

ВИКОРИСТАННЯ МОДЕЛЕЙ НА ОСНОВІ МЕХАНІЗМУ УВАГИ ДЛЯ ПЕРЕДБАЧЕННЯ ФІНАНСОВИХ ЧАСОВИХ РЯДІВ

**Текстова частина до магістерської роботи
за спеціальністю «Комп'ютерні науки»**

Керівник роботи
доктор фіз.-мат. наук, професор Глибовець М. М.

«__» _____ 2026 р.

Виконав студент

Задохін Д. В.

«__» _____ 2026 р.

ЗМІСТ

АНОТАЦІЯ	4
ВСТУП.....	7
1 ОГЛЯД НАУКОВИХ ДОСЛІДЖЕНЬ	10
1.1 Патчування для вирішення проблеми семантичної бідності часових точок.	12
1.2 Модифікації механізму уваги для роботи з часовими рядами.....	14
1.3 Використання частотної декомпозиції для роботи з нестационарними і періодичними часовими рядами	17
1.4 Проблеми при роботі з багатовимірними часовими рядами та інверсія вхідних даних.....	20
2 ПРАКТИЧНА ЧАСТИНА	22
2.1 Гібридна архітектура FEiTRANSFORMER.....	23
2.2 Опис задачі та тренувальних даних	25
2.3 Конфігурація і оцінювання моделей	30
2.4 Тренування моделей	34
3 РЕЗУЛЬТАТИ.....	36
3.1 РЕЗУЛЬТАТИ МОДЕЛІ TRANSFORMER	37
3.2 РЕЗУЛЬТАТИ МОДЕЛІ INFORMER	38
3.3 РЕЗУЛЬТАТИ МОДЕЛІ PATCHTST	39
3.4 РЕЗУЛЬТАТИ МОДЕЛІ iTRANSFORMER	40
3.5 РЕЗУЛЬТАТИ МОДЕЛІ FEDFORMER	41
3.6 РЕЗУЛЬТАТИ МОДЕЛІ FEiTRANSFORMER	42

3.7	ОСТАТОЧНЕ ПОРІВНЯННЯ МОДЕЛЕЙ І РЕТРОСПЕКТИВНЕ ТЕСТУВАННЯ	43
	ВИСНОВКИ.....	46
	СПИСОК ЛІТЕРАТУРИ.....	49

Перелік прийнятих скорочень

API - Application Programming Interface - прикладний програмний інтерфейс.

BTC - Bitcoin - криптовалютний актив Bitcoin.

BTC-USDT - торгова пара Bitcoin до Tether USD.

DirAcc - Directional Accuracy - точність визначення напрямку зміни ціни.

ETT - Electricity Transformer Temperature - набір даних часових рядів, який часто використовується для оцінювання моделей прогнозування.

FEB - Frequency Enhanced Block - блок частотного посилення, що використовується для обробки частотних компонентів часового ряду.

FEB-f - Frequency Enhanced Block with Fourier transform - варіант блоку частотного посилення з використанням перетворення Фур'є.

FEB-w - Frequency Enhanced Block with Wavelet transform - варіант блоку частотного посилення з використанням вейвлет-перетворення.

FEDFormer - Frequency Enhanced Decomposed Transformer - трансформерна архітектура з використанням декомпозиції часового ряду та частотного аналізу.

FEiTTransformer - Frequency Enhanced Inverted Transformer - запропонована у роботі гібридна архітектура, що поєднує інверсію вхідного багатовимірного представлення з елементами частотної декомпозиції.

FFT - Fast Fourier Transform - швидке перетворення Фур'є.

IFFT - Inverse Fast Fourier Transform - обернене швидке перетворення Фур'є.

IDWT - Inverse Discrete Wavelet Transform - обернене дискретне вейвлет-перетворення.

Informer - трансформерна архітектура для прогнозування довгих часових рядів із використанням розрідженого механізму уваги ProbSparse Attention.

iTransformer - Inverted Transformer - трансформерна архітектура, що використовує інверсію представлення багатовимірного часового ряду.

LR - Learning Rate - швидкість навчання моделі.

MSE - Mean Squared Error - середньоквадратична похибка.

NLP - Natural Language Processing - обробка природної мови.

OHLCV - Open, High, Low, Close, Volume - набір базових біржових показників: ціна відкриття, максимум, мінімум, ціна закриття та обсяг торгів.

PatchTST - Patch Time Series Transformer - трансформерна архітектура для часових рядів, що використовує розбиття послідовності на патчі.

PnL - Profit and Loss - прибуток і збиток торгової стратегії.

ProbSparse Attention - Probabilistic Sparse Attention - розріджений механізм уваги, що використовується в архітектурі Informer.

PyTorch - програмний фреймворк для побудови та навчання моделей глибокого навчання.

SOTA - State of the Art - найкращий або один із найкращих відомих результатів у певній задачі на момент дослідження.

TST - Time Series Transformer - трансформерна модель для роботи з часовими рядами.

USDT - Tether USD - стейблкоїн, прив'язаний до долара США.

Анотація

Робота присвячена дослідженню застосування моделей на основі механізму уваги для передбачення фінансових часових рядів. Метою роботи є оцінка ефективності архітектури Transformer та її модифікацій у задачі багатогоризонтного прогнозування динаміки фінансових активів за історичними біржовими даними.

У межах дослідження було розглянуто моделі Transformer, Informer, PatchTST, iTransformer та FEDFormer, а також запропоновано власну гібридну архітектуру FEiTransformer, що поєднує інверсію вхідного багатовимірного представлення з елементами частотної декомпозиції. Для експериментальної частини було підготовлено високочастотний датасет криптовалютного ринку, агрегований до п'ятихвилинних інтервалів. Моделі навчалися у двох варіантах представлення даних: одновимірному та багатовимірному.

Оцінювання виконувалося із застосуванням walk-forward підходу, що дозволило зберегти часову структуру даних та наблизити експеримент до реального сценарію використання прогнозних моделей. Якість моделей аналізувалася за середньоквадратичною похибкою, точністю визначення напрямку руху ціни, стабільністю результатів між фолдами та практичною придатністю прогнозів.

Результати дослідження показали, що моделі на основі механізму уваги здатні виявляти певні закономірності у фінансових часових рядах, однак їхні передбачення часто мають менше стандартне відхилення порівняно з фактичними значеннями. Це свідчить про схильність моделей формувати згладжені прогнози, зосереджені поблизу нуля. Найкращою з практичної точки зору виявилася модель, яка найменше страждала від цього обмеження та формувала більш виразні прогностичні сигнали.

Ключові слова: Transformer, Informer, PatchTST, iTransformer, FEDFormer, FEiTransformer, фінансові часові ряди, механізм уваги, walk-forward, глибинне навчання.

Вступ

Аналіз і моделювання фінансових часових рядів є важливою сферою академічних досліджень, яка має значний потенціал для практичного застосування у різноманітних галузях, таких як прогнозування цін активів, оцінка ризиків, алгоритмічна торгівля та управління портфелем. Економіка є надзвичайно складною системою, в якій безліч учасників — від приватних осіб до державних інституцій — перебувають у постійній взаємодії. Внаслідок цього фінансові дані характеризуються високим рівнем шуму, нестационарністю та складною часовою структурою, що робить їхнє моделювання особливо складним завданням. Крім того, такі часові ряди суттєво залежать від зовнішніх факторів, зокрема макроекономічних показників та новинного фону.

Традиційні підходи до вирішення цієї задачі значною мірою спираються на коректність початкових припущень, побудованих на експертних знаннях про економічні процеси. Разом із цим, зростаюча доступність фінансових даних та обчислювальних потужностей зумовила розвиток нового покоління моделей, які використовують алгоритми глибинного навчання та дають змогу прогнозувати динаміку економічних показників, вивчаючи закономірності безпосередньо з історичних даних.

Серед архітектур глибинних нейронних мереж особливе місце посідає Трансформер, центральним елементом якого є механізм уваги. Такі моделі демонструють високу ефективність при аналізі даних, що мають послідовну структуру — оригінально архітектура була розроблена для розв'язання задач обробки природної мови. Механізм уваги дозволяє розпізнавати складні залежності навіть між елементами, що знаходяться далеко один від одного, що робить такі архітектури надзвичайно привабливими для роботи з часовими рядами, зокрема фінансовими.

Актуальність даної теми обумовлена необхідністю дослідження ефективності трансформерних архітектур у задачах прогнозування фінансових часових рядів, а також аналізу факторів, що впливають на їхню точність. За останні роки дослідниками було запропоновано численні варіанти Трансформерів, успішно адаптовані для прогнозування, виявлення аномалій та класифікації часових рядів. Водночас залишається чимало відкритих викликів — зокрема, адаптація механізму уваги до специфіки фінансових даних: ефективне моделювання мінливої волатильності, захоплення сезонності та режимних переключень ринку, а також

забезпечення обчислювальної ефективності при обробці високочастотних фінансових даних. Особливий інтерес становить порівняння різних модифікацій трансформерних архітектур та дослідження можливостей їх поєднання у гібридні моделі.

Метою даної роботи є дослідження ефективності архітектури нейронних мереж Transformer та її модифікацій у задачі передбачення вартості фінансових активів за історичними даними.

Об'єктом дослідження є процес прогнозування фінансових часових рядів за допомогою глибинних нейронних мереж.

Предметом дослідження є архітектури на основі механізму уваги та їх модифікації, адаптовані для роботи з фінансовими часовими рядами.

Для досягнення поставленої мети використовувалися такі методи дослідження: методи глибинного навчання для побудови прогнозних моделей, методи обробки та підготовки фінансових часових рядів, експериментальні методи оцінки ефективності моделей із застосуванням Walk-Forward підходу, а також методи статистичного аналізу результатів.

Наукова новизна одержаних результатів полягає у порівняльному аналізі ефективності сучасних модифікацій трансформерних архітектур (Informer, PatchTST, iTransformer, FEDFormer) у задачі прогнозування фінансових часових рядів. Зокрема, спроектовано власну гібридну архітектуру FEiTransformer що поєднує переваги декількох розглянутих модифікацій, а також досліджено вплив використання багатоканального представлення даних на ефективність прогнозування.

Практичне значення одержаних результатів полягає у можливості використання отриманих висновків для побудови систем алгоритмічної торгівлі та оцінки доцільності застосування трансформерних моделей у фінансовій сфері. Результати роботи можуть бути застосовані при розробці систем прогнозування фінансових ринків, а також у навчальному процесі при вивченні методів глибинного навчання та аналізу часових рядів.

Робота складається із трьох розділів. У першому розділі розглянуто теоретичні основи моделювання фінансових часових рядів, архітектуру Трансформера та механізм уваги, а також сучасні модифікації трансформерних архітектур, адаптовані для роботи з часовими рядами. У другому розділі описано процес збору та підготовки даних, реалізацію

моделей, їх навчання та оцінку. Третій розділ присвячено аналізу отриманих результатів

Постановка задачі:

- Виконати аналіз сучасних методів прогнозування фінансових часових рядів та архітектур глибоких нейронних мереж на основі механізму уваги.
- Розглянути існуючі модифікації трансформерних архітектур, адаптовані для роботи з часовими рядами (Informer, PatchTST, iTransformer, FEDFormer).
- Спроекувати власну гібридну архітектуру Frequency Enhanced Inverted Transformer (FEiTransformer), що поєднує переваги декількох розглянутих модифікацій.
- Здійснити збір та підготовку фінансових даних для навчання моделей.
- Реалізувати та навчити розглянуті моделі у двох варіантах представлення даних: простому та багатоканальному.
- Провести експериментальну оцінку ефективності моделей із застосуванням Walk-Forward підходу та порівнянням із базовою статистичною моделлю.
- Виконати порівняльний аналіз отриманих результатів та зробити висновки щодо доцільності використання архітектури Transformer для прогнозування фінансових часових рядів.

