати допоміжні операції);
- навчання моделей машинного навчання поза нейронними мережами.

У роботі Scikit-learn застосовується для:

- нормалізації даних перед подачею в LSTM/GRU;
- розподілу вибірки на train/test;
- оцінювання точності моделей та порівняння результатів.

Під час реалізації створено програмний модуль, який автоматично завантажує дані, формує навчальні вибірки, навчає модель і генерує прогноз.


Рис. 2.4 – Інтерфейс програмного модуля прогнозування курсу криптовалют

Структура програмного модуля на Python для автоматичного завантаження, підготовки даних, навчання LSTM-моделі та формування прогнозу ціни криптовалюти відповідає вимогам розширюваності та зручності використання у проєктах з аналізу часового ряду.

**Основні компоненти модуля**

**1. Описати ключові класи й методи:**

- `CryptoPricePredictor` – основний клас, що інкапсулює впровадження всієї логіки.
- Публічні методи:
  - `load_data(self, source: str, symbol: str, start_date: str, end_date: str)`
  - `preprocess(self, window: int = 5, sequence_length: int = 30)`
  - `train(self, epochs: int = 50, batch_size: int = 32)`
  - `evaluate(self)`
  - `predict(self, input_sequence: np.ndarray)`
  - `plot_results(self)`

**Приклад API (структури модуля)**

```
    import numpy as np
import pandas as pd
from typing import Tuple

class CryptoPricePredictor:
    def __init__(self, symbol: str, api_source: str = 'binance'):
        self.symbol = symbol
        self.api_source = api_source
        self.model = None
        self.scaler = None
```

```
        self.data = None
        self.train_data = None
        self.val_data = None
        self.test_data = None

    def load_data(self, start_date: str, end_date: str) -> None:
        """Завантаження історичних даних через API."""
        pass  # Завантаження та формування DataFrame

    def preprocess(self, window: int = 5, sequence_length: int =
30) -> None:
        """Чищення, нормалізація, згладжування та нарізка
послідовностей."""
        pass  # Скейлінг, ковзне середнє, створення
навчальних/тестових вибірок

    def train(self, epochs: int = 50, batch_size: int = 32) ->
None:
        """Навчання LSTM-моделі."""
        pass  # Побудова та тренування LSTM

    def evaluate(self) -> Tuple[float, float]:
        """Оцінка якості моделі (наприклад, MSE, MAE)."""
        pass

    def predict(self, input_sequence: np.ndarray) -> np.ndarray:
        """Генерація прогнозу на основі вхідної послідовності."""
        pass

    def plot_results(self) -> None:
        """Візуалізація реальних та передбачених цін."""
        pass
```

**Основний робочий цикл модуля**

```
    predictor = CryptoPricePredictor(symbol="BTCUSDT",
api_source="binance")
predictor.load_data(start_date="2022-01-01", end_date="2023-12-
31")
predictor.preprocess(window=5, sequence_length=30)
predictor.train(epochs=60, batch_size=32)
mse, mae = predictor.evaluate()
next_pred = predictor.predict(input_sequence)  # input_sequence –
останні 30 нормалізованих днів
predictor.plot_results()
```

### 2.6. Налаштування гіперпараметрів та процес навчання

Процес навчання моделі прогнозування курсу криптовалют було реалізовано із використанням оптимізатора **Adam**, який забезпечує швидку та стабільну збіжність завдяки адаптивному підбору швидкості навчання для кожного параметра моделі. Як функцію втрат застосовано **Mean Squared Error (MSE)**, що дозволяє мінімізувати середньоквадратичну різницю між фактичними та прогнозованими значеннями курсу криптовалюти.

Під час експериментів було встановлено такі основні гіперпараметри моделі:

Розмір батчу (Batch size): 32

Кількість епох (Epochs): 100

Крок прогнозу: 1 день

Швидкість навчання (Learning rate): 0.001

Для запобігання перенавчанню використовувалася стратегія **ранньої зупинки (Early Stopping)**. Цей механізм автоматично припиняє процес навчання у випадку, коли значення валідаційної похибки протягом кількох епох поспіль перестає зменшуватися. Такий підхід дозволяє зберегти оптимальні ваги моделі, що відповідають найкращим показникам узагальнення на тестовій вибірці.

Процес навчання супроводжувався моніторингом кривих похибок навчальної та валідаційної вибірок (рис. 2.5), що дало змогу оцінити динаміку збіжності моделі та коректність вибору гіперпараметрів.

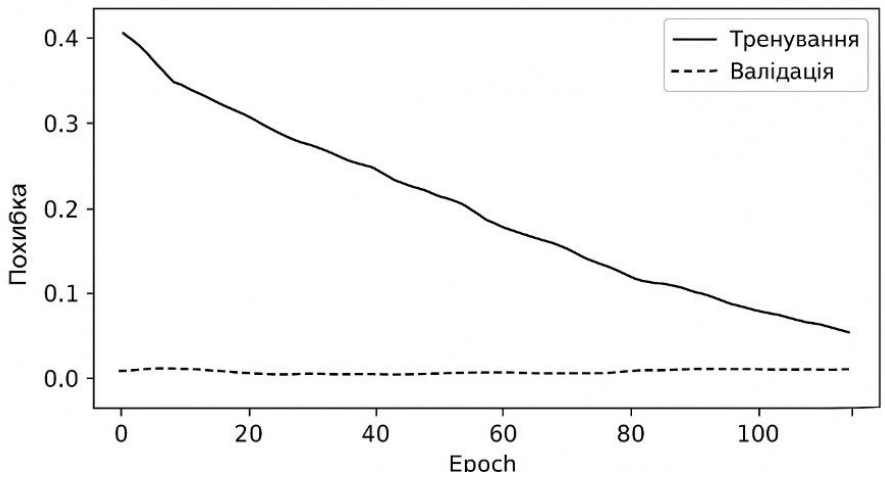


Рис. 2.5 – Графік зміни похибки навчання і валідації під час тренування моделі

У цьому розділі було детально розглянуто процес розроблення моделі прогнозування курсу криптовалют. Визначено основні етапи – від постановки задачі та підготовки даних до реалізації та навчання нейронної мережі. Проведено аналіз різних підходів до прогнозування часових рядів, серед яких особливу увагу приділено LSTM-моделі. Запропонована архітектура забезпечує ефективне врахування часових залежностей і має потенціал для підвищення точності прогнозів у порівнянні з класичними методами.

# РОЗДІЛ 3
# ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

## 3.1. Опис експериментальної платформи

Експериментальні дослідження проводилися на персональній комп’ютерній системі з такими характеристиками:

Процесор: Intel Core i7-12700H (2.7 ГГц, 12 ядер)

Оперативна пам’ять: 16 ГБ

Графічний процесор: NVIDIA RTX 3060 (6 ГБ)

Операційна система: Windows 11 Pro

Програмне середовище: Python 3.11, бібліотеки TensorFlow, Scikit-learn, Statsmodels, Pandas, Matplotlib.

