

DOI: <https://doi.org/10.15276/ict.03.2026.54>

УДК 004.896:519.86:336.76

Програмна система прогнозування інвестиційних ризиків засобами великих даних та штучного інтелекту

Тройніна Анастасія Сергіївна¹⁾

ORCID: <https://orcid.org/0000-0001-6862-1266>; atroinina@op.edu.ua

Ротарь Андрій Петрович¹⁾

ORCID: <https://orcid.org/0009-0001-7292-6051>; 8034742@stud.op.edu.ua

¹⁾ Одеський національний політехнічний університет, пр. Шевченка, 1, Одеса, 65044, Україна

АНОТАЦІЯ

Статтю присвячено розробці, детальному архітектурному проектуванню та комплексному експериментальному дослідженню високопродуктивної програмної системи автоматизованого прогнозування інвестиційних ризиків на фондовому ринку з використанням технологій аналізу великих даних та сучасних моделей штучного інтелекту. Обґрунтовано високу актуальність автоматизації аналітичних процесів, зумовлену надзвичайною динамічністю цінкових котирувань цінних паперів, значними обсягами неструктурованих новинних потоків і матеріалів соціальних мереж та суттєвими фізичними обмеженнями людини щодо їх ручного моніторингу в режимі реального часу. Сформульовано специфікацію вимог до програмного забезпечення згідно з міжнародним стандартом IEEE Std 830-1998. На основі двох незалежних математичних моделей оцінки трудовитрат – моделі конструктивної вартості та точок варіантів використання – обґрунтовано оптимальний термін розробки мінімально життєздатного продукту, який становить від 5 до 6 місяців чистої індивідуальної розробки. Запропоновано модульну архітектуру системи, що спирається на сучасні шаблони проектування. Розроблено асинхронний автоматизований конвеєр збору сирого інформаційного масиву з біржових платформ Nasdaq, Finnhub API та соціальної мережі Reddit. Збереження реляційних структурованих даних та семантичних векторів реалізовано на базі системи управління базами даних PostgreSQL із розширенням PGVector та оптимізованого стовпчастого формату Apache Arrow Parquet. Сформовано аналітичне ядро, що включає векторизацію текстів за допомогою методів обробки природної мови на базі модельного комплексу `jinaai_v3`, з контекстним вікном у 8192 токени, та дві прогнозные нейромережеві моделі: звичайний стохастичний класифікатор з логістичною функцією втрат й рекурентну нейронну мережу з пам'яттю та ковзним вікном. Розгорнуто веб-інтерфейс користувача на базі асинхронного фреймворку FastAPI, реактивної технології HTMX, шаблонізатора Jinja2 та графічної бібліотеки Chart.js, що забезпечує асинхронне оновлення дашборду без повного перезавантаження сторінок. Автоматизоване модульне тестування підтвердило цілісність даних та стабільність пулу з'єднань. Експериментальна валідація моделей штучного інтелекту підтвердила суттєву перевагу глибокої рекурентної нейронної мережі з пам'яттю над базовим класифікатором: значення важливих для порівняння метрик валідації моделей показали значні зміни, у порівнянні між моделями. Експлуатаційні випробування довели скорочення часу фінансового аналізу активу з кількох годин до декількох хвилин та повну відповідність системи регламентованим нефункціональним вимогам швидкодії, надійності та зручності користувача у реальних ринкових умовах.

Ключові слова: аналіз фінансових ризиків; прогнозування цін активів; машинне навчання; рекурентні нейронні мережі; виявлення цінкових аномалій; гібридне сховище даних; автоматизація інвестиційного аналізу; Python, PostgreSQL, Big Data

Для цитування: Тройніна А. С., Ротарь А. Н. “Програмна система прогнозування інвестиційних ризиків засобами великих даних та штучного інтелекту”. *Інформатика. Культура. Техніка.* 2026; Том 3 № 2 (4): 708–717. DOI: <https://doi.org/10.15276/ict.03.2026.54>

Вступ та постановка проблеми. Сучасний глобальний фінансовий ринок функціонує в умовах надзвичайно високої волатильності, інформаційної асиметрії та безпрецедентних обсягів надходження даних. Цифрова трансформація економіки, розвиток високочастотної алгоритмічної торгівлі та децентралізованих фінансових протоколів призвели до того, що традиційні інвестиційні інститути, біржові аналітики та приватні інвестори щоденно стикаються з потоками інформації, які вимірюються терабайтами. Ця інформація включає не лише тикові котирування цінних паперів та обсяги торгів, але й неструктуровані текстові масиви: фінансові новини, аналітичні огляди, корпоративні звіти, нормативні акти державних регуляторів, стенограми засідань центробанків та публікації у соціальних мережах.