1 Огляд наукових досліджень

Основна проблема застосування Transformer-моделей до фінансових часових рядів полягає в самій природі даних. На відміну від тексту, де окремий токен має відносно насичене семантичне значення, окрема точка фінансового ряду — наприклад денна ціна активу — майже не несе самостійного змісту. Фінансові ряди є шумними, нестаціонарними, залежать від зовнішніх подій, новин, макроекономіки та змін ринкового режиму. Через це закономірності, вивчені на тренувальній вибірці, можуть швидко втратити актуальність у майбутньому, а збільшення розміру Transformer-моделі не обов'язково покращує результат — навпаки, може призводити до деградації прогнозної здатності та зайвих обчислювальних витрат.

Ще одна проблема полягає в тому, що класичний механізм уваги був розроблений передусім для послідовностей, елементи яких мають відносно насичене змістове представлення. У фінансовому часовому ряді окрема точка, наприклад ціна активу за один торговий день, часто не має достатнього самостійного змісту. Якщо подавати до моделі довгу послідовність таких слабкоінформативних точок, механізм уваги може розсіюватися на випадковий ринковий шум замість того, щоб виділяти справді важливі залежності. У результаті частина обчислень стає надлишковою: модель не фокусується на ключових історичних подіях, а фактично змішує вхідні дані в узагальнене представлення з обмеженою прогностичною цінністю.

Окремим викликом є обчислювальна складність. У стандартному механізмі self-attention кількість операцій зростає квадратично зі збільшенням довжини послідовності. Для фінансових задач це особливо критично, оскільки якісний прогноз часто потребує аналізу довгої історії або використання високочастотних даних. Чим довший часовий ряд подається на вхід, тим більшими стають вимоги до пам'яті та обчислювальних ресурсів. При цьому збільшення довжини послідовності або розміру моделі не гарантує покращення результатів, адже додаткові дані можуть містити не корисний сигнал, а переважно шум.

Важливою проблемою також є багатовимірність фінансових даних. На практиці модель працює не лише з однією ціною, а з багатьма ознаками: цінами відкриття, закриття, максимумом, мінімумом, обсягами торгів, технічними індикаторами та даними інших фінансових інструментів. Ці змінні мають різну природу, масштаб і статистичний розподіл, тому їх просте

об'єднання в один токен для кожного моменту часу може бути неефективним. У такому представленні модель може втрачати окремі взаємозв'язки між змінними, зокрема кореляції між активами або відкладені ефекти, коли одна подія впливає на різні показники з різною часовою затримкою.

Таким чином, при застосуванні Transformer-моделей до фінансових часових рядів виникає кілька фундаментальних викликів: шумність і нестационарність даних, слабка інформативність окремих часових точок, висока обчислювальна вартість роботи з довгими послідовностями та складність коректного представлення багатовимірних фінансових ознак. Саме ці обмеження зумовлюють необхідність адаптації класичної архітектури Transformer до специфіки часових рядів. Подальші підрозділи будуть присвячені розгляду підходів, які по-різному модифікують механізм уваги, структуру вхідних токенів або спосіб представлення сигналу для подолання зазначених проблем.

1.1 Патчування для вирішення проблеми семантичної бідності часових точок

На відміну від моделювання мови, послідовне збільшення розмірів нейромереж є неефективним підходом при прогнозуванні часових рядів. Збільшення розміру моделей часто призводить не до покращення результатів, а до деградації прогностичної здатності та надмірних обчислювальних витрат [1].

Причина полягає у фундаментальній різниці між текстовими даними та числовими часовими рядами. У задачах NLP, кожне слово-токен може нести в собі досить багато інформації, а механізм уваги чудово працює для стиснення інформації та виявлення абстрактних довгострокових залежностей. Натомість окрема точка часового ряду (наприклад, ціна за актив протягом одного біржового дня) майже не має самостійного змісту. Якщо "згодовувати" гігантській моделі тисячі таких точок, механізм уваги починає "розсіюватися" на глобальний ринковий шум.

Для того, щоб надати більше семантики кожному елементу послідовності, дослідники почали групувати окремі точки часового ряду і створювати більш «змістовні» токени. Модель **PatchTST** [2] розбиває часовий ряд на сегменти, які називаються "патчами" (patches). Далі ці патчі токенізуються і передаються як вхідні дані до стандартного енкодера Transformer.

Схожий метод застосовується для передавання зображень в трансформери: зображення розділяється на прямокутні патчі невеликого розміру – тобто замість груп точок часового ряду маємо групи пікселів на зображенні.

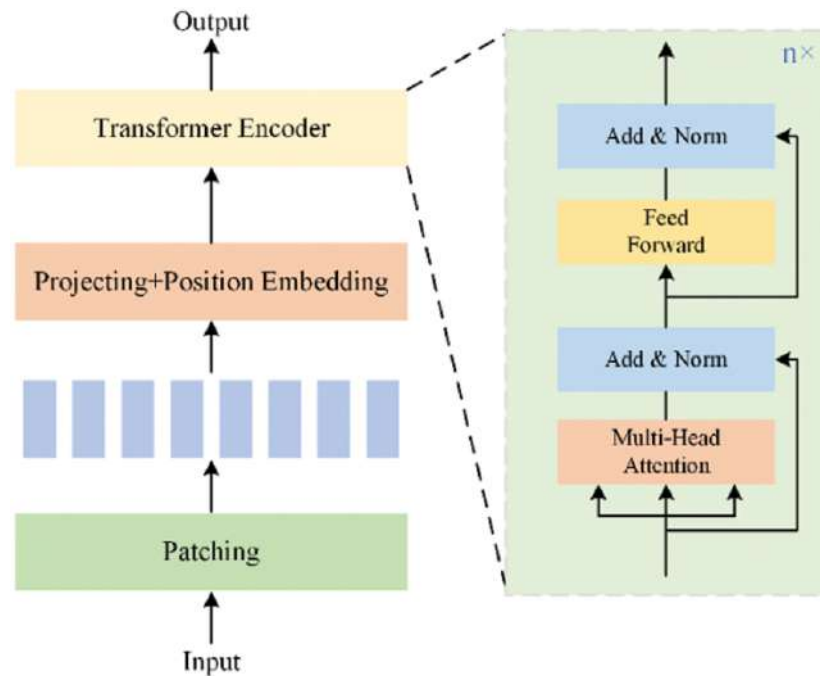


Рисунок 1.1 Структура мережі PatchTST

Такий підхід дає змогу класичній архітектурі Transformer вловлювати довгострокові залежності з набагато меншою кількістю токенів та меншими обчислювальними витратами, оскільки всі довжини послідовностей радикально зменшуються:

$$L^* = L/L_{patch} \quad (1.1)$$

Кількість математичних операцій, що виконуються в блоці уваги, квадратично залежить від довжини послідовності. Тому «патчування» дуже позитивно впливає на продуктивність моделей.

1.2 Модифікації механізму уваги для роботи з часовими рядами

Інший підхід базується на емпіричному спостереженні, що розподіл оцінок уваги у часових рядах має властивість «довгого хвоста». Це означає, що лише дуже невелика кількість пар «запит-ключ» (query-key) робить вагомий внесок у загальну увагу, тоді як більшість генерує тривіальні (майже рівномірні) значення, які можна ігнорувати без втрати якості.

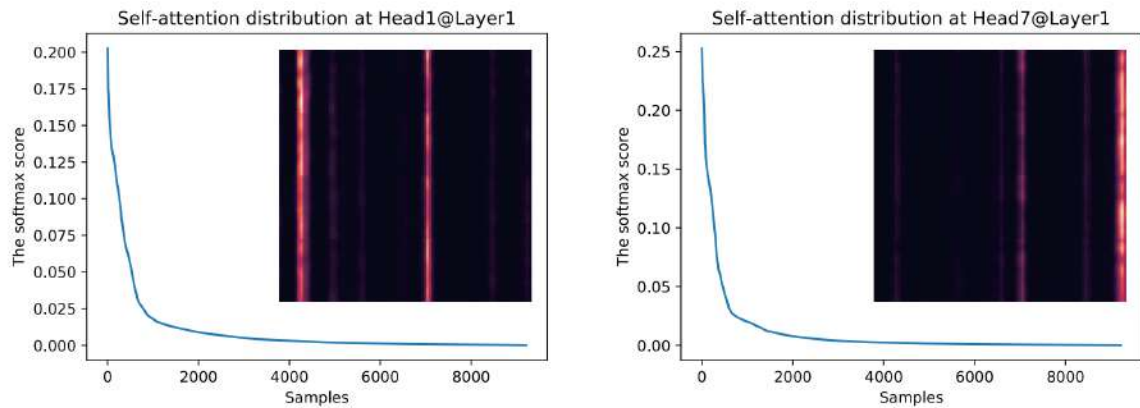


Рисунок 1.2 Розподіл оцінок softmax у класичному 4-шаровому Transformer, натренованому на датасеті ETTH

Розглянемо механізм уваги використаний в оригінальному Transformer. Після обчислення матриць ключів K , запитів Q і значень V для вхідної послідовності токенів, наступні представлення обчислюються за формулою:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad d - \text{розмірність токенів} \quad (1.2)$$

Якщо запит не має сильних кореляцій з жодним із ключів, скалярні добутки в матриці QK^T будуть приблизно однаковими. Застосування функції Softmax до рядка з однаковими значеннями перетворить їх на ідеально рівномірний розподіл ймовірностей.

Таким чином, рядок матриці $Softmax\left(\frac{QK^T}{\sqrt{d}}\right)$ буде близьким до масиву однакових дробів $\left[1/L_k, \dots, 1/L_k\right]$, де L_k – кількість ключів (довжина послідовності). Подальше множення цього рядка з матрицею значень V є еквівалентним до усереднення векторів значень $v_1, v_2, \dots, v_{L_k}$.

У такому випадку механізм уваги стає надлишковим та неінформативним, оскільки він не виділяє домінуючих залежностей чи корисних патернів між часовими кроками. Замість того, щоб сфокусуватися на важливих історичних подіях модель просто змішує всі вхідні дані в однорідну масу, що робить таку операцію марною для ефективного прогнозування.

Спеціальна адаптація механізму уваги – ProbSparse Attention, використана у моделі Informer [3], створена для того щоб ігнорувати ці «неінформативні» запити, для яких розподіл уваги близький до рівномірного розподілу.