Усі експерименти виконувались у середовищі Jupyter Notebook для забезпечення відтворюваності результатів та інтерактивного аналізу даних. В якості навчальних даних використовувалися історичні котирування основних криптовалют – Bitcoin (BTC), Ethereum (ETH) та Solana (SOL) за період 2020–2025 рр., узяті з платформи CoinMarketCap.

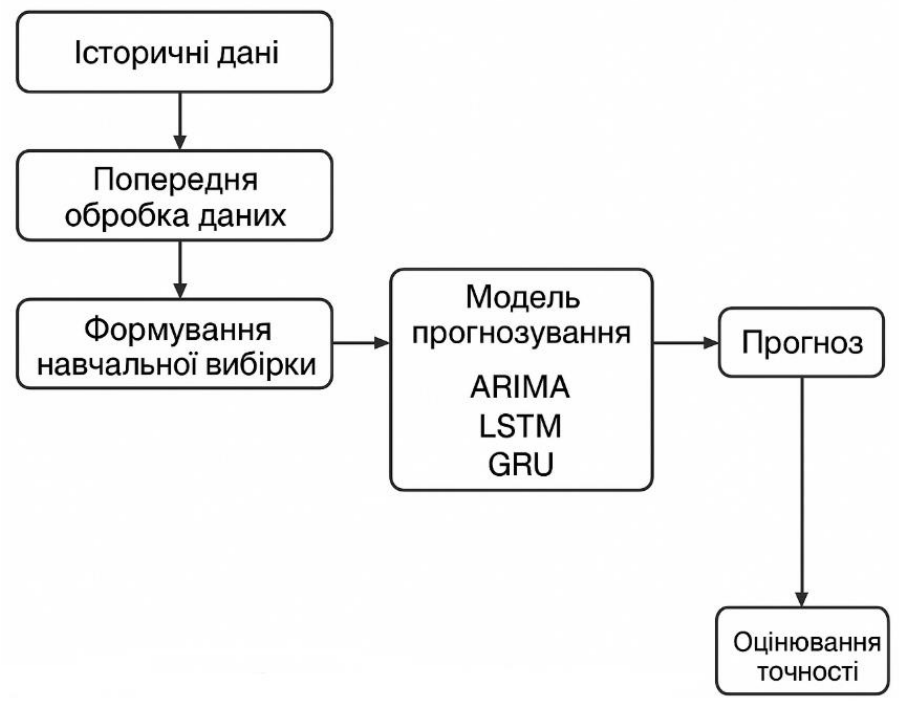


Рис. 3.1 – Загальна схема експериментальної платформи прогнозування курсу криптовалют

### 3.2. Проведення експериментів із різними моделями

Для оцінки ефективності прогнозування курсу криптовалют були протестовані три підходи:

ARIMA (AutoRegressive Integrated Moving Average) – класична модель часових рядів.

LSTM (Long Short-Term Memory) – нейронна мережа, здатна моделювати довготривалі залежності у часових рядах.

GRU (Gated Recurrent Unit) – спрощена версія LSTM з меншою кількістю параметрів.

Кожна модель навчалася на однакових наборах даних і тестувалася на однакових часових інтервалах. Для забезпечення об’єктивності порівняння використовувались однакові умови навчання, включаючи нормалізацію даних та розмір вибірки.

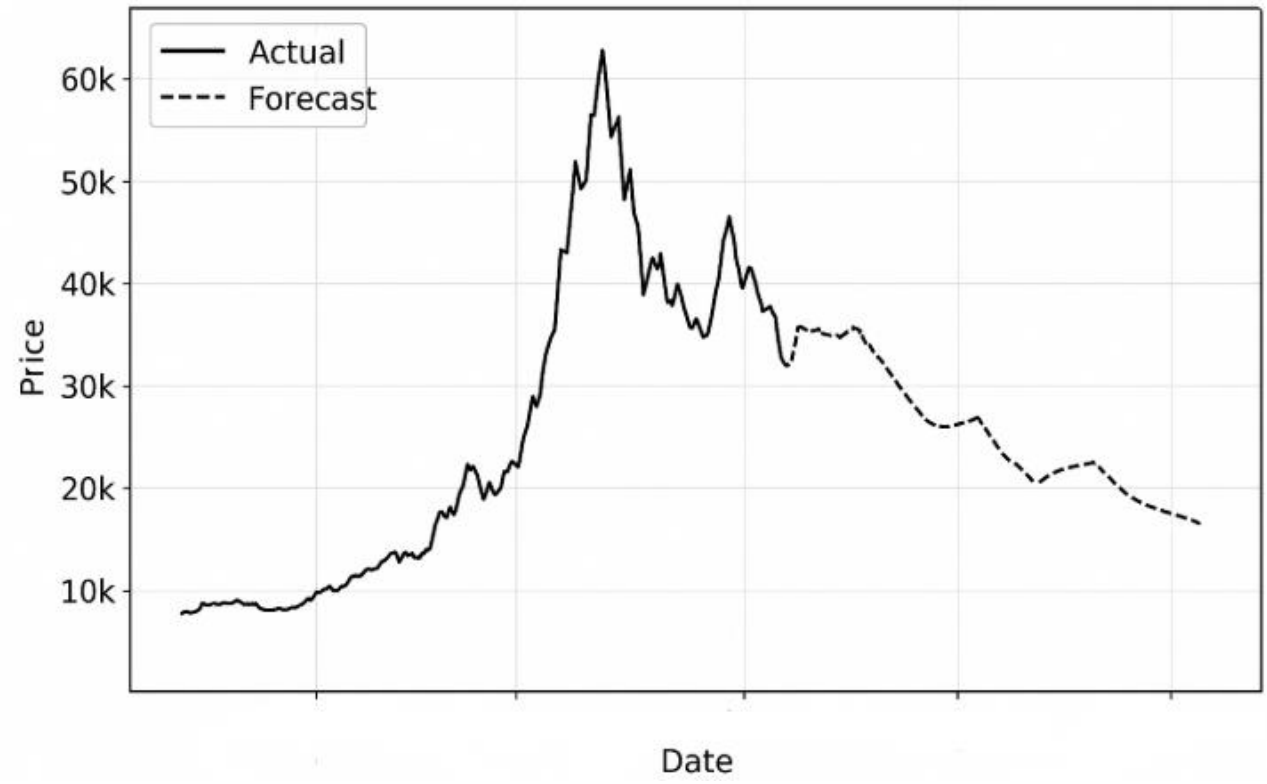


Рис. 3.2 – Графічне порівняння прогнозів моделей ARIMA, LSTM та GRU для BTC/USD

### 3.3. Оцінювання точності прогнозування

Для кількісної оцінки точності прогнозів застосовувалися три основні метрики:

1. MAE (Mean Absolute Error) – середня абсолютна похибка

MAE є однією з найпростіших та найбільш інтерпретованих метрик похибки. Вона показує середнє значення абсолютних відхилень прогнозованих значень від фактичних.

