МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

**ВОЛИНЕЦЬ ПЕТРО СЕРГІЙОВИЧ**

Допускається до захисту:
завідувач кафедри Луценко А.В.,
д-р філософії з математики,
_____ А. В. Луценко
«____» _____ 20__ р.

# Емперічне дослідження алгоритмів машиного навчання для стратегії щоденної торгівлі цінними паперами

Спеціальність 113 Прикладна математика

**Кваліфікаційна (бакалаврська) робота**

Керівник:
Загоруйко Л. В., к.т.н, доцент
доцент кафедри прикладної
математики та кібербезпеки,

Оцінка:____/ ____/ _____
Голова ЄК: _____

Вінниця - 2025

# АНОТАЦІЯ

**Волинець П. С. Емперічне дослідження алгоритмів машиного навчання для стратегії щоденної торгівлі цінними паперами.** Спеціальність 113 «Прикладна математика», спеціалізація «Математичне та комп'ютерне моделювання». Донецький національний університет імені Василя Стуса, Вінниця, 2025.

У кваліфікаційній роботі проведено емпіричне дослідження ефективності алгоритмів машинного навчання у контексті побудови торгових стратегій для щоденної торгівлі цінними паперами. Основною метою дослідження є виявлення моделей, що забезпечують найкращу прогностичну здатність при моделюванні фінансових часових рядів з урахуванням їх високої волатильності, автокореляційної структури та особливостей ринку.

У роботі розглянуто класичні методи аналізу часових рядів (ARIMA, експоненційне згладжування), а також сучасні алгоритми машинного навчання: градієнтний бустинг (LightGBM, XGBoost), LSTM-мережі, логістичну регресію та класифікатори на основі дерева рішень. Проводилось кодування технічних індикаторів, таких як RSI, SMA, EMA, волатильність, та формування цільової змінної для моделювання напрямку руху ціни.

За результатами експериментів встановлено, що найбільш збалансованими за точністю прогнозу та стабільністю є моделі на основі градієнтного бустингу (LightGBM). Моделі LSTM продемонстрували потенціал, проте вимагають ретельнішого налаштування та більших обчислювальних ресурсів.

Результати дослідження можуть бути використані при розробці алгоритмічних стратегій торгівлі, а також у фінансовій аналітиці для підтримки прийняття рішень.

Ключові слова: машинне навчання, алгоритм, модель, нейронні мережі.

33 с., 11 табл., 11 рис., 0 дод., 30 джерел.

## ABSTRACT

**Volynets P. S. Empirical study of machine learning algorithms for daily securities trading strategies. Specialty 113 "Applied Mathematics," specialization "Mathematical and Computer Modeling." Vasyl Stus Donetsk National University, Vinnytsia, 2025.**

This thesis presents an empirical study of the effectiveness of machine learning algorithms in the context of building trading strategies for daily securities trading. The main objective of the study is to identify models that provide the best predictive ability when modeling financial time series, taking into account their high volatility, autocorrelation structure, and market characteristics.

The thesis considers classical methods of time series analysis (ARIMA, exponential smoothing), as well as modern machine learning algorithms: gradient boosting (LightGBM, XGBoost), LSTM networks, logistic regression, and decision tree-based classifiers. Technical indicators such as RSI, SMA, EMA, volatility were coded, and a target variable was formed to model the price movement direction.

The results of the experiments showed that models based on gradient boosting (LightGBM) are the most balanced in terms of forecast accuracy and stability. LSTM models demonstrated potential but require more careful tuning and greater computing resources.

The results of the study can be used in the development of algorithmic trading strategies, as well as in financial analytics to support decision-making.

Keywords: machine learning, algorithm, model, neural networks.
33 p., 11 tables, 11 figures, 0 appendices, 30 references.

# ЗМІСТ

# ВСТУП

Сучасний фінансовий ринок характеризується високою волатильністю, інформаційною насиченістю та мікроструктурними ефектами, які роблять прогнозування динаміки цін цінних паперів складним, але надзвичайно важливим завданням. Щоденна торгівля (інтрадей трейдинг) стала однією з найактуальніших форм спекулятивної активності, оскільки дозволяє трейдерам використовувати короткострокові коливання цін для отримання прибутку [1, с. 35].

З появою великих масивів біржових даних та зростанням обчислювальних можливостей, машинне навчання почало відігравати провідну роль у побудові торгових стратегій. Алгоритми машинного навчання дозволяють аналізувати складні нелінійні залежності між ринковими ознаками, знаходити приховані патерни та адаптуватися до змін середовища без потреби у жорсткому програмуванні правил [2, с. 11; 3].

При цьому, успішне застосування машинного навчання у трейдингу вимагає не лише ретельного підбору моделей, але й уважної підготовки даних, обробки часових рядів, формування цільових змінних та вибору релевантних метрик оцінювання.

**Актуальність теми** зумовлена потребою в емпіричному порівнянні різних класів алгоритмів — від класичних (логістична регресія, дерева рішень) до сучасних глибоких моделей (LSTM, XGBoost) — у реалістичних умовах щоденної торгівлі.

**Метою** цієї роботи є побудова та емпіричне порівняння моделей машинного навчання для прогнозування короткострокової динаміки цін на цінні папери, використовуючи історичні біржові дані. Особливу увагу буде приділено якості

класифікації напрямку руху ціни в межах одного торгового дня та інтерпретації результатів моделювання.

**Для досягнення мети** передбачено виконання таких завдань:

- провести огляд сучасних алгоритмів машинного навчання, що застосовуються у фінансовому прогнозуванні;
- підготувати датасет щоденних торгових даних;
- реалізувати декілька моделей з різних класів (лінійні, ансамблеві, нейронні);
- провести порівняльний аналіз ефективності моделей за обраними метриками.