Для цього, для кожного запиту q_i рахується спеціальна метрика:

$$M(q_i, K) = \max_j \left\{ \frac{q_i k_j^T}{\sqrt{d}} \right\} - \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{q_i k_j^T}{\sqrt{d}} \quad (1.3)$$

Чим більшою є ця розбіжність, тим більше розподіл уваги відхиляється від рівномірного, а отже — такий запит є більш «різноманітним» і має високий шанс містити домінуючі, критично важливі зв'язки.

На основі цієї метрики, обирається логарифмічна ($u = c \ln L$) кількість запитів і формується розріджена матриця $\bar{Q}$, що надалі використовується при обчисленні наступних значень.

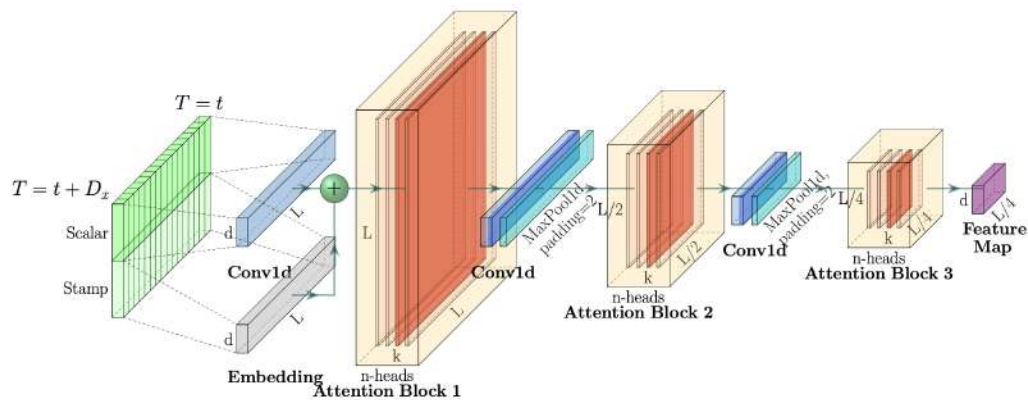


Рисунок 1.3 Структура мережі Informer

Завдяки такому «розрідженню» уваги обчислювальна та просторова складність механізму уваги зменшується з квадратичної $O(L^2)$ до

логарифмічної $O(L \times \log L)$, що дозволяє мережі ефективно обробляти набагато довші історичні послідовності.

Варто зазначити, що Informer – це не єдина спроба адаптувати і «пришвидшити» увагу для часових рядів.

1.3 Використання частотної декомпозиції для роботи з нестационарними і періодичними часовими рядами

Фінансові часові ряди, зокрема, характеризуються високою нестационарністю: ринкові тренди і волатильність постійно змінюються. В таких умовах, в задачах передбачення стандартні трансформери показують погані результати.

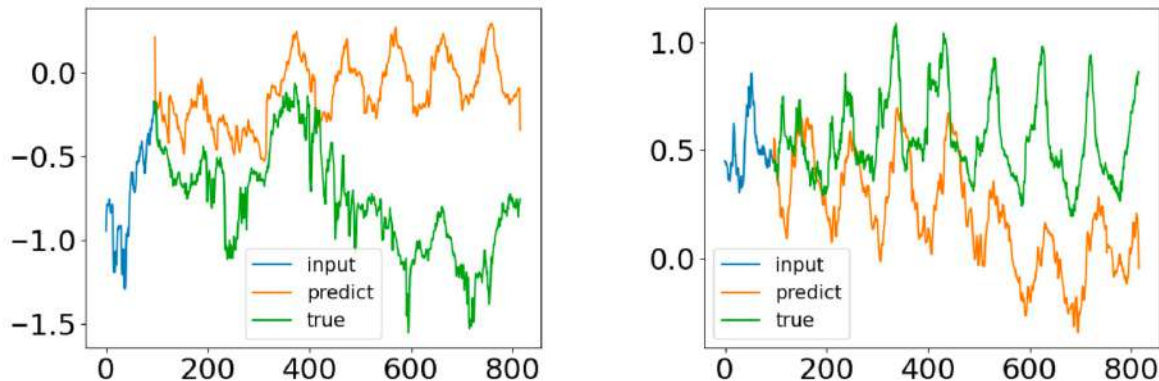


Рисунок 1.4 Розбіжність між передбаченнями Transformer і реальним рядом ETTm1

При роботі з нестационарними послідовностями із складними сезонними трендами часто застосовують частотний аналіз. За допомогою дискретного перетворення Фур'є сигнал розкладають на суму гармонічних складових, а вейвлет-перетворення дозволяє додатково локалізувати їх у часі. Таким чином, від аналізу в часовій області переходять до аналізу в частотній (або частотно-часовій) області.

Природньо, що дослідники спробували застосувати частотний аналіз при адаптації моделі Transformer до роботи з часовими рядами. Найпростіший підхід – це фільтрація вхідного сигналу на етапі препроцесингу. Після розкладу ряду на суму періодичних сигналів, деякі з складових можуть бути «відсіянні» - зазвичай високочастотні або з низькою амплітудою. При цьому на вхід енкодера передається вже «очищений» сигнал, що підвищує точність моделі.

Основний недолік такого підходу полягає у тому, що при такій фільтрації разом із шумом можна втратити важливу інформацію.

Налаштування такого фільтру – нетривіальна задача, що вимагає глибокого розуміння предметної області.

Значно цікавішими є адаптації архітектури Transformer, коли частотні перетворення стають невід’ємною частиною нейронної мережі. Модель FEDformer [4] пропонує спеціальний Decomposition Block (блок декомпозиції). Вхідний сигнал X_t розділяється на дві складові:

- плавний довгостроковий тренд T_t – по своїй суті це ковзне середнє ряду;
- сезонні коливання S_t , що є різницею між рядом і його трендом $X_t - T_t$.

Варто зазначити, що назви складових T_t і S_t співпадають з семантикою лише для першого шару мережі. На кожному наступному кроці, модель працює вже не з самим часовим рядом, а черговим внутрішнім представленнями послідовності.

Далі на кожен із цих підрядів чекає різний шлях в мережі. Трендова складова T_t проходить через звичайну лінійну проекцію (множення на матрицю, елементи якої є трендованими параметрами). Сезонні коливання S_t у свою чергу передаються на спеціальний блок уваги Frequency Enhanced Block (блок частотного посилення).

Саме тут до трендових відхилень застосовується швидке перетворення Фур’є (у варіації FEB-f) або вейвлет-перетворення (у варіації FEB-w). Після переходу в спектральну площину, алгоритм вибирає лише невелику підмножину домінуючих частот, які роблять найбільший внесок у загальну дисперсію сигналу. Далі, по аналогії з уже розглянутим ProbSparse Attention, увага застосовується виключно до цих відібраних частотних компонентів. Зворотній перехід до часової області виконується за допомогою зворотніх операцій: зворотне перетворення Фур’є (IFFT) чи зворотнє вейвлет перетворення (IDWT)

В кінці реконструюється наступне внутрішнє представлення вхідної послідовності:

$$\tilde{X} = \tilde{T}_t + \tilde{S}_t \quad (1.4)$$

Внесення елементів частотного аналізу до Transformer моделей дозволило якісно покращити їх результати, оскільки вони дозволяють

ефективно виділяти глобальні періодичні структури та відфільтровувати надлишковий високочастотний ринковий шум, що значно покращує здатність моделі фокусуватися на справжніх трендах і сезонності.

Побічним ефектом такої модифікації Transformer є радикальне зменшення обчислювальної і просторової складності при роботі алгоритму само-уваги, оскільки тепер обчислення матриць уваги відбуваються лише для обмеженої підмножини найважливіших (домінуючих) частот, а не для кожного окремого токена у довгій часовій послідовності.

1.4 Проблеми при роботі з багатовимірними часовими рядами та інверсія вхідних даних

Важливою характеристикою для фінансових часових рядів є їх багатовимірність. На біржі, ціна активу протягом певного періоду виражається чотирма величинами: ціною на момент відкриття та закриття періоду, а також мінімальною й максимальною ціною протягом нього. Окрім них, надзвичайно важливим показником є загальний обсяг торгів цим активом на біржі.

Надзвичайно корисними також є різноманітні індикатори технічного аналізу, що часто додаються до «сирих» навчальних даних для покращення результатів моделей. До таких технічних індикаторів відносяться різноманітні ковзні середні а також індекси, що дозволяють оцінити поточну волатильність чи силу тренду. Із кожним таким індикатором додається один чи декілька вимірів для точок часового ряду.



Рисунок 1.5 Аналіз індексу S&P 500 за допомогою різних технічних індикаторів

Ефективна обробка таких багатоканальних вхідних сигналів є ще одним викликом для моделей Transformer. Стандартний підхід, при якому вхідний токен формується із всіх змінних часового ряду, має проблеми.

По-перше, кожен атрибут для точки часового ряду може мати абсолютно різний фізичний зміст, одиниці виміру чи статистичний розподіл. Злиття всіх цих значень в один нероздільний токен призводить до того, що модель не може розрізняти взаємозв'язки між окремими змінними, адже механізм уваги застосовується вздовж часової осі і порівнює цілі токени між собою.

Також існує проблема відкладених подій. В реальному житті одна й та сама подія може вплинути на різні змінні з різною затримкою. Через «локальність» токенів для моделей такі кореляції стають неможливими для виявлення.

Творці моделі iTransformer [5] запропонували «інвертувати» класичний Transformer і кардинально змінити підхід при роботі з багатовимірними часовими рядами. Замість того, щоб створювати по одному токenu на кожен часовий крок, модель бере весь часовий ряд окремої змінної і перетворює його на один незалежний токен. Таким чином, тепер один токен, замість того щоб містити значення всіх змінних в окремий момент часу, містить всю історію однієї змінної.

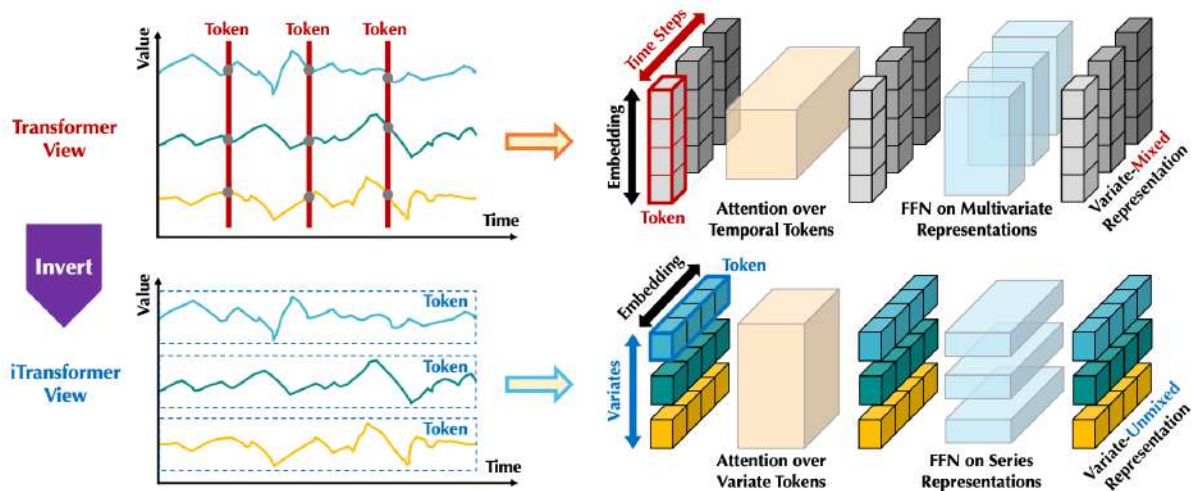


Рисунок 1.6 Відмінність представлення часового ряду у класичному і інвертованому Трансформерах

Окрім інверсії, усі компоненти Transformer залишились незмінними, але для задачі передбачення багатовимірних часових рядів отримана модель змогла досягти кращих результатів ніж моделі що застосовують складні частотні чи статистичні операції.