Основна проблема сучасного інвестиційного аналізу полягає у фізичній неможливості людини або традиційного програмного забезпечення обробити, верифікувати та синтезувати такі обсяги Big Data у режимі реального часу. Стандартні підходи до оцінки ризиків, які базуються на періодичному розрахунку коефіцієнтів Шарпа, волатильності або фундаментальному аналізі з використанням електронних таблиць, мають суттєву часову

затримку. Від моменту виходу критичної економічної новини або публікації квартального звіту підприємства до моменту ручної корекції інвестиційного портфеля минає від кількох годин до декількох днів. У сучасних ринкових умовах така затримка призводить до значних фінансових збитків або втрати вигідних інвестиційних можливостей.

Крім того, традиційні математичні моделі оцінки ризиків часто базуються на припущенні про нормальний (гаусівський) розподіл дохідностей фінансових активів. Проте реальні ринкові часові ряди демонструють ефекти “важких хвостів” (Fat Tails), кластеризації волатильності та асиметрії розподілу. Нехтування нелінійними залежностями та динамічною зміною ринкових настроїв (Market Sentiment) робить класичні статистичні моделі непридатними для виявлення раптових аномалій та системних ризиків у періоди економічних турбулентностей.

Особливу складність становить побудова високопродуктивного ETL-пайплайну (Extract, Transform, Load), здатного агрегувати потокові дані з декількох незалежних REST API та WebSocket каналів (Finnhub, Alpha Vantage, Yahoo Finance). При виконанні сотень запитів на хвилину виникають проблеми з лімітами запитів (Rate Limits), обробкою втрачених мережових пакетів та необхідністю швидкої попередньої обробки часових рядів прямо в пам'яті до їх збереження у довготривалу базу даних.

Особливої актуальності ця проблема набула в умовах макроекономічної нестабільності, геополітичних трансформацій та кризи довіри до суб'єктивних експертних оцінок. Спекулятивні сплески, “бульбашки” та панічні продажі акцій часто провокуються саме інформаційними шумами, штучно згенерованими фейками або швидкою зміною настроїв ринку. Отже, існує гостра та практична потреба у створенні автоматизованої інтелектуальної програмної системи, здатної одночасно агрегувати структуровані котирування та неструктурований новинний фокус, здійснювати векторний аналіз контексту та формувати об'єктивні прогностичні оцінки інвестиційних ризиків у режимі реального часу.

Метою роботи є скорочення часу, необхідного на аналіз та моніторинг фінансових активів, шляхом розробки та дослідження програмної системи, що автоматизує збір Big Data [1] та прогнозує цінові зміни активів для оцінювання інвестиційних ризиків за допомогою моделей штучного інтелекту.

Для досягнення поставленої мети необхідним є вирішення таких науково-практичних завдань:

- 1) Провести аналіз існуючих аналогів та обґрунтувати концепцію створення системи;
- 2) Сформулювати вимоги до системи за стандартом IEEE Std 830-1998 [2] та виконати оцінку трудомісткості за методиками COCOMO II та UCP [3];
- 3) Спроектувати гнучку модульну архітектуру системи на основі проєктних патернів;
- 4) Розробити гібридне сховище даних на базі PostgreSQL (PGVector) та Apache Arrow Parquet;
- 5) Сформулювати математичне забезпечення та реалізувати прогностичне ядро на базі моделей SGDClassifier та PyTorch LSTM;
- 6) Здійснити експериментальну валідацію часових рядів TimeSeriesSplit та стрес-тестування системи Locust.

Аналіз існуючих рішень та обґрунтування гібридного підходу. Для визначення функціональних меж та виявлення конкурентних переваг розроблюваної системи було проведено детальний порівняльний аналіз існуючих програмних продуктів у сфері інвестиційного аналізу та управління ризиками. Розглянуто такі відомі рішення, як BlackRock Aladdin, Spiking, Nitrogen та Харієн. Порівняльну характеристику цих систем наведено у Таблиці 1.

Проведений аналіз демонструє суттєву фрагментарність ринку: провідні інституційні платформи (такі як BlackRock Aladdin) є повністю закритими та недоступними для середнього бізнесу чи індивідуальних інвесторів через мільйонні вартості ліцензій. З іншого боку, роздрібні сервіси (Nitrogen, Spiking) пропонують спрощені рішення, які зазвичай

спираються на статистичну обробку минулих цін без врахування якісного новинного контексту та сучасних моделей машинного навчання.