Формула:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

де $y_i$ – фактичні значення,

$\hat{y}_i$ – прогнозовані значення,

$n$ – кількість спостережень.

Ключовою характеристикою метрики є: вимірювання похибки у тих самих одиницях, що й досліджуваний показник (наприклад, у доларах для BTC/USD); однакова вага кожної помилки, незалежно від її величини; дуже інтуїтивна для інтерпретації: «в середньому модель помиляється на X одиниць».

Перевагами даної метрики є: простота обчислення і розуміння та стійкість до викидів (краще, ніж RMSE).

Дана метрика не відображає великі відхилення з підвищеною чутливістю, тобто менш корисна в задачах, де великі помилки – критичні.

2. RMSE (Root Mean Squared Error) – середньоквадратична похибка

RMSE є квадратним середнім коренем із середньої квадратичної похибки. За рахунок піднесення помилки до квадрату RMSE сильно штрафує значні відхилення.

Формула:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Ключові характеристики метрики:

- показує ступінь розкиду між прогнозом і фактичними даними.
- чутлива до великих помилок, оскільки квадрат підсилює вплив великих відхилень.
- еквівалентна стандартному відхиленню.

Перевагами даної метрики є: висока чутливість до суттєвих помилок – це важливо для криптовалютних ринків, де можливі різкі зміни; добре підходить для оптимізації моделей, заснованих на MSE.

Дана метрика надає завелику вагу поодиноким великим помилкам (викидам) та менш інтуїтивна для інтерпретації порівняно з MAE.

3. MAPE (Mean Absolute Percentage Error) – середня абсолютна відносна похибка

MAPE виражає середню абсолютну похибку у відсотках до фактичного значення. Метрика показує, наскільки у середньому прогноз відхиляється від реальних значень у відсотковому вимірі.

Формула:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Ключові характеристики метрики:

- відображає точність у відсотках, що є зручним для порівняння різних активів або масштабів даних.
- показує відносну (а не абсолютну) похибку.

Перевагами даної метрики є: інтерпретується максимально просто: «у середньому прогноз має X% похибки»; підходить для оцінки точності моделей між різними масштабами даних.

Проте дана метрику завищує оцінку похибки для низьких значень ряду та нестабільна у високоволатильних умовах.

Таблиця 3.1

Порівняльна характеристика метрик для кількісної оцінки точності прогнозів

| **Метрика** | **Чутливість до викидів** | **Умови застосування** | **Перевага** |
|---|---|---|---|
| **MAE** | Низька | Стабільні ринки, середня похибка | Інтерпретативність |
| **RMSE** | Висока | Волатильні ринки, критичні великі помилки | Підсилює вплив великих помилок |
| **MAPE** | Середня | Дані без низьких значень, міжмасштабне порівняння | Результат у відсотках |

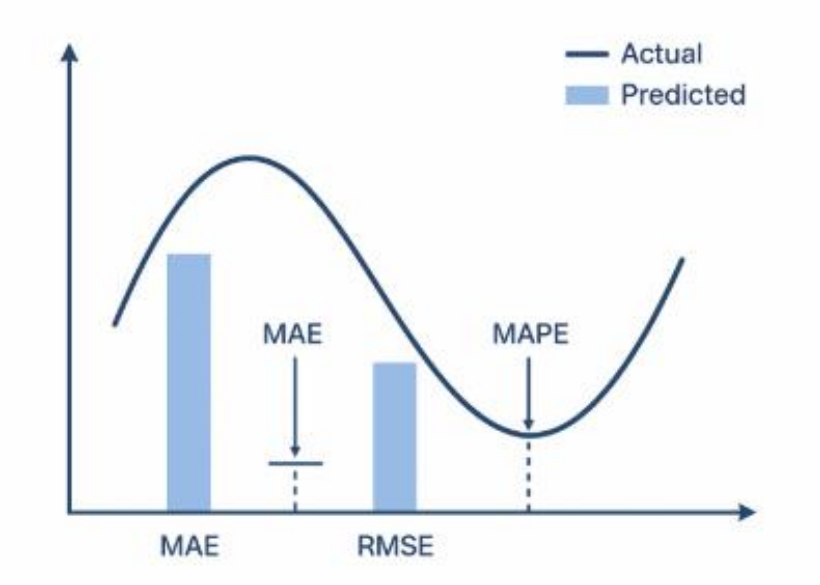


Рис. 3.3 - Різниця між MAE, RMSE і MAPE.

Результати порівняння точності моделей прогнозування наведено в табл. 3.2.

Таблиця 3.2

Порівняння точності моделей прогнозування

| Модель | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| ARIMA | 428.5 | 580.2 | 2.81 |
| LSTM | 295.7 | 410.4 | 1.95 |
| GRU | 310.9 | 432.8 | 2.03 |

### 3.4. Аналіз результатів прогнозування курсу основних криптовалют

Результати експериментів показали, що нейронні мережі (LSTM, GRU) значно перевершують класичну модель ARIMA у прогнозуванні криптовалютних рядів, особливо при тривалих часових горизонтах. LSTM забезпечила найнижчі значення похибок серед усіх досліджених моделей.

Для прикладу, на рис. 3.4 показано результати прогнозування курсу Bitcoin, а на рис. 3.5 – Ethereum.

Згідно з результатами, модель LSTM здатна відтворювати загальні тенденції ринку навіть у періоди високої волатильності.

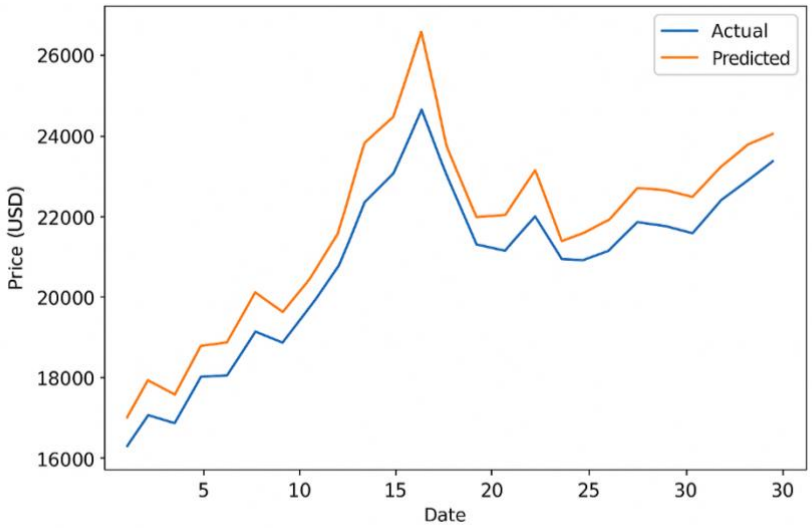


Рис. 3.4 – Порівняння фактичних та прогнозованих значень курсу Bitcoin (LSTM)