2 Практична частина

Мета практичної частини – порівняти розглянуті в минулому розділі моделі за ефективністю при передбаченні фінансових часових рядів.

В практичній частині роботи вихідний код із [6], [7], [8], [9] був використаний для тренування власних моделей глибинного навчання на історичних фінансових даних. Усі моделі доступні у двох варіантах: простий, що працює з одновимірним часовим рядом (тільки історія ціни цільового активу); та повний, в якому модель отримує на вхід багатовимірний часовий ряд з багатьма іншими показниками з біржі.

Окрім попередньо-розглянутих у практичній частині запропонована власна, гібридна архітектура, що поєднує в собі інверсію вхідних даних і блок частотної декомпозиції, що представлений в моделі FEDFormer.

2.1 Гібридна архітектура FEiTransformer

Розглянуті модифікації архітектури Transformer можна поділити на дві групи:

- зміна схеми роботи з вхідними даними (PatchTST і iTransformer);
- заміна внутрішніх компонентів Transformer модифікаціями, що надають моделі якісно нових можливостей з аналізу часових рядів (Informer і FEDFormer).

В практичній частині роботи було розроблену власну модель, що поєднує блок декомпозиції і застосування частотних перетворень, запропоновані розробниками FEDFormer [4] із інверсією, що показала SOTA результати при роботі з багатовимірними часовими рядами [5].

Кожен токен, що представляє історію одного атрибута розділяється на два токени:

- токен із трендовою складовою, побудований за допомогою ковзного середнього;
- токен із частотними характеристиками волатильного залишку, отриманим за допомогою швидкого перетворення Фур'є або вейвлет-перетворення.

Решта архітектури повністю аналогічна інвертованому Трансформеру, за виключенням подвоєної кількості токенів на вході.

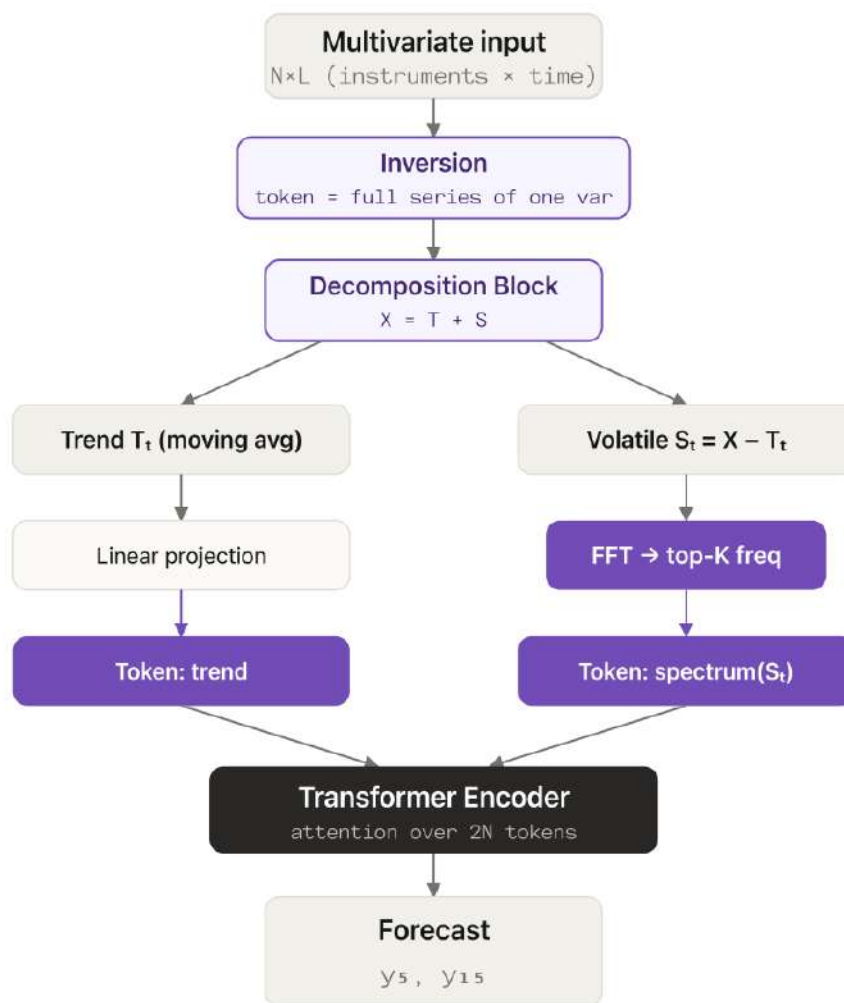


Рисунок 1.7 Схема гібридної моделі FETransformer

2.2 Опис задачі та тренувальних даних

В практичній частині роботи кожна з моделей: Transformer, PatchTST, Informer, iTransformer, FEDFormer і FEiTransformer була натренована в двох конфігураціях: одновимірній і багатовимірній. Перший варіант моделі працює з одновимірною часовою послідовністю і отримує на вхід лише історію зміни цільової величини (ціна цільового активу на момент закриття торгового періоду). Складніший варіант працює з багатовимірним часовим рядом, і окрім історії ціни цільового активу отримує інформацію про інші активи, а також додаткові показники із біржі, такі як об'єм торгів, кількість і сторона угод, тощо.

Моделі тренувались для задачі багатогоризонтного передбачення, ціль – ціна активу через 5 і 15 торгових періодів (P_{t+5} , P_{t+15}). Замість абсолютного значення ціни, моделі працювали з **log-returns**, що часто застосовуються для фінансових задач.

$$y_h = \log \left(\frac{P_{t+h}}{P_t} \right), h \in \{5, 15\}. \quad (2.1)$$

Перехід до відносної зміни ціни вирішує проблему нормалізації, а застосування функції логарифму – робить розподіл значень симетричним відносно нуля.

Для перевірки адекватності натренованих моделей було побудовано «наївну» статистичну модель, що просто повертала середній приріст за кожен часовий горизонт на розглянутому відрізку.

Під час першої ітерації роботи, у якості тренувальних даних було обрано добові показники індексу S&P500 починаючи з 2000 року. У багатовимірному варіанті, також використовувались 28 інших фінансових показників, що в дають моделі повну картину стану фінансових ринків.

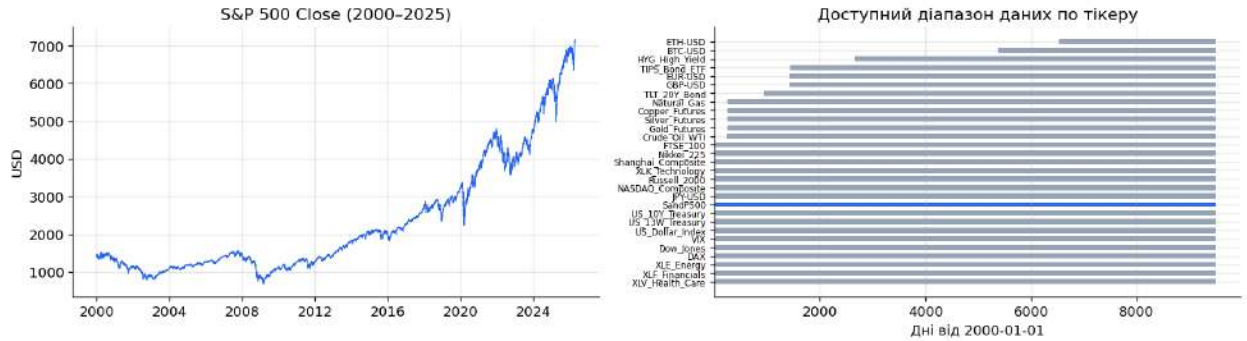


Рисунок 2.1 Візуалізація даних використаних для першої ітерації

Дані було отримано за допомогою сервісу Yahoo Finance, а саме Python API уfinance. В результаті дата сет містив майже 10 000 часових точок.

Такої кількості даних для моделей Transformer виявилось недостатньо, оскільки навіть при мінімізації розміру моделей за допомогою гіперпараметрів моделі мають тисячі тренуваних параметрів. Тобто загальний принцип «1 параметр – 10 екземплярів в датасеті» при роботі з добовими даними не виконується.

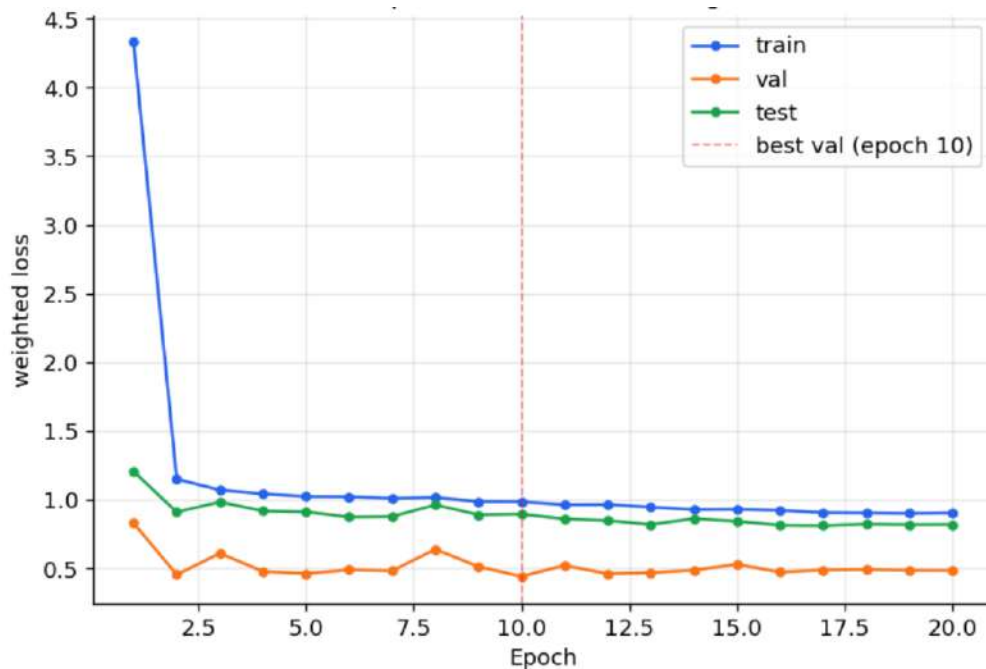


Рисунок 2.2 Криві навчання моделі FEiTransformer на добових цінах S&P500

Візуалізація тренувального процесу підтвердила недостатність розміру тренувальної вибірки. Всього за декілька епох навчання моделі були здатні «запам'ятати» тренувальну вибірку. Подальше тренування лише трохи знижувало втрати на навчальній вибірці, а значення втрат на валідаційній і тестовій вибірках змінювалось випадковим чином.