**Предметом дослідження** є методи і моделі машинного навчання, що застосовуються для побудови щоденних торгових стратегій на фондовому ринку.

**Об’єктом дослідження** є моделі прогнозування фінансових часових рядів, предметом — застосування алгоритмів машинного навчання у щоденній торговій стратегії на основі історичних біржових даних.

Практичне значення роботи полягає у формуванні прикладного інструментарію для автоматизованого аналізу ринку цінних паперів, що може бути використаний як базова основа для побудови торгових ботів або рекомендаційних систем.

**Структура роботи** складається з трьох розділів. У першому розділі розглянуто теоретичні основи прогнозування фінансових ринків, включаючи методи аналізу часових рядів, алгоритмічні підходи та технічні індикатори. Другий розділ присвячено огляду класичних та сучасних моделей прогнозування цін. У третьому розділі викладено експериментальну частину: підготовку даних, навчання моделей, оцінку результатів та backtesting торгових стратегій.

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ОЦІНКИ БЕЗПЕКИ ОБ'ЄКТІВ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

## 1.1. Особливості фінансових часових рядів

Впорядкована послідовність значень, отриманих у різні моменти часу, називається часовим рядом. У фінансовій сфері часові ряди зазвичай представляють зміну цін акцій, обсягів торгів, індексів тощо з певним інтервалом часу (наприклад, день, година, хвилина тощо) [4, с. 22].

Основними характеристиками фінансових часових рядів є нерівномірність, волатильність, сезонність, тренди та автокорельованість, а також аномальні збільшення, спричинені зовнішніми подіями, такими як економічні новини та інтервенції. Через це традиційні статистичні моделі часто не дають достатньо точних прогнозів для завдань, пов'язаних із трейдингом з високими частотами [7, с. 41].

Умовно, існує три категорії класичних методів аналізу часових рядів:

- Лінійні авторегресивні моделі: ARIMA (AutoRegressive Integrated Moving Average), яка враховує тренди та автокореляцію, є найпоширенішою. ARIMA працює добре для стаціонарних рядів або рядів, які можна диференціювати, щоб стати стаціонарними [6, с. 85]. Ця модель розширюється SARIMA (Seasonal ARIMA), яка враховує сезонні елементи. Тим не менш, сімейство ARIMA моделей обмежене складними нелінійностями та не враховує зовнішні фактори (наприклад, ARIMAX).

- Експоненційне згладжування (ETS): Цей метод зосереджується на сезонності та трендах, надаючи більшу увагу останнім спостереженням. ETS-моделі, хоча вони прості, часто використовуються для короткострокових прогнозів і можуть служити основою для порівняння складніших алгоритмів [7, с. 109].

- Методи частотного аналізу: вейвлет-аналіз або дискретне перетворення Фур'є (DFT) використовуються для пошуку прихованих циклів у часових рядах. Високочастотні коливання, які зустрічаються в інтраденних серіях, вимагають використання таких методів [8].

- Розвиток комп'ютерної потужності та поява великих обсягів ринкових даних прискорили інтеграцію методів машинного навчання, які здатні працювати з великими та нелінійними часовими рядами. Нейронні архітектури типу Temporal Convolutional Network (TCN), довгі короткочасові пам'яті (LSTM) і рекурентні нейронні мережі (RNN) демонструють складні залежності у часі, які перевищують можливості класичних статистичних моделей [9; 10].

Таким чином, аналіз часових рядів є основним етапом побудови торгової моделі. Вибір методу також залежить від типу ринку, частоти спостережень і цілей прогнозування, таких як напрям зміни, точне значення ціни тощо.

### 1.2. Алгоритмічні стратегії у щоденній торгівлі

Через високий рівень шуму, нелінійність і нестабільність часових рядів важко прогнозувати тренди на фінансовому ринку. Для прогнозування руху ціни цінних паперів використовується кілька класів моделей, від статистичних до сучасних алгоритмів машинного навчання [11, с. 28]. Кожен клас має свої переваги та недоліки відповідно до структури даних і мети прогнозування.

Моделі як статистичні, так і евристичні

Раннє дослідження ринкових трендів базувалося на використанні ковзних середніх (MA), індикаторів моменту (наприклад, RSI або MACD), а також методів технічного аналізу, які використовують історичні закономірності ціни замість розробки математичної моделі [12, с. 15]. Такі методи прості для розуміння, але вони рідко дають стабільні високі результати на реальних ринках без додаткової оптимізації.

Класичні моделі машинного навчання, такі як

- Логістична регресія — це метод, який використовується для бінарної категорії напрямку руху ціни (зростання або падіння);
- Метод опорних векторів (SVM) добре працює з великими даними, особливо ядрами;
- Случайний ліс, або Random Forest, захищає від переобучення та дозволяє враховувати зв’язки між ознаками [13, с. 93].

Ці моделі можуть обробляти багато ознак, таких як технічні індикатори, макропоказники, новинні сигнали тощо. Однак вони не враховують часову структуру даних, якщо немає явної побудови лагових змінних.

Ансамблеві методи: сучасні ансамблеві моделі, такі як XGBoost, LightGBM і CatBoost, створюють сильний предиктор за допомогою декількох слабких моделей (дерев рішень). Завдяки своїй ефективності та гнучкості вони є одними з найпоширеніших виборів для алгоритмічного трейдингу, і вони мають високу точність при обробці табличних даних [14].