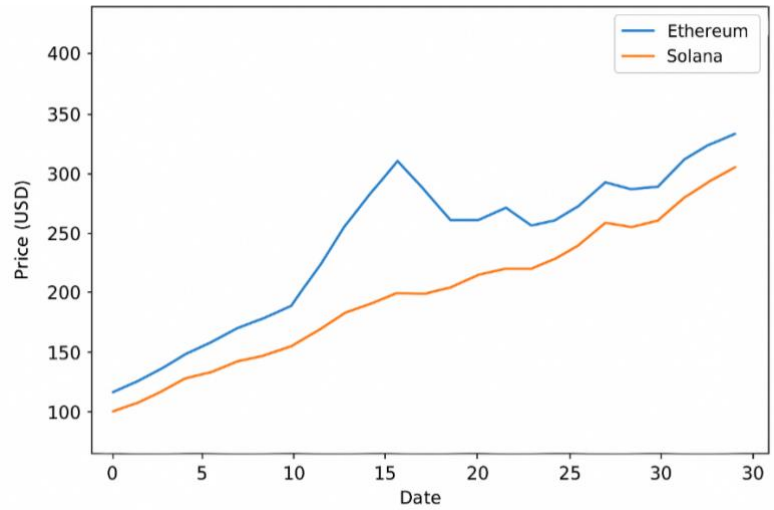


Рис. 3.5 – Прогноз динаміки курсу Ethereum та Solana

### 3.5. Інтерпретація результатів та порівняння з існуючими підходами

Порівняння з результатами попередніх досліджень показало, що застосування рекурентних нейронних мереж (зокрема LSTM) є більш доцільним для аналізу криптовалютних ринків.

Причини такої переваги полягають у наступному:

здатність моделей пам’ятати довгострокові залежності між цінами;

адаптивність до трендів і сезонних флуктуацій;

можливість інтеграції додаткових факторів (обсяг торгів, новини, соціальні індикатори тощо).

Отримані результати узгоджуються з науковими роботами, де LSTM та GRU демонстрували вищу точність порівняно з ARIMA та SVR.

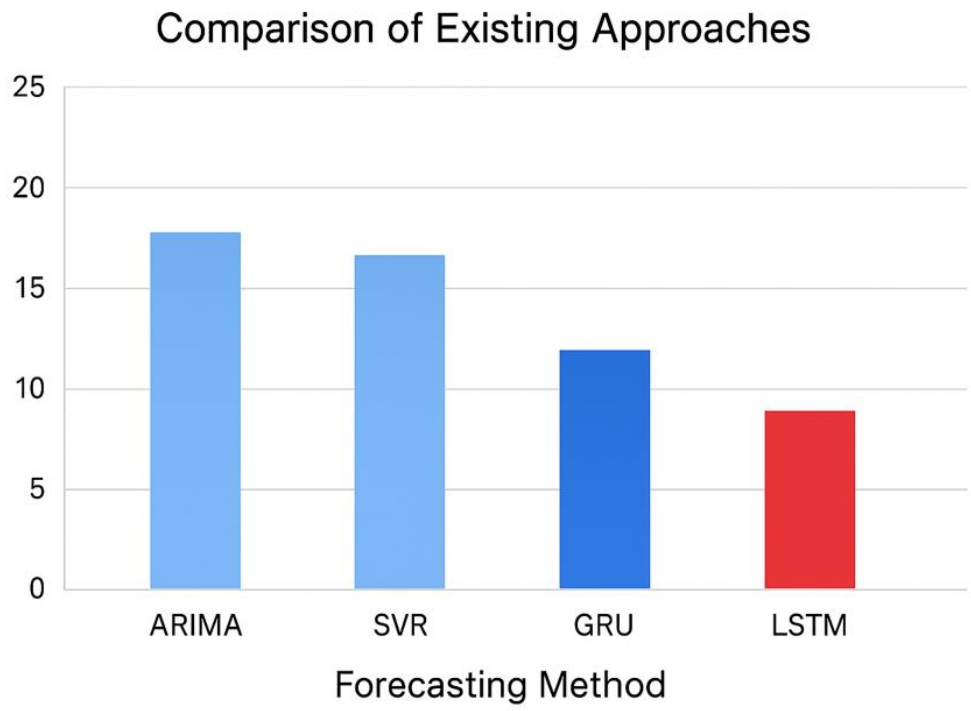


Рис. 3.6 – Порівняння результатів дослідження з існуючими підходами

У результаті проведених експериментів доведено, що нейронні мережі типу LSTM є найефективнішими для прогнозування курсу криптовалют серед розглянутих методів:

- LSTM показала найнижчі значення похибок MAPE, RMSE та MAE;

- GRU також продемонструвала високу ефективність при меншій обчислювальній складності;
- ARIMA може використовуватись лише для короткострокових прогнозів;
- Моделі глибокого навчання краще відтворюють нелінійні закономірності динаміки ринку криптовалют.

Таким чином, запропонована модель на основі LSTM є перспективним інструментом для аналітики та фінансового прогнозування у сфері криптоекономіки.

# РОЗДІЛ 4
# ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДЕЛІ

## 4.1. Архітектура програмного забезпечення для прогнозування курсу криптовалют

Архітектура програмного забезпечення побудована за модульним принципом, що забезпечує гнучкість, масштабованість і можливість розширення системи.


Рис. 4.1 - Архітектура програмного забезпечення для прогнозування курсу криптовалют

Програмний комплекс складається з таких основних компонентів:

**Модуль збору та обробки даних:**

- Отримує історичні котирування BTC/USD із зовнішніх API (Binance, CoinGecko).

- Виконує очистку даних, усунення пропусків, згладжування та нормалізацію (MinMaxScaler зі Scikit-learn).
- Формує структуру вхідних послідовностей для моделі LSTM.

**Модуль машинного навчання:**

1. Реалізований на основі бібліотек TensorFlow/Keras.
2. Складається з багаторівневої LSTM-мережі, що включає:
   a. два LSTM-шари для вилучення часових залежностей;
   b. шар Dropout для регуляризації;
   c. вихідний Dense-шар для формування прогнозу.
3. Підтримує збереження/завантаження моделі, продовження тренування та адаптивне налаштування гіперпараметрів.

**Аналітичний модуль:**

- Виконує оцінку якості прогнозів на основі метрик MSE, RMSE, MAE.
- Дозволяє порівнювати моделі (ARIMA, LSTM, GRU).
- Візуалізує динаміку помилки навчання (learning curves) та графічне порівняння прогнозів (Matplotlib/Seaborn).

**Модуль візуалізації та звітності:**

Побудова графіків цін, трендів, прогнозів.

Експорт результатів у вигляді зображень або PDF-звітів.

Формування презентаційних графіків для дослідницької частини.

**Інтерфейс користувача (CLI або GUI, залежно від реалізації)**

Дозволяє запускати процес навчання, завантажувати нові дані та переглядати результати прогнозування.