У свою чергу, збільшення розміру вибірки завдяки збільшенню часового вікна так само не виглядає доцільним, оскільки дані про поведінку ринків десятки років тому мають сумнівну цінність для моделей, що повинні аналізувати ринок у реальному часі. Як висновок – для адекватного тренування необхідно збільшувати частоту точок часового ряду.

Датасет «jorijnsmit/binance-full-history» доступний у відкритому доступі на порталі Kaggle містить щохвилинну історію зміни більше тисячі торгових пар на криптовалютній біржі Binance за 5-ти річний період. Програмним шляхом хвилинні японські свічки було об'єднано в менш волатильні 5 хвилинні.

Для передбачення було обрано ціну пари з найбільшою ліквідністю: BTC-USDT, а для багатовимірного варіанта моделі було обрано ще 4 високоліквідні пари що представлені довгою історією в датасеті.

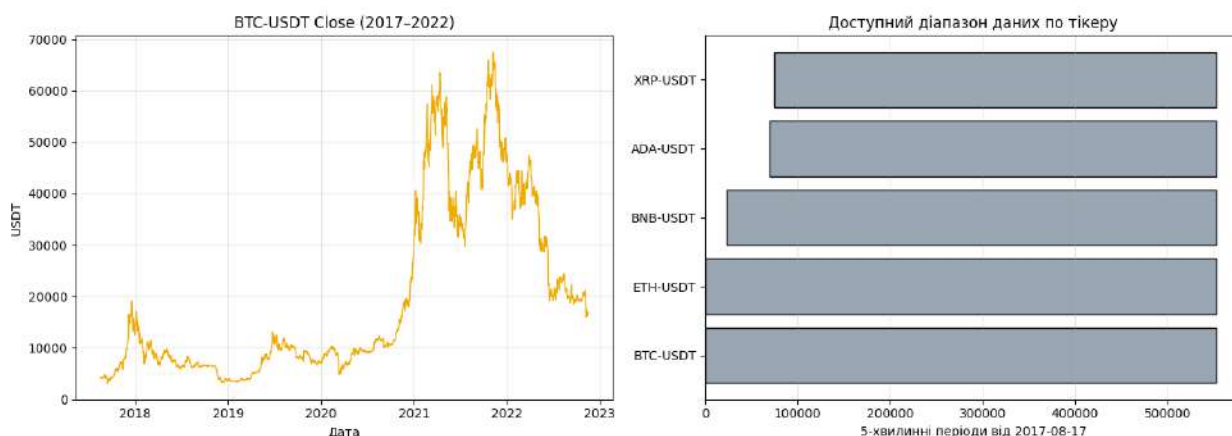


Рисунок 2.3 Візуалізація використаних даних для фінального навчання

В результаті для навчання було отримано часовий ряд довжиною близько 500 тисяч точок. У багатовимірному варіанту для кожної торгової пари було використано 9 атрибутів доступних в датасеті, тобто кожну часову точку модель сприймає через 45 (9x5) значень.

У межах підготовки даних сирі біржові показники були перетворені у форму, більш придатну для навчання моделей часових рядів. Початкові значення цін, обсягів торгів та кількості угод мають різну природу й масштаб: ціна Bitcoin може вимірюватися десятками тисяч, обсяг торгів — сотнями або мільйонами, а кількість угод — цілими числами. Якщо подавати такі дані безпосередньо в модель, вона може надмірно орієнтуватися на абсолютні величини, а не на реальну динаміку ринку. Тому основна мета препроцесингу полягає в тому, щоб перейти від сирих значень до ознак, які описують зміну стану ринку в часі.

Для цінових атрибутів застосовується перетворення у логарифмічні різниці, тобто фактично розраховуються логарифмічні прибутковості. Це дозволяє моделі працювати не з абсолютною ціною активу, а з відносною зміною ціни між сусідніми часовими кроками. Такий підхід є особливо важливим у фінансових часових рядах, оскільки абсолютна ціна активу сама по собі часто менш інформативна, ніж напрям і сила її зміни. Крім того, логарифмічні прибутковості роблять ряди більш стабільними та краще порівнюваними між різними торговими парами.

$$\widetilde{X}_t = \log(X_t) - \log(X_{t-1}) \quad (2.2)$$

Для атрибутів що кількісно відображають об'єми торгів на біржі логарифмічне стиснення з подальшим обчисленням різниць. Логарифм зменшує вплив надзвичайно великих значень, які часто трапляються на криптовалютному ринку під час різких рухів ціни або сплесків активності. Після цього різниця між сусідніми значеннями дозволяє моделі бачити не просто рівень торгової активності, а її зміну: зростання або спад обсягів, посилення купівельного тиску, зміну кількості угод тощо. Саме такі сигнали можуть бути корисними для прогнозування майбутньої поведінки ціни.

$$\widetilde{X}_t = \log(1 + X_t) - \log(1 + X_{t-1}) \quad (2.3)$$

Після цих перетворень дані додатково нормалізуються, щоб усі ознаки мали порівнюваний масштаб. Це допомагає нейронним моделям навчатися стабільніше, оскільки жодна колонка не домінує над іншими лише через більші числові значення. У результаті модель отримує не сирій набір біржових чисел, а узгоджене представлення ринкової динаміки: зміни цін, зміни обсягів, зміни активності учасників ринку та співвідношення цих змін між різними торговими парами. Для задачі прогнозування фінансових часових рядів це є принципово важливим, оскільки модель має виявляти

закономірності саме в поведінці ринку, а не в абсолютних масштабах окремих активів.

2.3 Конфігурація і оцінювання моделей

Оскільки розглянуті моделі мають досить велику кількість гіперпараметрів, а саме навчання вимагає великої кількості обчислювальних ресурсів та часу, повний перебір усіх комбінацій в просторі пошуку є нереалістичною задачею.

Був розроблений стохастичний двох фазовий пошук гіперпараметрів. Під час першої фази обирається лише невелика кількість можливих комбінацій і для кожної відбувається тренування і оцінка моделі. У підсумку обирається найкраща конфігурація, в навколо якої виконується повний ортогональний пошук (зміна тільки одного параметра за раз) оптимальних конфігурацій в другій фазі. Таким чином було значно зменшено кількість циклів тренування необхідних для пошуку найкращої конфігурації.

Для базової Transformer-моделі (і інвертованої версії) підбирались основні параметри архітектури encoder-a: *d_model*, *n_heads*, *n_layers*, *dropout* і *lr*. Параметр *d_model* визначає розмір внутрішнього векторного представлення, у яке модель перетворює вхідні часові ознаки. Чим більшим є це значення, тим більше інформації модель потенційно може закодувати, але тим вищий ризик перенавчання. *n_heads* задає кількість attention-heads, тобто кількість паралельних “поглядів”, через які модель аналізує залежності між часовими кроками. *n_layers* визначає глибину моделі — кількість послідовних encoder-блоків. *dropout* використовується як регуляризація, щоб зменшити перенавчання, випадково вимикаючи частину нейронних зв’язків під час тренування. *lr*, або *learning rate*, визначає швидкість оновлення ваг моделі під час оптимізації.

Для PatchTST, окрім базових параметрів, важливими є *patch_len*, *stride* і *pool*. Параметр *patch_len* визначає довжину фрагмента часового ряду, який модель розглядає як один патч. *stride* задає крок між сусідніми патчами: менше значення дає більше перекриття між фрагментами, а більше - зменшує кількість патчів і робить модель обчислювально легшою. Параметр *pool* визначає, як агрегується вихід моделі перед фінальним прогнозом: може використовуватись або представлення цільового каналу, або усереднення інформації з усіх каналів.

Для Informer унікальними параметрами є *factor* і *distil*. Параметр *factor* пов’язаний із механізмом ProbSparse attention і впливає на те, наскільки сильно модель скорочує кількість attention-зв’язків, залишаючи лише

найбільш інформативні. Параметр *distil* визначає, чи застосовується між шарами механізм дистиляції, тобто поступове зменшення довжини послідовності. Його мета — залишити найбільш важливу інформацію та зменшити обчислювальну складність моделі.

Для FEDformer унікальними були параметри *modes*, *version*, *moving_avg*, *wavelet_c* і *wavelet_k*. Параметр *version* визначає, який тип частотного представлення використовується: Fourier або Wavelets. *modes* задає кількість частотних компонентів, які використовуються в Fourier-представленні. *moving_avg* відповідає за розділення ряду на трендову та залишкову компоненти за допомогою ковзного середнього. Параметри *wavelet_c* і *wavelet_k* пов'язані з wavelet-перетворенням і контролюють особливості частотно-часового розкладу, коли модель працює у wavelet-режимі.

Насамкінець, гібридна модель FEiTransformer поєднує гіперпараметри властиві FEDFormer та iTransformer, тому простір пошуку для неї є найбільшим.

Для остаточного оцінювання моделей використовувався підхід ***walk-forward evaluation***, який імітує реальний сценарій застосування прогнозної моделі до фінансового часового ряду. На відміну від випадкового поділу даних на навчальну та тестову вибірки, у цьому підході зберігається природний часовий порядок спостережень: модель навчається лише на історичних даних, а перевіряється на наступному, ще не використаному часовому проміжку. Це особливо важливо для фінансових даних, оскільки використання майбутньої інформації під час навчання або нормалізації призвело б до некоректно завищеної оцінки якості моделі.

У роботі було використано 4 walk-forward фолди. У кожному наступному фолді навчальне вікно розширювалося, а тестування виконувалося на наступному шестимісячному періоді. Перший фолд використовував дані від початку спільного періоду датасету до 31 грудня 2020 року, а тестування проводилося на періоді з 1 січня до 30 червня 2021 року. Другий фолд розширював навчальну вибірку до 30 червня 2021 року і тестував модель на другій половині 2021 року. Третій фолд використовував історію до 31 грудня 2021 року і тестувався на першій половині 2022 року. Четвертий фолд навчав модель на даних до 30 червня 2022 року, а тестування виконувалося на періоді з 1 липня 2022 року до кінця доступного датасету, тобто до 16 листопада 2022 року.