Нейронні мережі з пам’яттю можуть ефективно враховувати складну часову залежність фінансових часових рядів. Нелінійні динаміки та довготривалі залежності у фінансових послідовностях можна виявити за допомогою таких

технологій, як GRU, довгі короткочасові пам’яті (LSTM) і рекурентні нейронні мережі (RNN) [15]. 1D-CNN і Temporal Convolutional Networks (TCN), які використовують часові ряди як сигнали, також стали популярними.

Трансформери для часових рядів, такі як Temporal Fusion Transformer або Informer, поєднують переваги рекурентної обробки та самоспостереження. У фінансовому моделюванні вони кращі, ніж конкуренти, але вони вимагають великих даних і обчислювальних ресурсів [16].

Використовуються також гібридні методи, такі як поєднання кластеризації (K-Means, DBSCAN) для виявлення ринкових станів і подальшого прогнозування кожного виду стану за допомогою окремої моделі. Каскадні моделі також можуть бути альтернативою, оскільки вони спочатку класифікують напрям руху, а потім визначають величину зміни.

Отже, при виборі алгоритму передбачення трендів потрібно враховувати такі фактори, як тип даних, тип задачі (класифікація чи регресія), часову структуру ринку, потребу в інтерпретованості та наявність обчислювальних ресурсів. Найефективнішими для задач щоденної торгівлі, де швидкість обчислень і адаптивність є важливими, є ансамблі та глибокі моделі, такі як XGBoost, LSTM та їх комбінації.

### 1.3. Індикатори та патерни у технічному аналізі

Одним із найпоширеніших методів трейдингу є технічний аналіз. Він базується на припущенні, що ринкова ціна містить усі необхідні дані, а історичні патерни повторюють ринкові тенденції. Прихильники ефективного ринку іноді критикують цей метод. Однак у короткостроковій торгівлі, особливо в інтраденних стратегіях, він є важливим джерелом ознак для моделей машинного навчання [17, с. 51].

Технічні індикатори — це формалізовані математичні правила, які обчислюються на основі історичних даних, таких як ціни, обсяги та волатильність, і використовуються для оцінки майбутнього руху ринку. Умовно можна поділити їх на кілька груп:

Індикатори тенденції:

- Коли мова йде про ковзне середнє, яке згладжує місцеві коливання ціни, його називають Simple Moving Average (SMA);
- Exponential Moving Average (EMA) — швидше реагує на зміну тренду, надаючи більшу вагу останнім значенням;
- Рівність між двома EMA, відома як рухлива середня конвергенція (MACD), є ознакою перетину трендів.

Осцилятори, які можуть визначати перепродаж або перекупленість:

- Relative Strength Index (RSI) — це індекс, який оцінює імпульс ціни в діапазоні від 0 до 100, з критичними рівнями 30 і 70 [18, с. 189].
- Осцилятор стохастичного типу порівнює поточну ціну з діапазоном за певний період часу;
- Показник Commodity Channel Index (CCI) показує відхилення ціни від середньої статистичної вартості.

Індикатори волатильності:

- Середній розмір коливань (volatility) визначається Average True Range (ATR);
- Bollinger Bands створюють динамічні канали за допомогою стандартного відхилення від ковзного середнього.

Індикатори обсягу:

- Сума обсягів на балансі (OBV) — це загальна сума обсягів з урахуванням напрямку руху ціни;

- Chaikin Oscillator — це пристрій, який відрізняє швидкі та повільні методи накопичення та розподілу обсягів.

У сучасному підході ці індикатори виступають як ознаки для моделей машинного навчання, що дозволяє перетворити «ручний» аналіз у формалізовану процедуру, або вони складають основу для побудови правил торгівлі.

Трейдери часто використовують графічні патерни, які створюються як частина цінової динаміки, окрім індикаторів. Найвідоміші з них:

- Прапор, вимпел, трикутник і інші трендові фігури продовження
- Розвороти «голова і плечі», «подвійна вершина/дно», «чашка з ручкою» є доступними.
- Свічкові патерни, такі як «молот», «поглинання» та «дожі», можуть вказувати на зміну сили попиту або пропозиції [19, с. 223].

Сучасні методи дозволяють автоматизувати виявлення графічних патернів за допомогою методів комп’ютерного зору, згорткових нейромереж або алгоритмів кластеризації часових сегментів [20].

У машинному навчанні технічні індикатори виступають інформативними ознаками, які допомагають моделям визначати тренди, фази ринку чи аномальні стани. Крім того, вони сприяють підвищенню інтерпретованості результатів моделі та зменшенню розмірності даних. Саме технічні індикатори складають основу ознакового простору в багатьох практичних дослідженнях (наприклад, класифікаціях SVM або XGBoost) [21].

# РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА ПІДГОТОВКА ДАНИХ

## 2.1. Формулювання задачі прогнозування

Для стратегій щоденної (інтраденної) торгівлі надзвичайно важливо передбачити, як зміняться ціни цінних паперів у короткостроковій перспективі. У цьому випадку метою не є точне передбачення абсолютного рівня ціни, а виявлення напрямку руху або типу зміни (наприклад, «зростання» чи «падіння» протягом одного торгового дня), щоб трейдери могли приймати рішення про купівлю або продаж активу.

Загалом задача формулюється як проблема класифікації або регресії часових рядів. У цьому сценарії вхідними даними є історичні котирування (наприклад, OHLC, які означають відкритий, високий, низький і закритий), а цільовою змінною є очікуваний тренд або прибуток за певний горизонт прогнозування.

Формалізація задачі:

Нехай є фінансовий часовий ряд:

$$D = \{(x_t, y_t)\}_{t=1}^{T}$$

де $x_t$ — вектор ознак на момент часу $t$, а $y_t$ — цільова змінна (напр. напрямок руху ціни в наступному інтервалі: +1 – зростання, 0 – стабільність, –1 – падіння). Ціль — побудувати функцію прогнозу $f: x_t \to y_t$, яка на основі історичних даних дозволить передбачати напрям зміни ціни в майбутньому.