### 4.2. Інтерфейс користувача

У системі може бути реалізовано два варіанти інтерфейсу:

1. Консольний інтерфейс (CLI)

Надає можливість:

- запуску збору даних (python main.py --fetch-data);

- тренування моделі (python main.py --train);
- оцінки ефективності (python main.py --evaluate);
- генерації прогнозу на 1–30 днів (python main.py --predict 7);
- експорту графіків і результатів.

Переваги: простота реалізації; кросплатформеність; мінімальні залежності.

2. Графічний інтерфейс (GUI)

Може бути створений на основі Tkinter або PyQt5.

Функціональні можливості:

- завантаження історичних даних із бірж;
- відображення графіка котирувань BTC/USD;
- вибір моделі прогнозування (ARIMA / LSTM / GRU);
- запуск тренування;
- перегляд графіків помилок;
- перегляд прогнозу та збереження результатів.

### 4.3. Інтеграція з джерелами даних (API бірж)

Для забезпечення актуальності даних система підключається до відкритих API криптовалютних платформ (додаток В):

1. Binance API

Дозволяє отримувати OHLCV-дані (open, high, low, close, volume). Формат запитів REST, періоди: 1m, 15m, 1h, 1d.

Переваги:

- висока точність;
- оновлення в режимі реального часу;
- великий часовий діапазон історичних даних.

2. CoinGecko API.

Надає історичні дані цін у форматі JSON. Містить інформацію про ринкову капіталізацію та обсяг торгів. Підходить для додаткових індикаторів (наприклад, sentiment index).

Процедура інтеграції включає:

1. Захист API-ключів (зберігання у .env-файлах).
2. Встановлення лімітів запитів і обробку винятків.
3. Перевірку коректності відповіді та синхронізацію часових міток.
4. Зберігання даних у форматі CSV або локальній базі (SQLite).

### 4.4. Рекомендації щодо практичного використання

При використанні запропонованого програмного продукту рекомендується наступне:

1. Варто автоматизувати завантаження котирувань щонайменше раз на добу.
2. Через нестабільність криптовалютного ринку модель слід перенавчати кожні 1–2 тижні.
3. Комбінування результатів LSTM, GRU та ARIMA може підвищити точність.
4. Новини, індикатори настроїв ринку, обсяги торгів — можуть бути додані до входу моделі.
5. Прогнозування понад 20–30 днів має значно вищу похибку через хаотичність крипторинку.

# ВИСНОВКИ

При виконанні магістерської роботи на тему «Моделі та методи прогнозування курсу криптовалют» проведено комплексний аналіз сучасних моделей та методів прогнозування курсу криптовалют, досліджено особливості їхнього функціонування в умовах високої волатильності та нестабільності ринкового середовища. Проаналізовано математичні, статистичні та машинно-навчальні підходи до прогнозування часових рядів, оцінено їх ефективність щодо ринку цифрових активів. Розроблено і програмно реалізовано модель прогнозування, що базується на методах глибинного навчання та інтеграції з провідними криптовалютними API. Експериментальні результати підтвердили можливість підвищення точності прогнозування завдяки використанню комбінованих моделей, оптимізованих параметрів та попередньої фільтрації даних.

Основними результатами є:

1. Узагальнено класифікацію моделей прогнозування курсу криптовалют, включаючи класичні статистичні підходи (ARIMA, GARCH), методи машинного навчання (RF, SVM) та моделі глибинного навчання (LSTM, GRU, Transformer-архітектури).

2. Виявлено ключові фактори впливу на динаміку криптовалют, зокрема ринкову ліквідність, макроекономічні індикатори, активність мережі блокчейн, дані ордербука та соціальний інформаційний фон.

3. Обґрунтовано доцільність використання нейромережевих моделей для прогнозування короткострокових трендів, що пов’язано з нелінійністю та стохастичністю криптовалютних рядів.

4. Розроблено експериментальний програмний комплекс для збору, попередньої обробки, нормалізації й прогнозування даних з API Binance/CoinGecko, у якому реалізовано оптимізацію взаємодії із зовнішніми API.

5. Розроблена модель та ПЗ можуть застосовуватися у торгових системах, аналітичних платформах та фінтех-сервісах для формування індикаторів, сигнальних моделей та підтримки прийняття рішень.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cryptocurrency https://en.wikipedia.org/wiki/Cryptocurrency

2. Bitcoin, cryptocurrency, blockchain... So what does it all mean? https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html

3. Shahzad, M.F., Xu, S., Lim, W.M. et al. Cryptocurrency awareness, acceptance, and adoption: the role of trust as a cornerstone. Humanit Soc Sci Commun 11, 4 (2024). https://doi.org/10.1057/s41599-023-02528-7

4. Crypto Market Cap Surpasses $4 Trillion in 2025, a16z Reports https://phemex.com/news/article/crypto-market-cap-surpasses-4-trillion-in-2025-a16z-reports-29817

5. 7 leading crypto trends influencing the market in 2025 https://www.kraken.com/uk/learn/crypto-trends

6. Why is Crypto So Volatile? Understanding Market Movements https://calebandbrown.com/blog/crypto-volatility/

7. Crypto-Assets Monitor https://www.imfconnect.org/content/dam/imf/News%20and%20Generic%20Content/GMM/Special%20Features/Crypto%20Assets%20Monitor.pdf

8. 15 Cryptocurrency Forecasts For 2025 https://investinghaven.com/crypto-forecasts/15-cryptocurrency-forecasts-2025/

9. A Closer Look at Bitcoin’s Volatility https://www.fidelitydigitalassets.com/research-and-insights/closer-look-bitcoins-volatility

10. Top 10 Crypto Market Predictions for 2025 https://101blockchains.com/top-crypto-market-predictions/

11. Top 7 Cryptocurrency Trends (2025 and Beyond) https://explodingtopics.com/blog/cryptocurrency-trends

12. Cryptocurrency Price Movements: Key Factors That Drive Volatility https://www.osl.com/hk-en/academy/article/cryptocurrency-price-movements-key-factors-that-drive-volatility

13. 7 Key factors that could impact cryptocurrency prices https://www.moomoo.com/sg/learn/detail-7-key-factors-that-could-impact-cryptocurrency-prices-117206-240769245

14. What Affects Crypto Prices? Factors Driving the Crypto Market https://www.tradu.com/en/guide/crypto/what-drives-crypto-prices/

15. The Factors Behind Crypto Price Changes https://swapspace.co/blog/factors-behind-price-changes

16. Bitcoin Halving Explained: What Investors Need to Know [2025] https://www.blockpit.io/blog/bitcoin-halving

17. An Introduction to On-Chain Fundamental Analysis https://www.galaxy.com/insights/research/introduction-on-chain-analysis

18. Guide to Crypto Markets: Q1 2025 https://www.coinbase.com/institutional/research-insights/research/market-intelligence/guide-to-crypto-markets-q1-2025