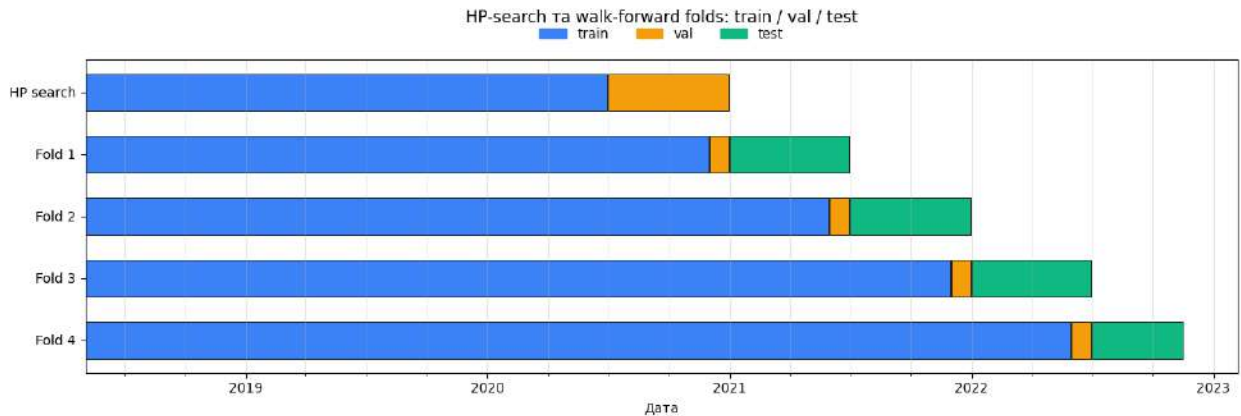


Рисунок 2.4 Візуалізація використання даних на різних етапах навчання

Всередині кожного фолду навчальний інтервал додатково поділявся на власне навчальну та валідаційну частини. Для валідації використовувалися останні приблизно 30 днів навчального проміжку, тоді як усі попередні дані використовувалися для оновлення параметрів моделі. Такий підхід дозволяв застосовувати ранню зупинку без звернення до тестової вибірки. Іншими словами, тестові дані залишалися повністю відкладеними до моменту фінальної оцінки моделі в межах відповідного фолду.

Для кожного фолду модель створювалася заново з обраною конфігурацією гіперпараметрів і навчалася лише на даних, доступних на цей момент часу. Після завершення навчання найкращий стан моделі, визначений за валідаційною функцією втрат, використовувався для прогнозування на тестовому проміжку. Отримані метрики обчислювалися окремо для кожного тестового періоду, після чого результати агрегувалися через середнє значення та стандартне відхилення між фолдами. Це дозволяло оцінити не лише середню якість моделі, а й стабільність її роботи на різних ринкових періодах.

Застосування walk-forward оцінювання є принципово важливим для задачі прогнозування фінансових часових рядів, оскільки ринкові дані не є незалежними та однаково розподіленими у класичному сенсі. Статистичні властивості ринку можуть змінюватися з часом: періоди високої волатильності, трендові фази, боковий рух або різкі зміни ліквідності можуть суттєво впливати на поведінку моделей. Тому перевірка на кількох послідовних майбутніх інтервалах дає більш реалістичне уявлення про здатність моделі узагальнювати закономірності, виявлені на історичних даних, на нові періоди.

Окрім середньоквадратичної похибки оцінювалась також точність визначення напрямку Ці метрики розраховувалися окремо для кожного горизонту прогнозування. Це дозволяло оцінити не лише числову близькість прогнозу до фактичного значення, а й здатність моделі правильно визначати напрям майбутнього руху ціни на різних горизонтах передбачення.

2.4 Тренування моделей

Навчання моделей виконувалося з використанням фреймворку PyTorch. Для всіх досліджуваних архітектур застосовувався єдиний підхід до організації навчального процесу, що забезпечувало коректність подальшого порівняння результатів. На вхід моделі подавалося фіксоване історичне вікно часового ряду довжиною 60 спостережень. Оскільки дані були попередньо агреговані до п'ятихвилинного інтервалу, таке вікно відповідало приблизно п'яти годинам історичних даних. Цільовими значеннями були майбутні логарифмічні прибутковості для двох горизонтів прогнозування: через 5 та 15 часових кроків, що відповідало прогнозу приблизно на 25 та 75 хвилин уперед.

Оптимізація параметрів моделей здійснювалася за допомогою алгоритму AdamW. Цей оптимізатор є модифікацією Adam і додатково враховує регуляризацію ваг через механізм weight decay. Значення weight decay було зафіксоване на рівні 10^{-3} , що використовувалося для зменшення ризику перенавчання моделей. Швидкість навчання (learning rate) не була сталою для всіх експериментів, а підбиралася під час пошуку гіперпараметрів. У межах цього пошуку розглядалися значення 10^{-3} та 5×10^{-4} .

Для керування швидкістю навчання також застосовувався планувальник CosineAnnealingLR. Його призначення полягає у поступовій зміні швидкості навчання протягом тренування за косинусним законом. Такий підхід дозволяє на початкових етапах виконувати більш суттєві оновлення параметрів моделі, а ближче до завершення навчання — поступово зменшувати величину цих оновлень. Це сприяє стабільнішій збіжності та точнішому уточненню знайденого розв'язку.

Як функція втрат використовувалася середньоквадратична похибка, нормалізована за стандартним відхиленням цільової змінної. Іншими словами, помилка прогнозу для кожного горизонту масштабувалася відповідно до дисперсії відповідного цільового ряду, обчисленої на навчальній вибірці. Такий підхід дозволяв збалансувати внесок різних горизонтів прогнозування у загальну функцію втрат. Це є важливим, оскільки цільові значення для різних горизонтів можуть мати різну варіативність, і без нормалізації горизонт із більшою дисперсією міг би непропорційно впливати на процес оптимізації.

Для запобігання перенавчанню використовувався механізм ранньої зупинки. Після кожної епохи модель оцінювалася на валідаційній вибірці, а отримане значення функції втрат порівнювалося з найкращим попереднім результатом. Якщо протягом заданої кількості епох покращення не спостерігалось, процес навчання завершувався достроково. Після цього для подальшої оцінки використовувався стан моделі, що відповідав найкращому значенню валідаційної функції втрат. На етапі пошуку гіперпараметрів використовувалося коротше навчання з максимальною кількістю 10 епох і терпимістю 4 епохи без покращення. Для фінального оцінювання в межах walk-forward підходу використовувалося до 20 епох із терпимістю 5 епох.

Підбір гіперпараметрів виконувався на окремому часовому проміжку, що не перетинався з тестовими інтервалами walk-forward оцінювання. Навчальна частина для цього етапу охоплювала історичні дані до кінця червня 2020 року, а валідаційна частина - період з липня до грудня 2020 року. Такий розподіл відповідає вимогам задач прогнозування часових рядів, оскільки модель навчається лише на даних, які передують валідаційному періоду, і не отримує доступу до майбутньої інформації.

3 Результати

У цьому розділі представлені результати, отримані після застосування натренованих моделей на тестовій вибірці. Вся тестова вибірка покриває період із початку 2021 до кінця наявних даних (листопад 2023) і поділяється на 4 частини відповідно до walk-forward фолдів. До кожної з частин застосовувалась відповідна модель, тренувана лише на попередній історії.

Перша візуалізація для кожної сім'ї моделей присвячена середньоквадратичній похибці передбачення та точності вгадування напрямку руху ціни для кожного тестового фолду. Для наочності результати обох варіацій представлені на одному графіку разом з результатами статистичної моделі. Це дозволяє оцінити перевагу, яку додаткові атрибути у вхідних даних надають кожній моделі, а також адекватність моделей порівняно з “наївною” версією.

Також для кожної варіації моделей і горизонту прогнозування було побудовано візуалізацію розподілу прогнозів на тестовій вибірці. Така візуалізація допомагає краще зрозуміти поведінку моделі на тестовій вибірці а також оцінити близькість розподілу оцінок до розподілу справжніх історичних даних.

Для середньої квадратичної похибки передбачення та точності вгадування напрямку руху ціни було побудовано двовимірні площину, на якій розміщені всі моделі. Вісь ординат відповідає за розмір моделей (кількість тренованих параметрів). Такі візуалізації дозволяють порівняти ефективність моделей між собою, а також дослідити кореляцію між розміром моделей та метриками.