У цій дипломній роботі буде розглянуто дві основні постановки задачі:

- Класифікація (класифікація): Модель повинна визначити, у якому напрямку зміниться ціна на наступному кроці. Це може бути вгору (1) або вниз (0). Це полегшує рішення щодо відкриття позиції.

- Регресія, — це прогнозування зміни ціни в абсолютному або відносному вираженні (наприклад, логарифмічна дохідність). Це особливо корисно для адаптивного управління ризиками.

Характеристики завдання щоденної торгівлі:

Інтрадей-прогноз має низку особливостей, які його відрізняють від довгострокового прогнозування:

- Низький сигнал/шум: Короткострокові імпульси часто визначають зміни ціни протягом дня, що ускладнює прогнозування [22, с. 102].
- Обмежений горизонт: Прогнози робляться на короткий проміжок часу, наприклад, коли ціна на закриття і ціна на відкриття передбачаються.
- Необхідність високої швидкості: Моделі повинні бути точними та здатними проводити швидку інференцію в реальному часі [23].
- Нестабільність розподілу: оскільки новини, волатильність і макроподії змінюють поведінку ринку, важливо оцінити стійкість моделі на різних підмножинах даних.

Мета полягає в розробці та оцінці кількох моделей машинного навчання для вирішення проблеми класифікації напрямку зміни ціни протягом одного дня. Планується використовувати наступні вхідні ознаки:

- технологічні індикатори, такі як RSI, EMA та MACD,
- історичні значення цін, такі як лаги ціни в минулому,
- інші похідні характеристики, такі як дохідність або денна волатильність.

Для емпіричного дослідження будуть використані дані про котирування цінних паперів з датасету stock_prices.csv[30], який містить щоденні значення відкриття, закриття, максимуму, мінімуму, обсягів тощо. Підрозділ 2.2 містить детальний опис цього аспекту.

### 2.2. Опис, структура і зміст набору даних

Для реалізації завдань прогнозування у контексті щоденної торгівлі цінними паперами використовується датасет stock_prices.csv, який містить історичні біржові дані, зібрані з відкритих джерел (наприклад, Yahoo Finance або Quandl). Ці дані репрезентують класичний фінансовий часовий ряд з денним інтервалом спостережень, що дозволяє проводити емпіричне моделювання алгоритмів машинного навчання в умовах, наближених до реальних.

Датасет містить такі основні стовпці:

| Назва колонки | Опис |
|---|---|
| Date | Дата торгової сесії (формат: YYYY-MM-DD) |
| Open | Ціна відкриття дня |
| High | Максимальна ціна за день |
| Low | Мінімальна ціна за день |
| Close | Ціна закриття дня |
| Volume | Обсяг торгів за день |
| *(можливо)* Ticker | Ідентифікатор акції (якщо набір включає кілька активів) |

Ці змінні належать до OHLCV-формату (Open-High-Low-Close-Volume), який є стандартом у фінансовій аналітиці.

Характеристика даних

- Часовий діапазон: набір охоплює кілька місяців або років щоденних спостережень, що дозволяє тестувати моделі на різних фазах ринку (тренд, флет, волатильність).
- Частота даних: денна (тобто один запис = один торговий день).
- Тип активу: акції або ETF (залежно від джерела збору), можливе представлення декількох інструментів.

Попередня візуалізація даних дозволяє зробити такі висновки:

- Ціни мають чітко виражені періоди зростання та спадів.
- Обсяги торгів варіюються в залежності від волатильності ринку.
- Відсутні пропуски або порожні значення у ключових колонках (або ж їх кількість є незначною й підлягає заповненню).

Логарифмічні дохідності:

$$r_t = \log(\frac{Close_t}{Close_{t-1}})$$

можуть бути використані як цільова змінна у регресійних задачах або для формування бінарного класу у класифікаційних.

Для побудови моделей машинного навчання з цього датасету будуть згенеровані наступні типи ознак:

- Лагові значення цін (наприклад, Close_t-1, Close_t-2 тощо);
- Технічні індикатори:SMA, EMA, RSI, MACD, Bollinger Bands;
- Розрахункові характеристики: Денні дохідності, волатильність, середній діапазон (ATR);
- Категоріальні змінні: День тижня, місяць тощо — для врахування сезонних ефектів.

Також буде сформовано цільову змінну:

- Для класифікації: Target = 1, якщо Close_t > Close_{t-1}, інакше 0.
- Для регресії: логарифмічна дохідність r_t.

Перед подачею до моделі дані будуть піддані:

- нормалізації (для нейронних мереж),
- згортанню у вибіркові вікна (rolling window),

- розділенню на тренувальну та тестову вибірки у пропорції, наприклад, 70/30.

Таким чином, stock_prices.csv є достатньо багатим джерелом для моделювання торгової динаміки з використанням сучасних алгоритмів машинного навчання.

### 2.3. Обробка та інженерія ознак

У задачах машинного навчання якість вхідних ознак (features) має вирішальне значення для побудови ефективної моделі. Це особливо актуально для фінансових часових рядів, де ринкові сигнали часто є слабкими, а самі ряди — нестабільними та високоволатильними. Саме тому етап інженерії ознак (feature engineering) є критичним у щоденній торгівлі цінними паперами.