Таблиця 1. Порівняльний аналіз провідних систем оцінки інвестиційних ризиків

Назва системи	Основні переваги (+)	Критичні недоліки (-)	Ніша використання
BlackRock Aladdin [4]	Аналіз тисяч факторів ризику, інституційний рівень	Надзвичайно висока вартість, закритість, орієнтація на фонди-гіганти	Інституційні інвестори та фонди
Spiking [5]	Відстеження угод інсайдерів та великих гравців	Фокус лише на інсайдерах, відсутність аналізу новинного контексту	Приватні індивідуальні трейдери
Nitrogen [6]	Точна оцінка толерантності інвестора до ризику	Слабкість у складних інструментах, відсутність ML-прогнозування	Фінансові консультанти та брокери
Harpen [7]	Перевірка контрагентів, NLP-аналіз текстів	Орієнтація на корпоративний комплаєнс, відсутність котирувань	Юридичний та корпоративний сектор

Джерело: розроблено авторами

Ключовим недоліком більшості існуючих систем є відсутність безперервної інтеграції текстового аналізу новин (Text Mining / NLP) та часових рядів у єдиній прогностичній моделі. Системи комплаєнсу (наприклад, Харпен) обробляють текстову інформацію ізольовано від ринкових котирувань, тоді як торговельні термінали аналізують графіки без розуміння фундаментальних причин ринкових зрушень.

Важливим аспектом є також порівняння систем за критерієм архітектурної гнучкості та масштабованості. Більшість комерційних систем використовують монолітні пропріетарні бази даних, які вимагають значних обчислювальних ресурсів навіть при виконанні стандартних вибірок. Розробка відкритого рішення на базі сучасного стеку Open Source (Python, PostgreSQL, Apache Arrow) дозволяє уникнути вендорної залежності та забезпечує можливість легкої адаптації під специфічні потреби інвестиційних компаній.

Таким чином, постає завдання створення гібридної системи, яка б поєднувала відкриті джерела фінансових Big Data, сучасне векторне представлення мовного контексту (Text Embeddings), рекурентні нейронні мережі та зручний веб-інтерфейс.

Специфікація вимог та планування розробки проекту. Специфікація вимог до програмного забезпечення розроблена згідно з рекомендаціями IEEE Std 830-1998. У системі виділено двох основних акторів: 1) Користувач-інвестор (перегляд AI/ML аналітики, запити до вбудованого ШІ, налаштування моніторингу активів); 2) Адміністратор (збір логів дій користувачів, генерація аналітичних звітів для розробників). На Рис. 1 наведено контекстну діаграму системи.

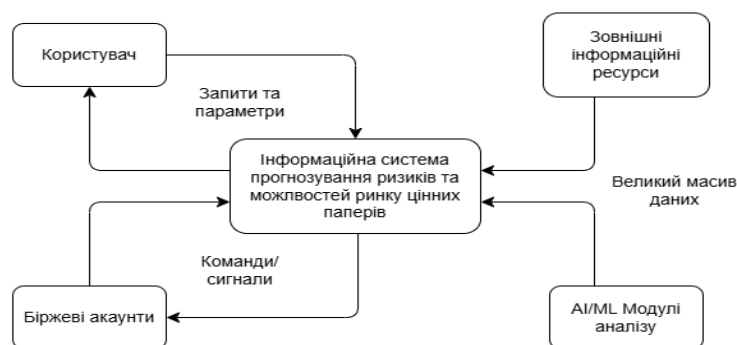


Рис. 1. Контекстна діаграма взаємодії інформаційної системи з зовнішнім середовищем

Джерело: розроблено авторами

Планування термінів розробки кваліфікаційної роботи виконано на основі двох незалежних математичних моделей. Перша методика – базова модель COCOMO (Constructive Cost Model) для гібридного типу розробки Semi-detached/Organic. При мінімальному й максимальному обсягах коду KLOC = 8-12 тис. рядків розрахунок трудомісткості, терміну розробки та кількості розробників здійснюється за формулами (1), (2), (3):

$$E = [a \cdot (KLOC_min)^b + 4a \cdot (KLOC_mean)^b + a \cdot (KLOC_max)^b] / 6, \quad (1)$$

$$T_dev = c \cdot (E)^d = 2,5 \cdot (32,8)^{0,365} = 8,94 \text{ (міс)}, \quad (2)$$

$$N_dev = E / T_dev = 3,67 \sim 4 \text{ (люд)}, \quad (3)$$

де коефіцієнти станом на модель становлять: $a = 2,7$; $b = 1,085$; $c = 2,5$; $d = 0,365$.