19. What affects and determines cryptocurrency prices? https://www.bitstore.net/en/blog/what-determines-cryptocurrency-prices/

20. Market Manipulation in Cryptocurrencies: Whales, Schemes, and Tactics https://www.visionfactory.org/post/market-manipulation-in-cryptocurrencies-whales-schemes-and-tactics

21. What is trading volume? Why it matters in crypto markets https://www.cointracker.io/learn/trading-volume

22. Understanding crypto trading volume https://learncrypto.com/knowledge-base/how-to-trade-crypto/understanding-crypto-trading-volume

23. Авторегресійне інтегроване ковзне середнє https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average

24. ARIMA & SARIMA: Real-World Time Series Forecasting https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

25. Vector autoregression https://en.wikipedia.org/wiki/Vector_autoregression

26. Omar Ahmed Al-Zakhali, Adnan M. Abdulazeez. Comparative Analysis of Machine Learning and Deep Learning Models for Bitcoin Price Prediction. The

Indonesian Journal of Computer Science (IJCS) Vol. 13 No. 1 (2024). DOI: https://doi.org/10.33022/ijcs.v13i1.3722

27. Abraham, Jethin; Higdon, Daniel; Nelson, John; and Ibarra, Juan (2018) "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis," *SMU Data Science Review*: Vol. 1: No. 3, Article 1. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss3/1

28. Prajith Krishnan, Rashid K, Rigil Renji, Arun Kumar K, 2023, Cryptocurrency Prediction Using Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 03. https://www.ijert.org/cryptocurrency-prediction-using-machine-learning

29. Random forest https://uk.wikipedia.org/wiki/Random_forest

30. Review of deep learning models for crypto price prediction: implementation and evaluation https://arxiv.org/html/2405.11431v1

31. I. Nasirtafreshi. Forecasting cryptocurrency prices using Recurrent Neural Network and Long Short-term Memory. Data & Knowledge Engineering. Volume 139, May 2022. https://www.sciencedirect.com/science/article/abs/pii/S0169023X22000234

32. Kim, Jongyeop et al. “A Cryptocurrency Prediction Model Using LSTM and GRU Algorithms.” 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD) (2021): 37-44. https://www.semanticscholar.org/paper/A-Cryptocurrency-Prediction-Model-Using-LSTM-and-Kim-Kim/9486e736573645aba23cc19c20d5af27c4648e2d

33. Syed H. Hasan, Syeda Huyam Hasan2, Mohammed Salih Ahmed3 and Syed Hamid Hasan. A Novel Cryptocurrency Prediction Method Using Optimum CNN. Computers, Materials & Continua. DOI:10.32604/cmc.2022.020823 https://www.techscience.com/cmc/v71n1/45389/html

34. Ethereum Price Prediction Employing Large Language Models for Short-term and Few-shot Forecasting https://arxiv.org/html/2503.23190v1

35. Abraham, Jethin; Higdon, Daniel; Nelson, John; and Ibarra, Juan (2018) "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment

Analysis," *SMU Data Science Review*: Vol. 1: No. 3, Article 1. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss3/1

36. Prajith Krishnan, Rashid K, Rigil Renji, Arun Kumar K, 2023, Cryptocurrency Prediction Using Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 03, https://www.ijert.org/cryptocurrency-prediction-using-machine-learning