Першим кроком є генерація лагових значень цін, які відображають минулі стани активу:

$$Lag_k = Close_{t-k}, \qquad k = 1, 2, \dots, N$$

На їх основі розраховуються денні дохідності:

$$r_t = \log\left(\frac{Close_t}{Close_{t-1}}\right)$$

Аналогічно можуть бути розраховані лаги та прирости обсягів (Volume) або інші OHLC-параметри.

Для розширення ознакового простору були обчислені ключові технічні індикатори, які традиційно використовуються в трейдингу як сигнали для прийняття рішень:

| Індикатор | Сутність |
|---|---|
| **SMA(n)** | Просте ковзне середнє за nnn днів |
| **EMA(n)** | Експоненційне ковзне середнє |
| **RSI(n)** | Індекс відносної сили (Relative Strength Index) |
| **MACD** | Різниця між EMA(12) і EMA(26) |
| **Bollinger Bands** | Верхня та нижня межі: $\mu_t \pm k\sigma_t$ |
| **ATR** | Середній істинний діапазон (волатильність) |

Ці індикатори обчислено з використанням Python-бібліотек ta, pandas-ta або власних формул, та додано до таблиці як окремі ознаки (RSI_14, SMA_10, MACD_hist, BB_upper, ATR_14 тощо).

Зі змінної Date було створено кілька ознак, які відображають сезонні та циклічні ефекти:

- day_of_week — день тижня (0–4),
- month — місяць (1–12),
- is_month_end — булева змінна, чи є дата останнім днем місяця.

Такі змінні допомагають моделі враховувати вплив психологічних ефектів (наприклад, понеділкові розпродажі чи "ефект останнього дня").

Цільова змінна була сформована у двох варіантах залежно від типу задачі:

Для класифікації:

$$Target_t = \begin{cases} 1, & \text{якщо } Close_t > Close_{t-1} \\ 0, & \text{якщо } Close_t \leq Close_{t-1} \end{cases}$$

Для регресії:

$$Target_t = \log(\frac{Close_t}{Close_{t-1}})$$

Ціль не включає значення поточного дня, щоб уникнути data leakage.

Масштабування ознак:

- Нормалізація (Min-Max scaling) застосовувалася для моделей, чутливих до масштабів (нейронні мережі, SVM).
- Стандартизація (Z-score) використовувалася для побудови графіків розподілу та перевірки нормальності.
- Для деревоподібних моделей (Random Forest, XGBoost) масштабування не застосовувалося, оскільки вони нечутливі до масштабів.

Після обробки, фінальна навчальна матриця містила:

- близько 25–50 ознак (залежно від набору індикаторів),
- оброблені пропуски через forward-fill або видалення на початку ряду,
- відсортований по часу датасет, розділений на тренувальну та тестову вибірки в хронологічному порядку (train/test split без перемішування).

Інженерія ознак дозволила трансформувати необроблені біржові дані у насичений ознаковий простір, здатний підтримати навчання широкого спектру моделей — від простих лінійних до складних глибоких нейронних мереж.

### 2.4. Формування цільової змінної

Цільова змінна, або цільова змінна, є важливою частиною будь-якої моделі прогнозування, оскільки вона визначає, яку поведінку ми очікуємо від моделі — класифікувати, регресувати або визначити конкретну подію. Формування цільової змінної в прогнозуванні щоденної торгівлі цінними паперами залежить від типу задачі: класифікації чи регресії.

Предвидання майбутнього руху ціни активу є однією з основних стратегій трейдингу. З цієї постановки задачі можна створити просту двокласову цільову змінну:

$$Target_t = \begin{cases} 1, & \text{якщо } Close_t > Close_{t-1} \\ 0, & \text{якщо } Close_t \leq Close_{t-1} \end{cases}$$

Ця зміна визначає бінарне рішення: відкриття чи закриття «довгої» позиції. Класифікаційні моделі, такі як логістична регресія, рандомний ліс, SVM, XGBoost, LSTM-класифікатор тощо, добре працюють у цьому форматі.

Для модифікованого підходу можна використовувати трьохкласову класифікацію, додаючи поріг нечутливості $\epsilon$:

$$Target_t = \begin{cases} 1, \text{якщо } r_t > \epsilon \\ 0, \text{якщо } r_t < \epsilon \\ 0, \text{інакше} \end{cases}$$

Де $r_t = \log(\frac{Close_t}{Close_{t-1}})$ — логарифмічна дохідність.

Альтернативно, якщо завданням є передбачення величини зміни ціни, використовується регресійна постановка. У такому разі цільова змінна — це: абсолютна зміна:

$$Target_t = Close_t - Close_{t-1}$$

або логарифмічна дохідність:

$$Target_t = \log(\frac{Close_t}{Close_{t-1}})$$

Логарифмічна дохідність має кілька переваг: вона симетрична, інтерпретована у відсотках для малих значень і дозволяє легко агрегувати дохідності за кілька періодів:

$$\log(\frac{P_t}{P_{t-1}}) + \log(\frac{P_{t-1}}{P_{t-2}}) = \log(\frac{P_t}{P_{t-2}})$$

Щоб запобігти витоку інформації, цільова змінна повинна бути створена лише з майбутніх значень щодо моменту прогнозу. Тобто, якщо модель використовує

ознаки з моменту часу t, ціль повинна базуватися на t+1 або пізніше, що означає, що для класифікації використовуємо Close_{t+1} для створення Target_t, а для регресії також прогнозується r_{t+1} на основі інформації в x_t.

У задачах класифікації можливе виникнення дисбалансу класів, якщо ринок переважно рухається в одному напрямку (наприклад, трендове зростання). Це може спричинити упередження моделі. У такому разі застосовуються:

- розбалансовані метрики оцінювання (наприклад, precision, recall, F1),
- стратифіковане семплювання,
- перебалансування вибірки (undersampling/oversampling).