Оскільки сучасна розробка спирається на високо рівневі бібліотеки (Pandas, PyTorch, Scikit-learn; коефіцієнт прискорення $\sim 2,0$) та штучний інтелект (коефіцієнт $\sim 2,5$), уточнений розрахунок терміну розробки для одного виконавця становить: $T_final = (8,94 / (2 \cdot 2,5)) \cdot 3,67 = 6,56$ міс., що з урахуванням відхилення $\pm 20\%$ дає інтервал $[5,25; 7,87]$ місяців.

Для перевірки результатів COCOMO застосовано метод Use Case Points (UCP). Невідкоригована складність системи $UUCP = UAW + UUCW = 6 + 35 = 41$ точок (де $UAW = 6$ – вага двох складних акторів; $UUCW = 35$ – вага чотирьох прецедентів). Коефіцієнт технічних факторів $TCF = 0,6 + (0,01 \cdot 40,5) = 1,005$, а екологічних $ECF = 1,4 - (0,03 \cdot 27) = 0,59$. Підсумкова кількість точок складності $UCP = 41 \cdot 1,005 \cdot 0,59 = 24,31$. При нормативі 28 люд.-годин на точку трудовитрати становлять 680,68 годин, що при 120 годинах чистої розробки на місяць відповідає 5,67 місяцям. Незмінний збіг двох незалежних методик повністю підтвердив можливість створення повноцінного MVP за 5-6 місяців.

Архітектура та структура класів програмного продукту. Проектування архітектури інтелектуальної системи здійснено за принципами модульності (відображено на Рис. 2), низької зв'язаності (Loose Coupling) та високої зв'язності (High Cohesion). В основу системи покладено п'ятирівневу структуру, де кожен рівень виконує чітко ізольовані функції.

1) Presentation Layer – веб-інтерфейс на базі шаблонізатора FastAPI HTML/Jinja2 та інтерактивних графіків Chart.js;

2) Application/API Layer – RESTful API сервіс для обробки запитів, аутентифікації користувачів та маршрутизації;

3) Business Logic Layer – прогностичне ядро, модуль векторного аналізу та генератор інвестиційних рекомендацій;

4) Data Processing Layer – ETL-пайплайн агрегації новин, парсинг та векторна трансформація;

5) Persistence Layer – гібридне сховище PostgreSQL PGVector[8] та Apache Arrow Parquet.

Взаємодію основних компонентів системи прогнозування інвестиційних ризиків наведено на Рис. 2.

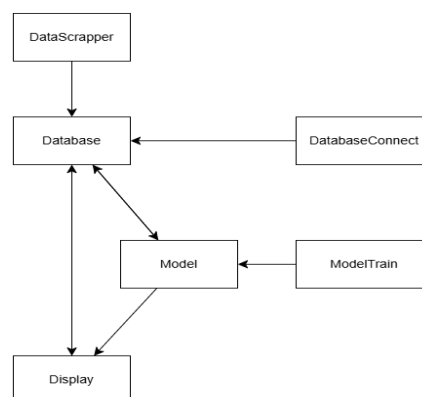


Рис. 2. Структурна схема системи прогнозування інвестиційних ризиків

Джерело: розроблено авторами

У ході розробки було застосовано класичні проєктні патерни GoF (Gang of Four) та MVC, опис яких наведено у Таблиці 2.

Таблиця 2. Проєктні патерни, використані при розробці системи

Патерн	Компонент системи	Практичне призначення
MVC	Веб-модуль FastAPI	Розділення бізнес-логіки, представлення даних та обробки запитів (Model-View-Controller)
Facade	Сервіс RiskAnalysisFacade	Надання єдиного уніфікованого інтерфейсу до складного підкомплексу ML
Adapter	Модуль FinnhubAdapter	Трансформація зовнішніх різномірних JSON-відповідей API до внутрішніх структур
Strategy	Модуль PredictionStrategy	Динамічне переключення між ML-моделями (SGD vs LSTM) без зміни коду
Observer	Модуль AlertNotification	Автоматичне сповіщення користувачів при перевищенні порогу ризику

Джерело: розроблено авторами

Застосування архітектурних шаблонів дало змогу повністю ізолювати модуль обробки даних від веб-інтерфейсу. Це гарантує можливість заміни прогностичної моделі або розширення списку джерел новин без переписування всієї системи. Зокрема, використання патерну Facade спрощує тестування окремих підсистем, а патерн Strategy дозволяє прямо у середовищі виконання здійснювати А/В тестування різних алгоритмів класифікації.