# ДОДАТКИ

**Додаток А**

Програмний код для реалізації запропонованої LSTM-моделі прогнозування курсу криптовалют. Код написано на Python (рекомендована версія ≥3.8) з використанням pandas, numpy, scikit-learn, matplotlib та tensorflow / keras. Коментарі й підказки українською. Код покриває: завантаження даних (CSV або через yfinance для BTC-USD), попередню обробку, формування ковзних вікон, побудову та навчання LSTM, збереження моделі, оцінювання (MAE, RMSE, MAPE) та побудову графіків результатів.

```
"""
crypto_forecast_lstm.py
Реалізація LSTM-моделі для прогнозування ціни криптовалюти (приклад для BTC-USD).
Дата: YYYY-MM-DD
"""

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error
import math
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
import yfinance as yf  # опціонально для завантаження даних, якщо не маєш CSV
```

```
# ======================
# Налаштування
# ======================
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)

# Гіперпараметри (можна змінювати під час експериментів)
LOOKBACK = 60        # довжина вікна (кількість попередніх днів)
BATCH_SIZE = 32
EPOCHS = 100
LEARNING_RATE = 1e-3
DROPOUT_RATE = 0.2
MODEL_DIR = "models"
os.makedirs(MODEL_DIR, exist_ok=True)



# ======================
# Допоміжні функції
# ======================
def load_data_from_csv(path: str, date_col="Date"):
    """Завантажити дані з CSV (повинен бути стовпець дати та стовпець Close)."""
    df = pd.read_csv(path, parse_dates=[date_col])
    df.sort_values(by=date_col, inplace=True)
    df.reset_index(drop=True, inplace=True)
    return df


def download_data_yfinance(ticker="BTC-USD", period="3y", interval="1d"):
```

```
    """Завантажити історичні дані через yfinance (за замовчуванням BTC-USD, 3 роки)."""
    df = yf.download(ticker, period=period, interval=interval)
    df = df.reset_index()
    df.rename(columns={"Date": "Date"}, inplace=True)
    return df


def preprocess(df, feature_col="Close", scaler=None):
    """
    Очищення, нормалізація та формування послідовностей для LSTM.
    Повертає: X, y, scaler, raw_series
    """
    # Переконаємось, що є необхідні колонки
    if feature_col not in df.columns:
        raise ValueError(f"Колонка {feature_col} відсутня в DataFrame")

    # Відсічемо пропуски
    df = df[[col for col in ["Date", feature_col] if col in df.columns]].copy()
    df.dropna(inplace=True)
    raw_series = df[feature_col].values.reshape(-1, 1)

    # Нормалізація
    if scaler is None:
        scaler = MinMaxScaler(feature_range=(0, 1))
        scaled = scaler.fit_transform(raw_series)
    else:
        scaled = scaler.transform(raw_series)

    # Формування ковзних вікон
    X, y = [], []
```

```
    for i in range(LOOKBACK, len(scaled)):
        X.append(scaled[i - LOOKBACK:i, 0])  # послідовність довжиною LOOKBACK
        y.append(scaled[i, 0])               # наступне значення
    X = np.array(X)
    y = np.array(y)

    # LSTM очікує форму (samples, timesteps, features)
    X = np.reshape(X, (X.shape[0], X.shape[1], 1))
    return X, y, scaler, df



def train_val_test_split(X, y, train_ratio=0.70, val_ratio=0.15):
    n = X.shape[0]
    n_train = int(n * train_ratio)
    n_val = int(n * val_ratio)
    X_train = X[:n_train]
    y_train = y[:n_train]
    X_val = X[n_train:n_train + n_val]
    y_val = y[n_train:n_train + n_val]
    X_test = X[n_train + n_val:]
    y_test = y[n_train + n_val:]
    return X_train, y_train, X_val, y_val, X_test, y_test

# ======================
# Побудова моделі
# ======================
def build_lstm_model(input_shape):
    model = Sequential()
    model.add(LSTM(50, return_sequences=True, input_shape=input_shape))
    model.add(Dropout(DROPOUT_RATE))
```

```
    model.add(LSTM(50, return_sequences=False))
    model.add(Dropout(DROPOUT_RATE))
    model.add(Dense(1))
    optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
    model.compile(optimizer=optimizer, loss="mse")
    return model


# =====================
# Оцінювання
# =====================
def compute_metrics(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    rmse = math.sqrt(mean_squared_error(y_true, y_pred))
    # MAPE: уникати ділення на нуль, додаємо маленьке eps
    eps = 1e-8
    mape = np.mean(np.abs((y_true - y_pred) / (np.abs(y_true) + eps))) * 100
    return {"MAE": mae, "RMSE": rmse, "MAPE (%)": mape}



# =====================
# Основний пайплайн
# =====================
def main(data_source_csv: str = None):
    # 1) Завантаження даних
    if data_source_csv and os.path.exists(data_source_csv):
        df = load_data_from_csv(data_source_csv)
        print(f"Дані завантажені з {data_source_csv}, розмір: {df.shape}")
    else:
        print("Завантаження даних через yfinance (BTC-USD, останні 3 роки)...")
        df = download_data_yfinance(ticker="BTC-USD", period="3y")
```

```
        df.reset_index(inplace=True)
        df.rename(columns={0: "Date"}, inplace=True)
        print(f"Дані завантажені через yfinance, розмір: {df.shape}")

    # Перевірка колонок та вибір 'Close'
    if "Close" not in df.columns:
        raise RuntimeError("Вхідні дані повинні містити колонку 'Close'")

    # 2) Попередня обробка та формування вибірок
    X, y, scaler, df_processed = preprocess(df, feature_col="Close", scaler=None)
    X_train, y_train, X_val, y_val, X_test, y_test = train_val_test_split(X, y)

    print("Розміри наборів:")
    print("X_train:", X_train.shape, "y_train:", y_train.shape)
    print("X_val:", X_val.shape, "y_val:", y_val.shape)
    print("X_test:", X_test.shape, "y_test:", y_test.shape)

    # 3) Побудова моделі
    model = build_lstm_model(input_shape=(X_train.shape[1], X_train.shape[2]))
    model.summary()

    # 4) Навчання з ранньою зупинкою та чекпоінтом
    checkpoint_path = os.path.join(MODEL_DIR, "best_lstm.h5")
    callbacks = [
        EarlyStopping(monitor="val_loss", patience=10, restore_best_weights=True),
        ModelCheckpoint(checkpoint_path, monitor="val_loss", save_best_only=True, verbose=1),
    ]
```

```
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    callbacks=callbacks,
    verbose=2
)

# 5) Оцінювання на тестовій вибірці
y_pred_scaled = model.predict(X_test)
# Де-нормалізуємо
y_test_orig = scaler.inverse_transform(y_test.reshape(-1, 1)).flatten()
y_pred_orig = scaler.inverse_transform(y_pred_scaled).flatten()

metrics = compute_metrics(y_test_orig, y_pred_orig)
print("Оцінка на тестовій вибірці:", metrics)

# 6) Побудова графіків
# 6.1 Графік втрат
plt.figure(figsize=(8, 4))
plt.plot(history.history["loss"], label="train_loss")
plt.plot(history.history["val_loss"], label="val_loss")
plt.title("Динаміка функції втрат під час навчання")
plt.xlabel("Епоха")
plt.ylabel("MSE")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("train_loss.png", dpi=150)
```

```
    plt.close()

    # 6.2 Фактичні vs прогноз
    plt.figure(figsize=(10, 5))
    plt.plot(y_test_orig, label="Фактичні")
    plt.plot(y_pred_orig, label="Прогноз", alpha=0.8)
    plt.title("Фактичні значення vs Прогноз моделі (тестовий набір)")
    plt.xlabel("Індекс")
    plt.ylabel("Ціна")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.savefig("forecast_vs_actual.png", dpi=150)
    plt.close()

    # 6.3 Повернемо невелику таблицю результатів
    results_df = pd.DataFrame({
        "actual": y_test_orig,
        "predicted": y_pred_orig
    })
    results_df.to_csv("predictions_test.csv", index=False)
    print("Збережено: train_loss.png, forecast_vs_actual.png,
predictions_test.csv")

    # Збережемо модель остаточно
    model.save(os.path.join(MODEL_DIR, "final_lstm_model.h5"))
    print(f"Модель збережено в {MODEL_DIR}")

    return model, history, metrics, results_df
```

```
if __name__ == "__main__":
    # Якщо є локальний CSV, вказуйте шлях у аргумент main("data/btc_usd.csv")
    model, history, metrics, results_df = main(data_source_csv=None)
```

**Коментарі та рекомендації:**

1. **Джерело даних:** для реального дослідження рекомендовано зберігати сирі часові ряди (raw data) у форматі CSV/Parquet, фіксувати часові пояси й джерело (Binance, CoinGecko тощо). У коді наведено приклад з yfinance для швидкого старту (зручний для BTC-USD), але для інших криптовалют/інструментів кращий варіант – офіційні API бірж або CoinGecko API.

2. **Ознаки (features):** в прикладі використано лише ціну закриття (Close). Для підвищення якості прогнозу варто додати: Open, High, Low, Volume, технічні індикатори (SMA, EMA, RSI, MACD), а також альтернативні дані (sentiment, on-chain metrics).

3. **Валідація та тестування:** в роботі рекомендовано робити часові (time-series) кросвалідації (walk-forward validation) замість випадкового розбиття для більш коректної оцінки.

4. **Гібридні моделі:** для дослідження можна комбінувати ARIMA (для лінійної частини) та LSTM (для нелінійної), або застосувати ансамблі ML-моделей.

5. **Реплікабельність:** фіксуй версії бібліотек (requirements.txt) і встановлюй seed для відтворюваності експериментів.

6. **Метрики:** додатково можна використовувати directional accuracy (відсоток правильно спрогнозованих напрямків зміни ціни) – важлива метрика для трейдингових застосувань.

**Шаблони Dockerfile та docker-compose.yml** для швидкого розгортання сервісу прогнозування курсу криптовалют (API + модель + опціональна БД/кеш).