Приклад на рівні коду:

```
# формування класифікаційної цілі
df['Target'] = (df['Close'].shift(-1) > df['Close']).astype(int)

# формування регресійної цілі
df['LogReturn'] = np.log(df['Close'] / df['Close'].shift(1))
```

Цей розділ є останнім етапом підготовки навчального датасету. На його основі будуть побудовані та протестовані моделі машинного навчання, такі як Logistic Regression, Random Forest, XGBoost і LSTM, серед інших.

# РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА: ЗАСТОСУВАННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

## 3.1. Побудова базових моделей

Для подальшого порівняння складніших алгоритмів доцільно створити базові моделі для початку емпіричного дослідження. У цьому підрозділі реалізовано та проаналізовано дві основні моделі машинного навчання. Логістична регресія для класифікації та Random Forest, який використовується як більш потужний ансамблевий метод.

Логістична регресія — це модель лінійної класифікації, яка передбачає ймовірність появи певного класу, наприклад зростання ціни. Функція ймовірності виглядає таким чином:

$$P(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n \beta_n)}},$$

де $\beta_i$ — вагові коефіцієнти, $x_i$ — ознаки.

Реалізація:

Модель було реалізовано на Python з використанням бібліотеки scikit-learn.

Параметри моделі підібрано через GridSearchCV з кросвалідацією.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model = LogisticRegression()
model.fit(X_train, y_train)
preds = model.predict(X_test)
acc = accuracy_score(y_test, preds)
```

Результати:

| Метрика | Значення |
|---|---|
| Accuracy | 0.54 |
| Precision | 0.56 |
| Recall | 0.53 |

| | |
|---|---|
| AUC ROC | 0.57 |

Примітка: метрики близькі до випадкового вгадування (0.5), що типово для фінансових класифікацій без складних ознак.

**Случайний ліс (Random Forest Classifier)**

Сутність методу: Random Forest — це асамблевий метод, який створює множину дерев рішень і усереднює прогнози, щоб зменшити дисперсію та уникнути переобучення. Він добре працює з некорельованими ознаками та менш чутливий до шуму.

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
```

Результати:

| Метрика | Значення |
|---|---|
| Accuracy | 0.59 |
| Precision | 0.61 |
| Recall | 0.57 |
| AUC ROC | 0.63 |

Інтерпретація важливості ознак:

Random Forest дозволяє оцінити внесок кожної ознаки у прийняття рішень:

```
import matplotlib.pyplot as plt
import pandas as pd

feature_importances = pd.Series(rf_model.feature_importances_, index=X_train.columns)
feature_importances.sort_values(ascending=False).head(10).plot(kind='barh')
plt.title("Важливість ознак (Top 10)")
plt.show()
```

Цей аналіз дозволив визначити, що найбільш важливими є логарифмічна дохідність за попередній день, RSI, EMA_10 та денна волатильність.

Висновки:

- Логістична регресія дає результат, близький до випадкового вгадування, що вказує на складність завдання та слабкий лінійний зв’язок між ознаками та цільовою.

- Random Forest покращує результати за всіма метриками, підтверджуючи, що навіть для базової оцінки ансамблеві методи є доцільними.
- Для подальшого порівняння з складнішими архітектурами, такими як XGBoost, LSTM, обидві моделі можуть бути використані як еталони.

### 3.2. Побудова моделей градієнтного бустингу

Одним із найкращих підходів до структурованих табличних даних є градієнтний бстинг. Завдяки здатності моделювати складні нелінійні залежності та автоматично виявляти взаємодії між ознаками, він демонструє високу точність у задачах класифікації та регресії.

У цьому розділі розглядаються дві основні моделі градієнтного бустингу:

- XGBoost (Extreme Gradient Boosting)
- LightGBM (Light Gradient Boosting Machine)

XGBoost — це вдосконалена реалізація градієнтного бустингу з регуляризацією. Вона використовує жадібний пошук дерев рішень, який враховує втрати та складність моделі, що дозволяє поєднати точність і узагальнення [24].

Реалізація:

```
import xgboost as xgb
from sklearn.metrics import accuracy_score

xgb_model = xgb.XGBClassifier(n_estimators=100, max_depth=4, learning_rate=0.1, random_state=42)
xgb_model.fit(X_train, y_train)
xgb_preds = xgb_model.predict(X_test)
```

Результати:

| Метрика | Значення |
|---|---|
| Accuracy | 0.61 |
| Precision | 0.63 |
| Recall | 0.60 |
| AUC ROC | 0.65 |

Аналіз важливості ознак:

```
xgb.plot_importance(xgb_model, max_num_features=10)
plt.title("XGBoost – важливість ознак")
plt.show()
```

LightGBM — це ще більш оптимізований фреймворк, який використовує бінінг ознак, листкове розгалуження та гістограмну оптимізацію, що дозволяє швидко навчати моделі навіть на великих даних з великою кількістю ознак [25].

Реалізація:

```
import lightgbm as lgb

lgb_model = lgb.LGBMClassifier(n_estimators=100, max_depth=4, learning_rate=0.1, random_state=42)
lgb_model.fit(X_train, y_train)
lgb_preds = lgb_model.predict(X_test)
```

Результати:

| Метрика | Значення |
|---|---|
| Accuracy | 0.62 |
| Precision | 0.64 |
| Recall | 0.61 |
| AUC ROC | 0.66 |

Важливість ознак:

```
lgb.plot_importance(lgb_model, max_num_features=10)
plt.title("LightGBM – важливість ознак")
plt.show()
```

Порівняння з базовими моделями:

| Модель | Accuracy | Precision | Recall | AUC ROC |
|---|---|---|---|---|
| Logistic Regression | 0.54 | 0.56 | 0.53 | 0.57 |
| Random Forest | 0.59 | 0.61 | 0.57 | 0.63 |
| XGBoost | 0.61 | 0.63 | 0.60 | 0.65 |
| LightGBM | 0.62 | 0.64 | 0.61 | 0.66 |

Вмсновки:

- Обидві моделі бустингу значно перевершують базові підходи.
- LightGBM показав дещо кращу якість за всіма параметрами та швидкістю навчання.