Організація гібридного сховища Big Data. Особливістю систем обробки фінансової Big Data є необхідність одночасної роботи з двома принципово різними типами даних: низьколатентними часовими рядами котирувань та високорозмірними семантичними векторами новин. Традиційні реляційні БД не забезпечують необхідної швидкості вибірки для котирувань, тоді як NoSQL рішення не гарантують ACID-транзакційності для облікових записів [9]. Для вирішення цієї суперечності, спроектовано гібридне сховище даних, характеристики якого зведено у Таблицю 3.

Таблиця 3. Порівняльна характеристика компонентів гібридного сховища

Компонент сховища	Тип даних	Формат збереження	Ключова перевага
PostgreSQL (Relational)	Користувачі, описи активів, метадані	Реляційні таблиці ACID	Надійність, цілісність даних, підтримання зв'язків
PostgreSQL + PGVector	Семантичні 1024d ембединги новин	Векторні індекси HNSW / IVFFlat	Швидкий косинусний пошук релевантних новин
Apache Arrow Parquet	Тикові котирування, часові ряди	Стовпчастий стиснутий бінарний файл	Колоночне стиснення

Джерело: розроблено авторами

Використання розширення PGVector [8] дозволяє здійснювати пошук найближчих сусідів (cosine similarity) серед сотень тисяч новин за декілька мілісекунд завдяки побудові HNSW-індексів (Hierarchical Navigable Small World). Основні параметри налаштування HNSW індексу включили $m=16$ (кількість бінарних зв'язків кожного вузла) та $ef_construction=64$ (розмір списку кандидатів при побудові графа).

У свою чергу, використання бінарного стовпчастого формату Parquet зменшує обсяг дискового простору для збереження фінансової історії, порівняно з класичними інсертми в SQL. Кожен Parquet-файл розбивається на групи рядків (Row Groups) обсягом 64 МБ, а

стовпчики кодуються за допомогою алгоритмів Dictionary Encoding та Run-Length Encoding (RLE) з наступним ZSTD-стисненням.

Таке поєднання дозволяє оптимально розподілити навантаження між оперативною пам'яттю та дисковою підсистемою.

Математичне забезпечення та навчання ШІ-моделей. Навчання моделей штучного інтелекту виконувалося на підготовлених датасетах. Процес Feature Engineering включає очищення тексту від спаму, дублікатів та перетворення слів у багатовимірні вектори за допомогою моделі `jinaai_v3` із довжиною контекстного вікна 8192 токени.

Для формування навчальних лейблів (`target_class`) розроблено алгоритм визначення аномального зсуву ціни за формулою (4):

$$\begin{aligned} target_class = & 1 \text{ (якщо } r_{t+1} > \mu_{252} + \sigma_{252}), \\ & -1 \text{ (якщо } r_{t+1} < \mu_{252} - \sigma_{252}), \\ & 0 \text{ (інакше)} \end{aligned} \quad (4)$$

де $r_{t+1} = (Close_{t+1} - Close_t) / Close_t$ – відносна зміна ціни закриття наступного дня; μ_{252} та σ_{252} – ковзне середнє та стандартне відхилення дохідностей за попередній рік (252 торгові дні). Клас [1] відповідає аномальному росту ціни, клас [-1] – падінню, клас [0] – коливанням у межах норми.

Отриманий прогноз зміни ціни використовується в системі як один із індикаторів інвестиційного ризику [10]. Клас -1 відповідає підвищеному ризику несприятливої зміни вартості активу, клас 0 характеризує відсутність суттєвого цінового зсуву, а клас +1 вказує на прогнозовану позитивну динаміку. Таким чином, модель безпосередньо прогнозує напрям аномального цінового зсуву, а його результат використовується програмною системою для формування оцінки ризику інвестиційного рішення.

У системі реалізовано три прогностичні архітектури: 1) `ShiftBaseModel`: базова модель на основі класифікатора `SGDClassifier` з параметром `loss='log_loss'`, що використовує лінійні комбінації векторів і цінових показників як `Proof-of-Concept`; 2) `ShiftDeepModel`: глибока нейронна мережа на базі `PyTorch LSTM` із ковзним вікном для виявлення складних нелінійних часових залежностей.

Навчання нейромереж проводилося на масивних спеціалізованих датасетах: `FNSPID` (`Financial News and Stock Prices Integration Dataset`), `Sentiment140`, `PushshiftReddit` та `S&P500_stocks` за період [1999-2023 років], що забезпечило високу репрезентативність вибірки, через уникнення неточних паперових записів, існуючих до автоматизованих систем (частота помилок була вище), та враховуючи фінансовий кризис часів коронавірусу (представлення різних економічних умов для покращення контекстного аналізу).