3.1 Результати моделі Transformer

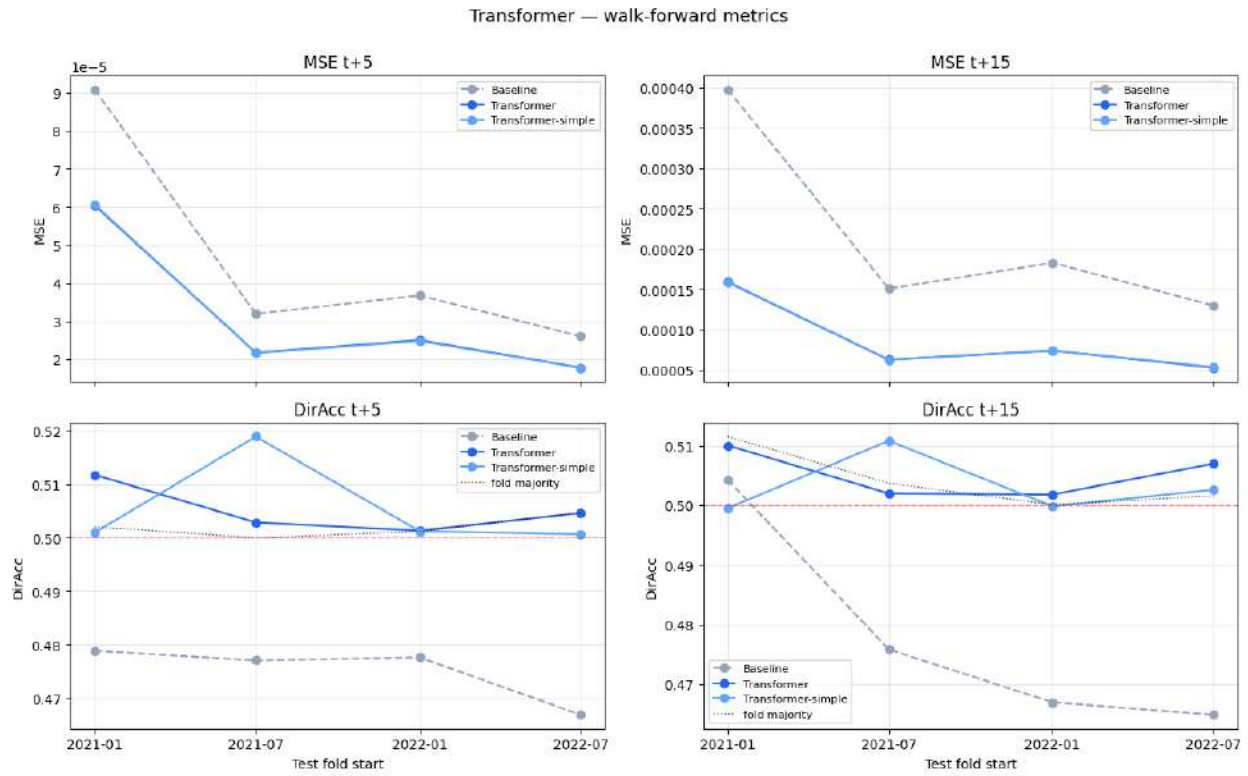


Рисунок 3.1 Walk-forward метрики для моделі Transformer

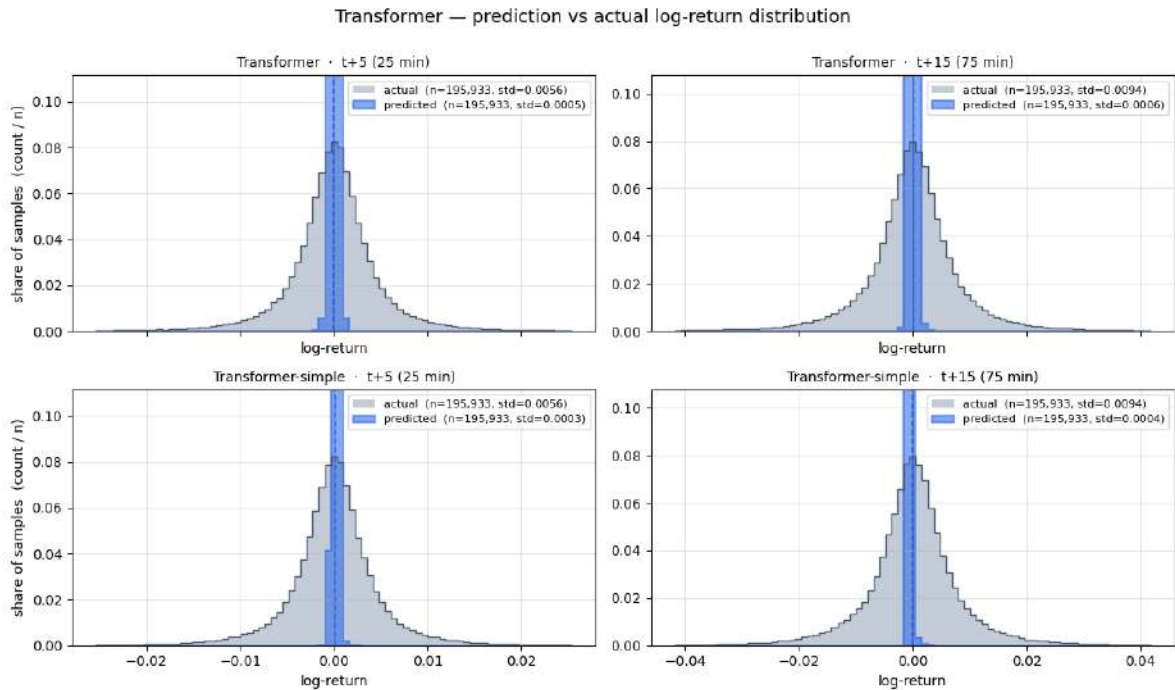


Рисунок 3.2 Розподіл тестових передбачень моделі Transformer

3.2 Результати моделі Informer

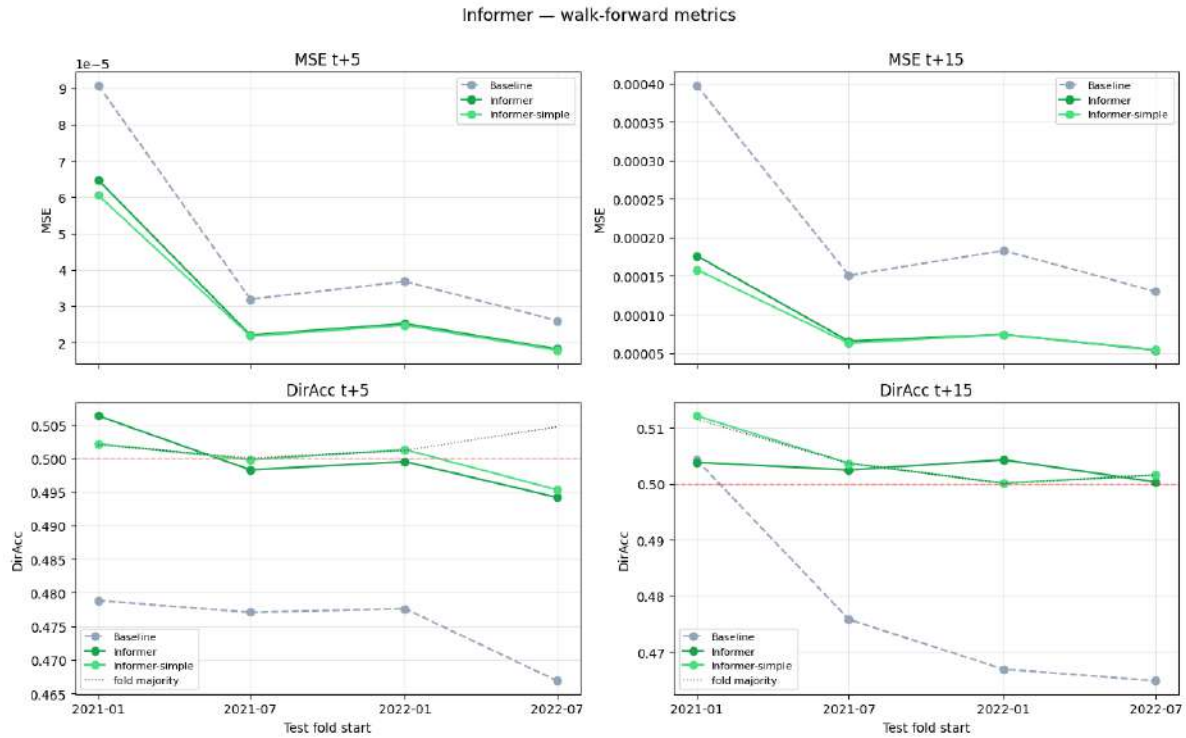


Рисунок 3.3 Walk-forward метрики для моделі Informer

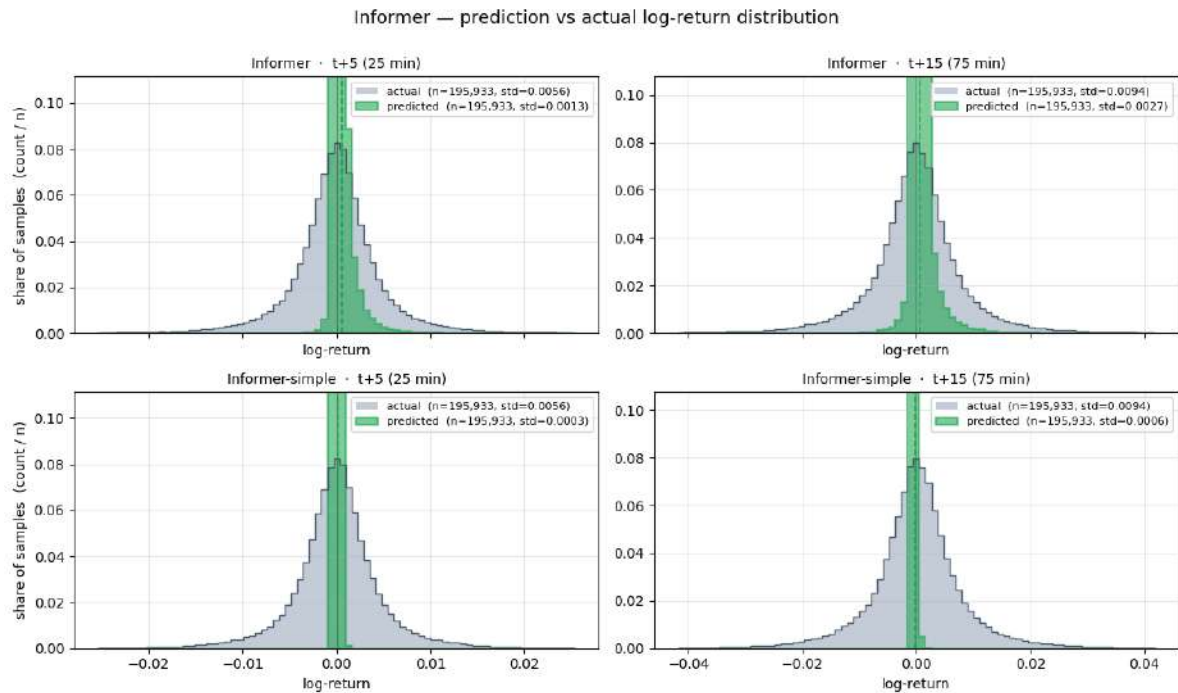


Рисунок 3.4 Розподіл тестових передбачень моделі Informer

3.3 Результати моделі PatchTST

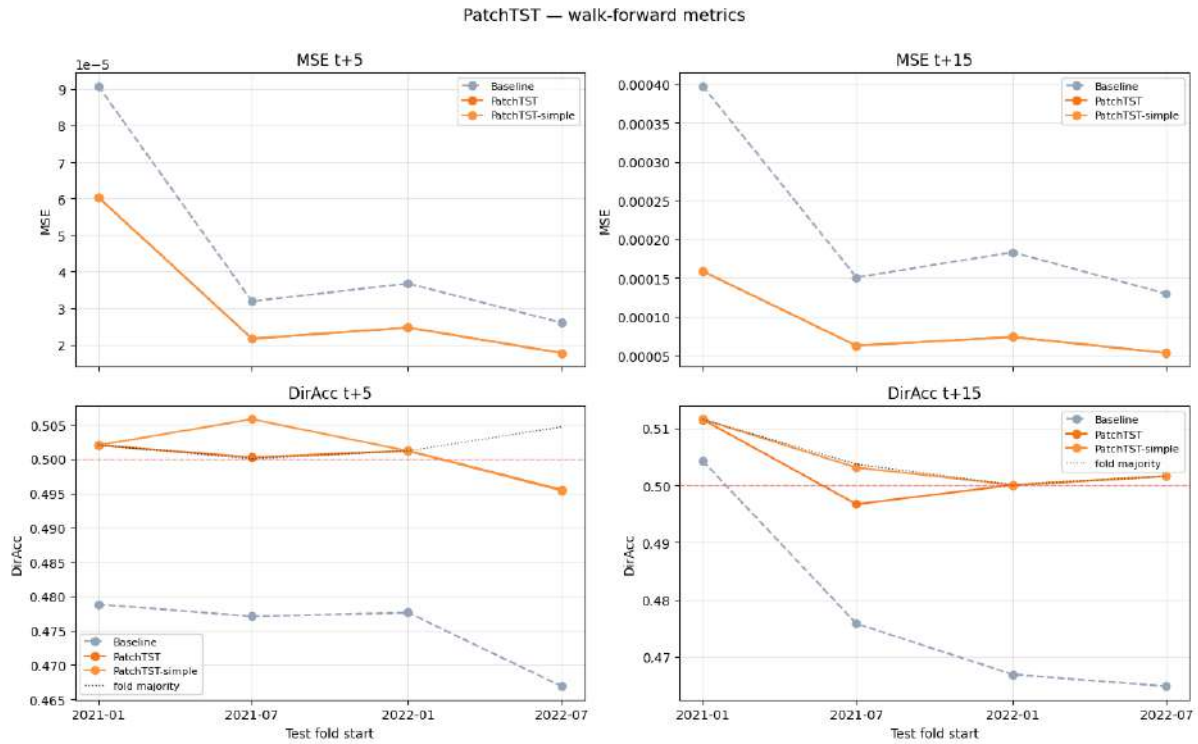