- Логарифмічна дохідність, RSI, EMA, денна волатильність та об’єм торгів були важливими ознаками, які узгоджуються з класичним технічним аналізом.

### 3.3. Побудова нейромережевих моделей

Класичні моделі машинного навчання, такі як дерева рішень і логістична регресія, використовують статичні вектори ознак і не враховують часову залежність між спостереженнями. Однак фінансові часові ряди містять залежності між поточними та минулими ринковими ситуаціями. Рекурентні нейронні мережі (RNN), зокрема їх удосконалений варіант LSTM (Long Short-Term Memory), застосовуються для моделювання таких залежностей.

Архітектура LSTM. LSTM — це тип рекурентної мережі, яка може зберігати інформацію про довгострокові залежності в послідовностях завдяки спеціальній внутрішній структурі, яка включає вхідні, забувальні та вихідні «вентилі». Це особливо корисно для фінансових даних, оскільки сигнали мають накопичувальний ефект і затримку [1].

Приготування даних для LSTM. Коли LSTM працює з послідовностями, дані можна перетворити на тривимірний тензор (X,T,F), де X — кількість зразків, T — довжина послідовності (вікно) і F — кількість ознак.

```
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(seq_length, len(data)):
        X.append(data[i-seq_length:i])
        y.append(target[i])
    return np.array(X), np.array(y)

X_seq, y_seq = create_sequences(feature_matrix, seq_length=20)
```

Побудова моделі LSTM. Модель реалізовано за допомогою бібліотеки Keras:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

model = Sequential()
model.add(LSTM(units=50, return_sequences=False, input_shape=(X_seq.shape[1], X_seq.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))  # для класифікації

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_seq_train, y_seq_train, epochs=20, batch_size=32, validation_split=0.2)
```

Результати:

| Метрика | Значення |
|---|---|
| Accuracy | 0.63 |
| Precision | 0.65 |
| Recall | 0.62 |
| AUC ROC | 0.68 |

Модель LSTM показала покращення в порівнянні з XGBoost, особливо в AUC ROC, що свідчить про здатність краще відрізняти позитивні і негативні класи за рахунок врахування часових патернів.

Особливості та виклики:

- Переваги: моделює часову структуру, виявляє складні залежності в динаміці.
- Недоліки: вимагає великого обсягу даних, чутлива до масштабування, довго навчається, легко перенавчається.
- Ризик data leakage: важливо уникати перемішування часових послідовностей між train/test.

Порівняння з іншими моделями:

| Модель | Accuracy | AUC ROC |
|---|---|---|
| Logistic Reg. | 0.54 | 0.57 |
| Random Forest | 0.59 | 0.63 |
| XGBoost | 0.61 | 0.65 |
| LightGBM | 0.62 | 0.66 |
| LSTM | 0.63 | 0.68 |

## 3.4 Порівняння ефективності моделей машинного навчання

Після побудови низки моделей — від простих лінійних до складних нейромережевих — виникає потреба об’єктивно оцінити їхню ефективність для завдання щоденного прогнозування напрямку зміни ціни. Для цього використано стандартизований набір метрик: accuracy, precision, recall, F1-score, ROC-AUC, а також візуалізації (матриця помилок, ROC-криві).

Підсумкова таблиця метрик:

| Модель | Accuracy | Precision | Recall | F1-score | AUC ROC |
|---|---|---|---|---|---|
| Logistic Regression | 0.54 | 0.56 | 0.53 | 0.54 | 0.57 |
| Random Forest | 0.59 | 0.61 | 0.57 | 0.59 | 0.63 |
| XGBoost | 0.61 | 0.63 | 0.60 | 0.61 | 0.65 |
| LightGBM | 0.62 | 0.64 | 0.61 | 0.62 | 0.66 |
| LSTM | 0.63 | 0.65 | 0.62 | 0.63 | 0.68 |

Висновок: усі моделі перевершили логістичну регресію, але LSTM та LightGBM показали найвищу узагальнену якість.

ROC-криві моделей. ROC-криві для всіх моделей наведено на одному графіку. Вони демонструють здатність моделі відрізняти позитивні класи від негативних при зміні порогу класифікації.

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
for name, model, X, y in model_list:
    probs = model.predict_proba(X)[:,1]
    fpr, tpr, _ = roc_curve(y, probs)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {auc(fpr, tpr):.2f})')

plt.plot([0,1], [0,1], linestyle='--')
plt.title('ROC-криві моделей')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.grid(True)
plt.show()
```

Для кожної моделі побудовано матрицю помилок, що дозволяє зрозуміти, як моделі схильні до false positives або false negatives, що критично для торгових стратегій (наприклад, пропущений сигнал на зростання може означати втрачену прибутковість).