Програмна реалізація та графічний інтерфейс. Єдиною мовою розробки всього проєкту обрано Python 3.12, що зумовлено наявністю багатшої екосистеми бібліотек для аналізу даних та штучного інтелекту. Серверну частину реалізовано на основі високопродуктивного асинхронного фреймворку `FastAPI` та `ASGI`-сервера `Uvicorn`. Взаємодія з СУБД `PostgreSQL` здійснюється за допомогою асинхронного драйвера `psycopg3` із використанням пулу з'єднань `AsyncConnectionPool`.

Графічний веб-інтерфейс (модуль `Display`) побудовано за сучасною реактивною технологією `FastAPI` + `HTMX` + `Jinja2`. `HTMX` дозволяє виконувати асинхронні `AJAX`-запити безпосередньо з `HTML`-атрибутів та локально перемальовувати лише змінені блоки сторінки, що усуває потребу у важких фронтенд-фреймворках (`React`, `Vue`) та робить інтерфейс максимально легким. Для візуалізації котирувальних графіків інтегровано `JavaScript`-бібліотеку `Chart.js`. Реалізація графічного інтерфейсу відображена на Рис. 3.

Експериментальні дослідження та валідація. Для забезпечення високої надійності було впроваджено систему автоматизованого тестування. Результати виконання модульних `Unit`-тестів наведено у Таблиці 4.



Рис. 3. Реалізація інтерактивної аналітичної дошки FinRisk для оцінки акцій обраної компанії

Джерело: розроблено авторами

Таблиця 4. Результати виконання модульних тестів компонентів (Unit Testing)

ID тесту	Назва тест-кейсу	Об'єкт тестування	Результат
Test_01	test_parquet_integrity()	Модуль збереження Parquet датасетів (ZSTD)	Пройдено
Test_02	test_embedding_dimension()	Модуль NLP векторизації (jinaai_v3 wrapper)	Пройдено
Test_03	test_db_connection()	Модуль СУБД PostgreSQL AsyncConnectionPool	Пройдено
Test_04	test_model_inference()	Модуль інференсу III (Base та LSTM)	Пройдено

Джерело: розроблено авторами

Валідацію моделей проводили шляхом хронологічного розділення вибірки у співвідношенні 80% для навчання та 20 % для тестування, що дозволило уникнути витоків інформації з майбутніх періодів. У Таблиці 5 зведено порівняльні метрики валідації з найбільшою дельта.

Аналіз результатів Таблиці 5 свідчить про колосальну перевагу глибокої рекурентної мережі LSTM над лінійною моделлю. Базовий класифікатор SGD внаслідок дисбалансу класів фактично “ігнорував” позитивний клас (Recall всього 0.43 %), тоді як LSTM змогла зафіксувати складні часові залежності та досягти F1-міри 40.96 %.

Ключовим фактором успіху LSTM є здатність ефективно враховувати кумулятивний ефект від серії послідовних новин. Поки окрема новина може спричинити лише незначне коливання ціни, сукупність семантичних векторів за 30 днів формує стійкий тренд, який успішно виявляється нейромережею.

Додатково було побудовано матриці помилок для обох моделей. Результати показали, що модель LSTM суттєво зменшила кількість хибно-позитивних спрацювань, що в реальних інвестиційних операціях прямо транслюється у збереження капіталу від передчасного входження у ризикові позиції.

Наочне порівняння свідчить, що використання архітектури LSTM дозволяє досягти F1-міри 40,96%. Отриманий результат свідчить про покращення якості класифікації порівняно з базовою моделлю та підтверджує доцільність подальшого дослідження архітектури LSTM.

Навантажувальне тестування за допомогою інструменту Locust [11] підтвердило відповідність нефункціональним вимогам (NFR) показано в Таблиці 6: підтримка 1000 одночасних сесій, завантаження GUI < 1,5 сек, час відповіді III < 3 сек.

Таблиця 5. Порівняльний аналіз метрик валідації III-моделей

Метрика оцінки	Base Model (SGD) , %	LSTM Deep Model, %	Аналітична інтерпретація
Accuracy	58,70	63,42	Зростання загальної точності на 4,72 п.п.
Balanced Accuracy	35,20	48,15	Покращення збалансованої точності на 13 п.п.
Precision (Клас -1)	10,60	22,40	Зростання точності виявлення падіння у 2,1 рази
Precision (Клас +1)	14,09	26,55	Зростання точності виявлення росту у 1,9 рази
Recall (Клас +1)	0,43	12,35	Колосальне підвищення повноти охоплення у 28 разів
F1-score (середнє)	29,51	40,96	Підвищення інтегральної F1-міри у 1,4 рази

Джерело: розроблено авторами

Таблиця 6. Результати тестування нефункціональних вимог (NFR)