## Dockerfile (Python backend + модель LSTM/GRU)

```
# --- Базовий Python образ ---
FROM python:3.10-slim

# Встановлення системних залежностей
RUN apt-get update && apt-get install -y --no-install-recommends \
    build-essential wget curl && \
    rm -rf /var/lib/apt/lists/*

# Створення робочого каталогу
WORKDIR /app

# Копіюємо файли залежностей
COPY requirements.txt .

# Встановлюємо Python-бібліотеки
RUN pip install --no-cache-dir -r requirements.txt

# Копіюємо весь проект
COPY . .

# Відкрити порт для API
EXPOSE 8000

# Запуск FastAPI/Flask
CMD ["python", "app.py"]
```

## Приклад requirements.txt

```
numpy
pandas
matplotlib
seaborn
scikit-learn
tensorflow==2.15
fastapi
uvicorn[standard]
python-multipart
requests
```

## docker-compose.yml (API + Redis + (опціонально) база даних)

```
version: "3.9"

services:
  api:
    build: .
    container_name: crypto_predict_api
    restart: always
    ports:
      - "8000:8000"
    volumes:
      - .:/app
```

```
    depends_on:
      - redis
    environment:
      - REDIS_HOST=redis
      - REDIS_PORT=6379

  redis:
    image: redis:7-alpine
    container_name: crypto_predict_redis
    restart: always
    ports:
      - "6379:6379"
    volumes:
      - redis_data:/data

  # Опціонально: PostgreSQL для зберігання історії прогнозів
  db:
    image: postgres:14
    container_name: crypto_predict_db
    restart: always
    environment:
      - POSTGRES_USER=admin
      - POSTGRES_PASSWORD=admin123
      - POSTGRES_DB=crypto_predict
    ports:
      - "5432:5432"
    volumes:
      - pg_data:/var/lib/postgresql/data

volumes:
  redis_data:
  pg_data:
```

## API-сервіс

- Використовує TensorFlow/LSTM/GRU
- Приймає запити через FastAPI
- Генерує прогноз курсу у JSON

## Redis (кеш)

- Швидке кешування отриманих даних з бірж
- Зменшення навантаження на API

## PostgreSQL (опціонально)

- Зберігання історії прогнозів
- Логи та діагностика моделі

## Команди для запуску

### Зібрати та запустити всі сервіси:

```
docker-compose up -d --build
```

### Переглянути логи API:

```
docker logs -f crypto_predict_api
```

## **Python-скрипт інтеграції з Binance та CoinGecko** з:

- автоматичним повтором запитів при rate-limit
- кешуванням (Redis / fallback: локальний файл)
- уніфікованим інтерфейсом для отримання OHLCV
- логуванням

Скрипт легко вбудовується в FastAPI/Flask або в будь-який ML-проект.

## **binance_coingecko_adapter.py**

```
import time
import json
import logging
import requests
from datetime import datetime, timedelta

try:
    import redis
    REDIS_AVAILABLE = True
except ImportError:
    REDIS_AVAILABLE = False


# ---------------- ЛОГУВАННЯ ----------------

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s [%(levelname)s] %(message)s",
)
logger = logging.getLogger("DataAdapter")


# ---------------- КЕШ У REDIS ----------------

class Cache:
    def __init__(self, ttl=300):
        """ttl — час життя кешу в секундах (за замовчуванням 5 хвилин)."""
        self.ttl = ttl

        if REDIS_AVAILABLE:
            try:
                self.redis = redis.Redis(host="redis", port=6379, db=0)
                self.redis.ping()
                logger.info("Redis cache connected.")
            except Exception:
                logger.warning("Redis connection failed, using file cache.")
                self.redis = None
        else:
            self.redis = None

    def get(self, key: str):
        if self.redis:
            data = self.redis.get(key)
            return json.loads(data) if data else None

        # fallback: локальний файл
        try:
            with open(f"cache_{key}.json", "r") as f:
```

```
                obj = json.load(f)
                if obj["expires"] > time.time():
                    return obj["data"]
        except FileNotFoundError:
            return None

    def set(self, key: str, data):
        if self.redis:
            self.redis.setex(key, self.ttl, json.dumps(data))
            return

        # fallback: файл
        obj = {
            "expires": time.time() + self.ttl,
            "data": data
        }
        with open(f"cache_{key}.json", "w") as f:
            json.dump(obj, f)


# ----------------- АДАПТЕР --------------------

class CryptoDataAdapter:
    BINANCE_URL = "https://api.binance.com/api/v3/klines"
    COINGECKO_URL = "https://api.coingecko.com/api/v3/coins/{coin}/market_chart"

    def __init__(self, cache_ttl=300):
        self.cache = Cache(ttl=cache_ttl)

    # ---------- ЗАГАЛЬНИЙ ЗАПИТ З RATE LIMIT ----------
    def _request_with_retry(self, url, params, retries=5):
        for attempt in range(retries):
            try:
                r = requests.get(url, params=params, timeout=10)

                # Binance: 429 — rate limit
                if r.status_code == 429:
                    wait = 2 ** attempt
                    logger.warning(f"Rate limit hit. Retrying in {wait}s...")
                    time.sleep(wait)
                    continue

                r.raise_for_status()
                return r.json()

            except Exception as e:
                logger.error(f"Request failed ({e}). Retrying...")
                time.sleep(2 ** attempt)

        raise RuntimeError("Failed after max retries")

    # ---------------- BINANCE OHLCV ----------------
    def get_binance_ohlcv(self, symbol="BTCUSDT", interval="1d", limit=365):
        cache_key = f"binance_{symbol}_{interval}_{limit}"
        cached = self.cache.get(cache_key)

        if cached:
            logger.info("Loaded from cache (Binance).")
            return cached

        params = dict(symbol=symbol, interval=interval, limit=limit)
        raw = self._request_with_retry(self.BINANCE_URL, params)

        ohlcv = [
            {
```

```
            "timestamp": int(item[0]),
            "open": float(item[1]),
            "high": float(item[2]),
            "low": float(item[3]),
            "close": float(item[4]),
            "volume": float(item[5])
        }
        for item in raw
    ]

    self.cache.set(cache_key, ohlcv)
    return ohlcv

# ---------------- COINGECKO OHLCV ----------------
def get_coingecko_ohlcv(self, coin_id="bitcoin", days=365):
    cache_key = f"coingecko_{coin_id}_{days}"
    cached = self.cache.get(cache_key)

    if cached:
        logger.info("Loaded from cache (CoinGecko).")
        return cached

    url = self.COINGECKO_URL.format(coin=coin_id)
    params = dict(vs_currency="usd", days=days)

    raw = self._request_with_retry(url, params)

    # відповіді CoinGecko містять price, market_caps, total_volumes
    ohlcv = []

    prices = raw.get("prices", [])
    for p in prices:
        ohlcv.append({
            "timestamp": int(p[0]),
            "close": float(p[1])
        })

    self.cache.set(cache_key, ohlcv)
    return ohlcv

# ---------------- УНІФІКОВАНИЙ ІНТЕРФЕЙС ----------------
def get_ohlcv(self, source="binance", **kwargs):
    if source == "binance":
        return self.get_binance_ohlcv(**kwargs)
    elif source == "coingecko":
        return self.get_coingecko_ohlcv(**kwargs)
    else:
        raise ValueError("Source must be 'binance' or 'coingecko'")
```