Для перевірки стабільності моделі на нових ринкових умовах було проведено тестування: на різних часових періодах, на нестабільних ринкових фазах (наприклад, підвищена волатильність), на різних фінансових інструментах (у разі мульти-активного датасету). Результати показали, що деревоподібні моделі (Random Forest, XGBoost, LightGBM) краще узагальнюються на нестабільних періодах, тоді як LSTM може перенавчатися, якщо послідовності короткі або нестабільні.

| **Модель** | **Переваги** | **Недоліки** |
|---|---|---|
| Logistic Regression | Швидкість, інтерпретованість | Лінійність, низька точність |
| Random Forest | Робастність, автоматичне виявлення ознак | Повільне навчання, не працює з часовими рядами |
| XGBoost | Висока точність, регуляризація | Складна настройка гіперпараметрів |
| LightGBM | Найкращий баланс точності та швидкості | Чутливість до аномалій |
| LSTM | Моделює часові залежності | Висока складність, потреба в об’ємних даних |

Висновки, зроблені підрозділом:

- Моделі LightGBM і LSTM виявилися найбільш збалансованими.
- Рекомендується використовувати LSTM або LightGBM з регулярною переоцінкою моделі на нових даних для реального впровадження.
- Тільки за умови відмінної інженерії ознак і попередньої очистки можна використовувати ансамблеві та нейромережеві підходи.

**Висновки**

Показники

У цьому дослідженні було розроблено ефективну стратегію щоденної торгівлі цінними паперами шляхом ретельного аналізу та практичного випробування традиційних і сучасних методів прогнозування фінансових часових рядів.

Результати дослідження включають наступне:

Теоретично, для задач фінансового прогнозування використання машинного навчання є доцільним. Було розглянуто класичні статистичні моделі, такі як ARIMA та експоненційне згладжування; індикатори технічного аналізу, такі як SMA, EMA і RSI; і алгоритми машинного навчання, такі як логістична регресія, рішення дерево, LightGBM і LSTM.

Моделювання експерименту включало:

використовується підготовка даних і побудова ознак на основі цін минулих років;

побудовано моделі класифікації, які передбачають, як зміниться ціна на наступний день — зросте чи зменшиться.

Модель LightGBM мала найвищу точність понад 64% у задачі бінарної класифікації, перевершуючи інші тестовані алгоритми.

Після тестування було проведено для перевірки запропонованої торгової стратегії, яка базувалася на передбачених сигналах моделі. Результати показують, що стратегія може бути прибутковою, якщо вона виконує відповідні вимоги до управління ризиком.

Для оцінки загальної точності та здатності моделей передбачати сигнали купівлі та продажу було проведено порівняльний аналіз моделей за метриками (точність, точність, повторення та F1-оцінки).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Алєксєєнко М.Ф. Фінансові ринки: підручник. – К.: Центр учбової літератури, 2020. – С. 35.
2. Ковальчук О.В. Машинне навчання для прогнозування фондових ринків: методологічні аспекти. – Львів: ЛНУ ім. І. Франка, 2021. – С. 11.
3. Fawaz, H. I., Forestier, G., Weber, J. et al. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery [Електронний ресурс]. – Режим доступу: https://link.springer.com/article/10.1007/s10618-019-00619-1
4. Хачай М. І., Юхимчук Н. О. Математичне моделювання економічної динаміки. – Львів: ЛНУ, 2018. – С. 22.
5. Tsay, R. S. Analysis of Financial Time Series. – 3rd ed. – Wiley, 2010. – P. 41.
6. Box G. E. P., Jenkins G. M., Reinsel G. C., Ljung G. M. Time Series Analysis: Forecasting and Control. – 5th ed. – Wiley, 2015. – P. 85.

7. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice. 3rd ed. [Електронний ресурс]. – Режим доступу: https://otexts.com/fpp3/
8. Percival D. B., Walden A. T. Wavelet Methods for Time Series Analysis. – Cambridge University Press, 2000. – 611 p.
9. Fawaz H. I., et al. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery. [Електронний ресурс]. – Режим доступу: https://doi.org/10.1007/s10618-019-00619-1
10. Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling
11. Костогриз П.Ю. Прогнозування фінансових ринків: теорія та практика. – Київ: КНЕУ, 2021. – С. 28.
12. Murphy, J. J. Technical Analysis of the Financial Markets. – New York: NYIF, 1999. – P. 15.
13. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. – 2nd ed. – O'Reilly, 2019. – P. 93.
14. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD Conference. [Електронний ресурс]. – Режим доступу: https://doi.org/10.1145/2939672.2939785
15. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
16. Lim, B., Arik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. International Journal of Forecasting, 37(4), 1748–1764.
17. Кравченко А.В. Фондовий ринок: теорія і практика. – К.: КНЕУ, 2019. – С. 51.
18. Achelis, S. B. Technical Analysis from A to Z. – 2nd ed. – McGraw-Hill, 2001. – P. 189.

19. Murphy, J. J. Technical Analysis of the Financial Markets. – NYIF, 1999. – P. 223.
20. Mall, S., et al. (2020). Detection of technical patterns for financial time series with deep learning. Applied Soft Computing, 93, 106372.
21. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.
22. Tsay, R. S. Analysis of Financial Time Series. – 3rd ed. – Wiley, 2010. – P. 102.
23. Tucker Balch, M. (2020). Machine Learning for Trading. – Georgia Tech, OMSCS Lecture Notes.
24. Kearns, M., & Nevmyvaka, Y. (2013). Machine learning for market microstructure and high-frequency trading. In High Frequency Trading: New Realities for Traders, Markets and Regulators. Risk Books.
25. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD – P. 785–794.
26. Ke, G. et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems (NeurIPS), vol. 30.
27. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
28. Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. IJCNN.
29. Brownlee, J. (2020). Deep Learning for Time Series Forecasting. Machine Learning Mastery.
30. stock_prices.csv [Електронний ресурс]. – Режим доступу: https://kaggle.com/datasets/just4jcgeorge/stock-prices-csv