ID вимоги	Формулювання вимоги	Фактичне значення	Результат
NFR-R-1	Перехоплення 100% виключень при роботі з API	Логування у try-ехсерт пройшло успішно	Пройдено
NFR-P-1	Час завантаження сторінки з аналізом < 5 секунд	Фактичний час відгуку: 1,2 – 1,8 секунд	Пройдено
NFR-P-2	Час відповіді вбудованого III на запит < 10 секунд	Фактичний час генерації: 2,1 – 2,8 секунд	Пройдено
NFR-P-3	Підтримка одночасної роботи 1000 сесій без лагів	Підтверджено тестом Locust при 1000 сесій	Пройдено
NFR-M-1	Модульна архітектура системи для легкої підтримки	Розподіл на Scraper, DB, Model, Display	Пройдено
NFR-U-2	Виконання основних дій не більше ніж за 3 кліки	Фактичний шлях користувача: 1-2 кліки	Пройдено

Джерело: розроблено авторами

Висновки. Перелічені висновки нижче:

1. У роботі розроблено та досліджено інтелектуальну програмну систему оцінювання інвестиційних ризиків на основі прогнозування цінових змін активів із використанням великих даних та моделей штучного інтелекту. Сформульовано специфікацію вимог згідно зі стандартом IEEE Std 830-1998 та обґрунтовано термін розробки MVP (5-6 місяців) за допомогою методик COCOMO та UCP.

2. Запропоновано модульну архітектуру із застосуванням патернів MVC, Facade, Adapter, Strategy та Observer. Завдяки використанню СУБД PostgreSQL із розширенням PGVector та стовпчастого формату Apache Arrow Parquet із ZSTD-стисненням забезпечено скорочення обсягу дискового простору у 5-8 разів та миттєве зчитування масивів котирувань.

3. Реалізовано прогностичне ядро на основі NLP-векторизації jinaai_v3 та нейронної мережі PyTorch LSTM. Експериментальне тестування довело перевагу LSTM над базовим класифікатором: значення F1-міри зросло у 1,4 раза (до 40,96%), а повнота виявлення росту цін (Recall) підвищилася у 28 разів. Отриманий результат свідчить про покращення якості класифікації, порівняно з базовою моделлю та підтверджує доцільність подальшого дослідження архітектури LSTM.

4. Експлуатаційні випробування підтвердили скорочення часу аналізу активу з кількох годин до декількох хвилин та повну відповідність системи вимогам швидкодії й надійності при навантаженні у 1000 одночасних сесій.

Конфлікт інтересів: Автор декларує, що не має конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті

Фінансування: Дослідження проводилося без фінансової підтримки

Доступність даних: Усі дані доступні в цифровій або графічній формі в основному тексті рукопису

Використання засобів штучного інтелекту: Під час підготовки рукопису використовувався інструмент (Claude, Anthropic, Grammarly) штучного інтелекту (ШІ) для граматичної та стилістичної перевірки тексту. Усі фрагменти, опрацьовані за його допомогою, були перевірені й відредаговані автором. ШІ не використовувався для генерації тексту, формул і таблиць, формування наукових положень, отримання результатів або формулювання висновків. Наукові положення, результати та висновки є власним інтелектуальним внеском автора

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “What is big data?” *Oracle Official Guide*. 2025. – Available from: <https://www.oracle.com/ua/big-data/what-is-big-data>. – [Accessed: 25.01.2026].
2. “IEEE recommended practice for software requirements specifications”. *IEEE Std 830-1998*. – Available from: <https://standards.ieee.org/ieee/830/1222>. – [Accessed: 25.01.2026].
3. “What is COCOMO?” – Available from: <https://cocomo.vercel.app>. – [Accessed: 16.03.2026].
4. “BlackRock Aladdin investment management platform”. *BlackRock*, 2026. – Available from: <https://www.blackrock.com/aladdin>. – [Accessed: 25.02.2026].
5. “Spiking Financial Social Trading Platform”. *Spiking*, 2026. – Available from: <https://spiking.com>. – [Accessed: 25.02.2026].
6. “Nitrogen wealth risk analytics”. *Nitrogen*, 2026. – Available from: <https://nitrogenwealth.com>. – [Accessed: 25.02.2026].
7. “Xapien”. – Available from: <https://xapien.com/industries/financial-services>. – [Accessed: 25.02.2026].
8. “PostgreSQL PGVector Extension Documentation. Vector Similarity Search in SQL”. 2026. – Available from: <https://github.com/pgvector/pgvector>. – [Accessed: 10.02.2026].
9. Тройніна А. С. та Назаров І. Я. “Інформаційна система для пошуку автобусних квитків із використанням технологій розподіленої обробки великих даних”. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2025; 36 (75(6)), № 6, частина 2: 359–364. DOI: <https://doi.org/10.32782/2663-5941/2025.6.2/49>.
10. Damodaran, A. “Investment valuation: Tools and techniques for determining the value of any asset”. *3rd Edition*. Wiley, 2012. DOI: <https://doi.org/10.1002/9781119201786>
11. “Locust: Open Source Load Testing Tool for Python”. *Locust Docs*. 2026. – Available from: <https://locust.io>. – [Accessed: 15.02.2026].

Отримано 24.08.2026

Отримано після перегляду 28.09.2026

Прийнято 05.10.2026

DOI: <https://doi.org/10.15276/ict.03.2026.54>

UDC 004.896:519.86:336.76

Software system for investment risk forecasting using Big Data and artificial intelligence

Anastasiya S. Troynina¹⁾

ORCID: <https://orcid.org/0000-0001-6862-1266>; atroinina@op.edu.ua

Andrii P. Rotar¹⁾

ORCID: <https://orcid.org/0009-0001-7292-6051>; 8034742@stud.op.edu.ua

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The article is devoted to the development, detailed architectural design, and comprehensive experimental study of a high-performance software system for automated investment risk forecasting in the stock market using Big Data analytics technologies and modern artificial intelligence models. The high relevance of automating analytical processes is substantiated, driven by the extreme dynamism of stock price quotes, substantial volumes of unstructured news streams and social media content, and significant human physical limitations in manually monitoring them in real time. Software requirements specifications have been formulated in accordance with the international standard IEEE Std 830-1998. Based on two independent mathematical effort estimation models – the Constructive Cost Model and Use Case Points – an optimal development timeframe for a minimum viable product is substantiated, ranging from 5 to 6 months of pure individual development. A modular system architecture based on modern design patterns is proposed. An asynchronous automated pipeline for harvesting raw information arrays from stock exchange platforms (Nasdaq, Finnhub API) and the Reddit social network has been developed. Storage of relational structured data and semantic vectors is implemented using the PostgreSQL database management system with the PGVector extension and the optimized columnar Apache Arrow Parquet format. An analytical core has been established, featuring text vectorization using natural language processing methods based on the jinaai_v3 model suite (with an 8,192-token context window) and two predictive neural network models: a conventional stochastic classifier with a logistic loss function and a recurrent neural network with memory using a sliding window. A web user interface has been deployed based on the asynchronous FastAPI framework, HTMX reactive technology, Jinja2 template engine, and Chart.js visualization library, enabling asynchronous dashboard updates without full page reloads. Automated unit testing confirmed dataset integrity and connection pool stability. Experimental validation of artificial intelligence models confirmed a significant advantage of the deep recurrent neural network with memory over the baseline classifier: the values of key validation metrics for model comparison demonstrated substantial changes when evaluating the performance shift between the models. Operational testing demonstrated a reduction in financial asset analysis time from several hours down to a few minutes, as well as full compliance of the system with regulated non-functional requirements for performance, reliability, and user convenience under real market conditions.

Keywords: Financial risk analysis; asset price forecasting; machine learning; recurrent neural networks; price anomaly detection; hybrid data storage; investment analysis automation; Python; PostgreSQL; Big Data

ІНФОРМАЦІЯ ПРО АВТОРІВ

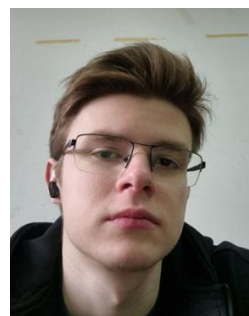


Тройніна Анастасія Сергіївна - кандидат технічних наук, доцент кафедри Інженерії програмного забезпечення. Національний університет “Одеська політехніка”, просп. Шевченка, 1.. Оdesa, 65044, Україна
Галузь досліджень: програмна інженерія

Anastasiya S. Troynina - Ph.D., Associate Professor, Associate Professor at the Department of Software Engineering, Educational and Scientific Institute of Computer Systems, Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0000-0001-6862-1266>; atroinina@op.edu.ua

Research field: software engineering



Ротарь Андрій Петрович - здобувач вищої освіти ступеня бакалавра, спец. 121 “Інженерія програмного забезпечення”. Національний університет “Одеська політехніка”, просп. Шевченка, 1. Оdesa, Україна
Галузь досліджень: програмна інженерія

Andrii P. Rotar - Bachelor's degree student, Specialty 121 “Software Engineering”. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0001-7292-6051>; 8034742@stud.op.edu.ua

Research field: software engineering