Рисунок 3.5 Walk-forward метрики для моделі PatchTST

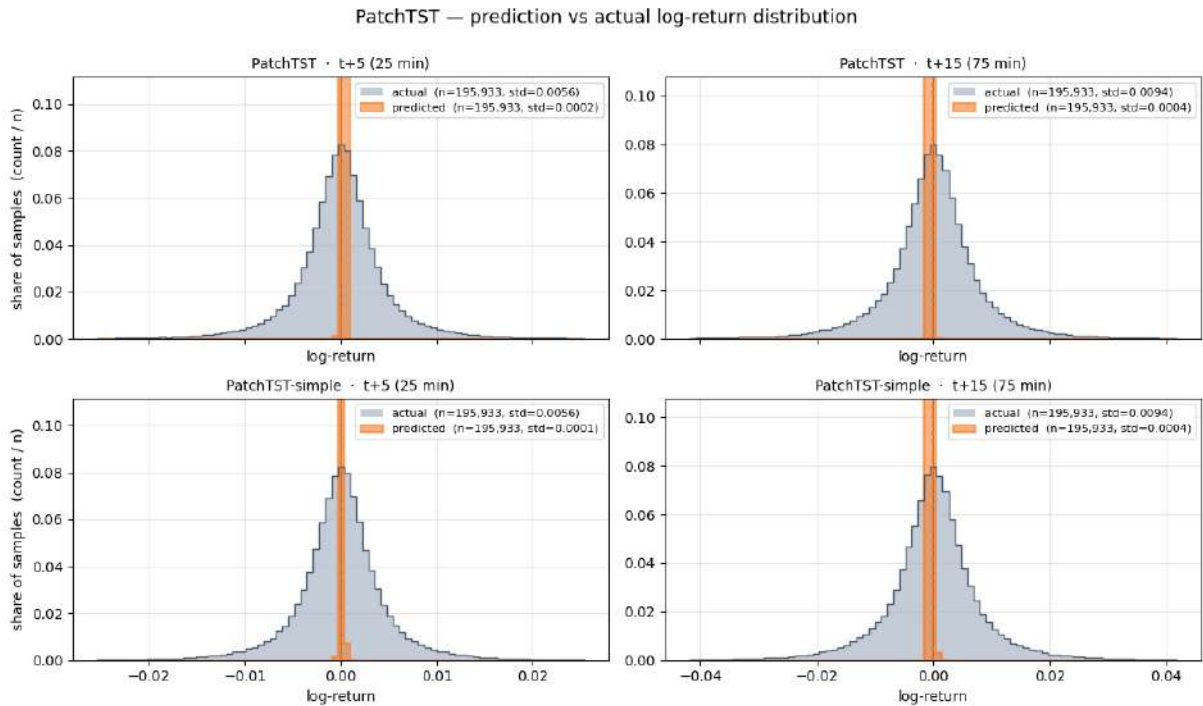


Рисунок 3.6 Розподіл тестових передбачень моделі PatchTST

3.4 Результати моделі iTransformer

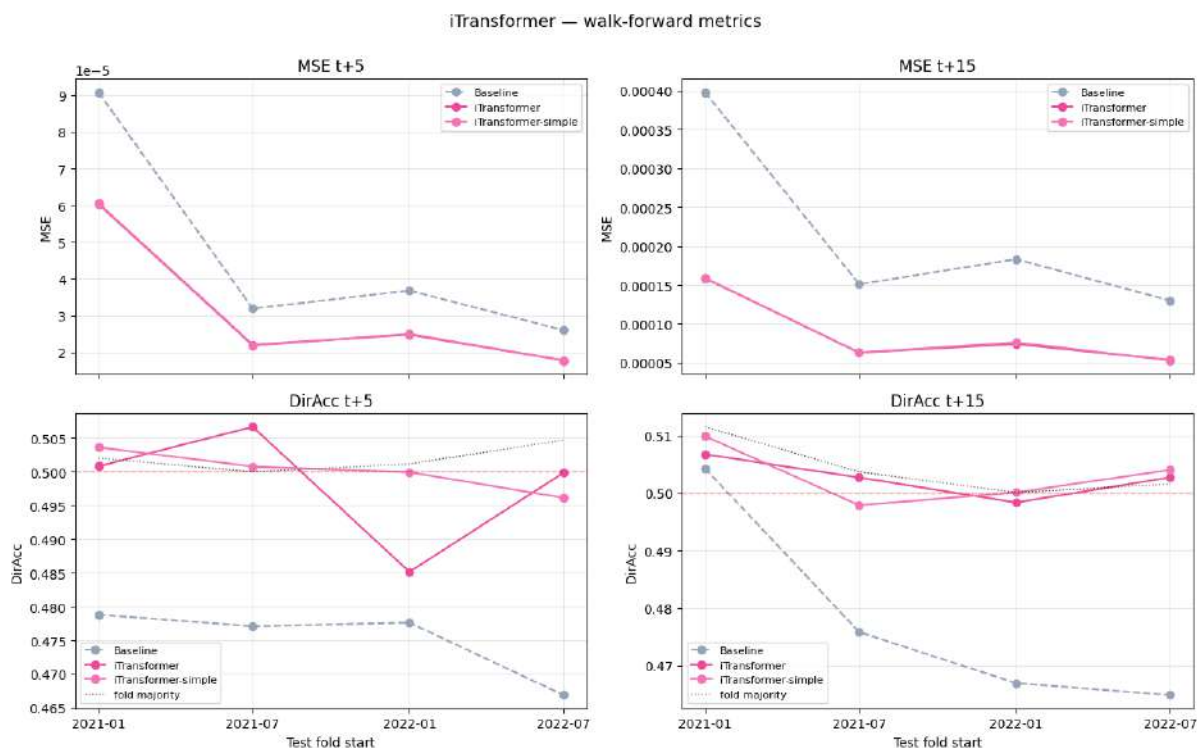


Рисунок 3.7 Walk-forward метрики для моделі iTransformer

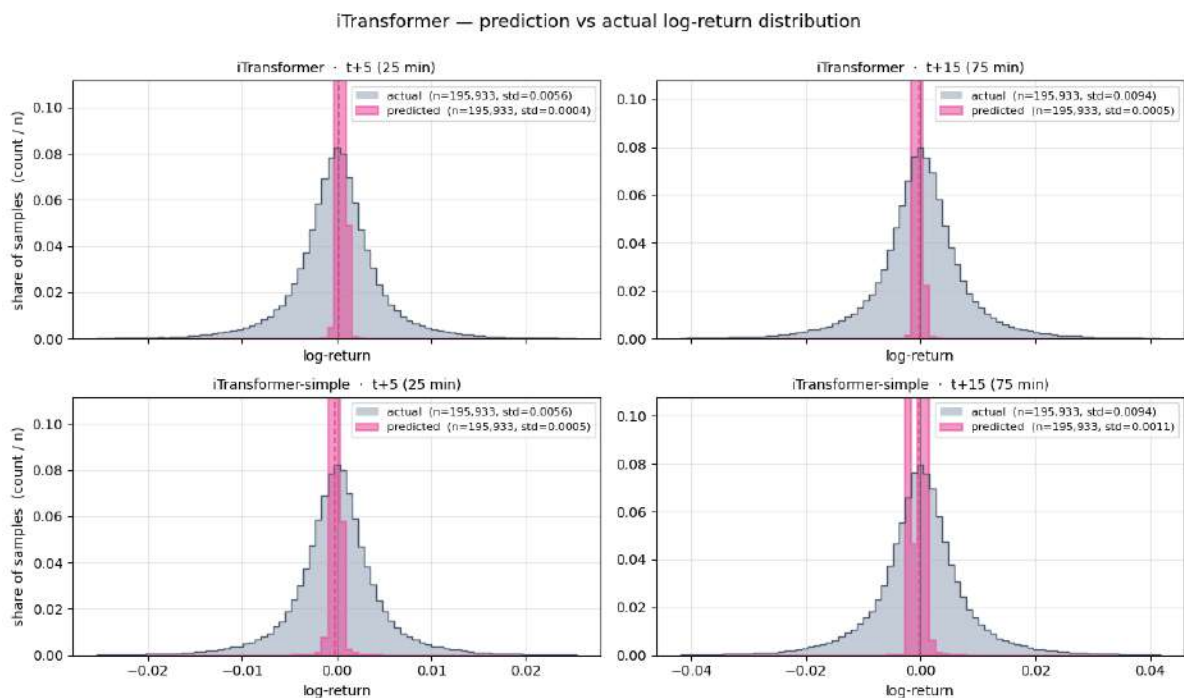


Рисунок 3.8 Розподіл тестових передбачень моделі iTransformer

3.5 Результати моделі FEDFormer

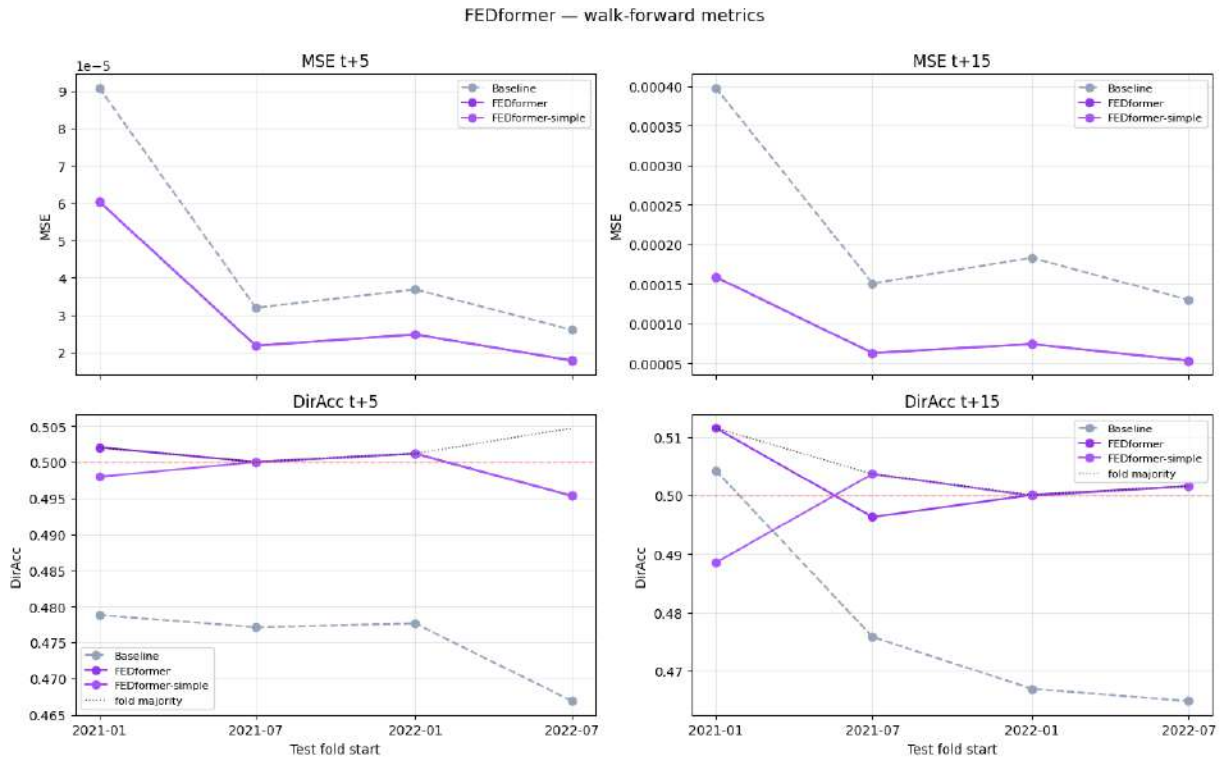


Рисунок 3.9 Walk-forward метрики для моделі FEDFormer

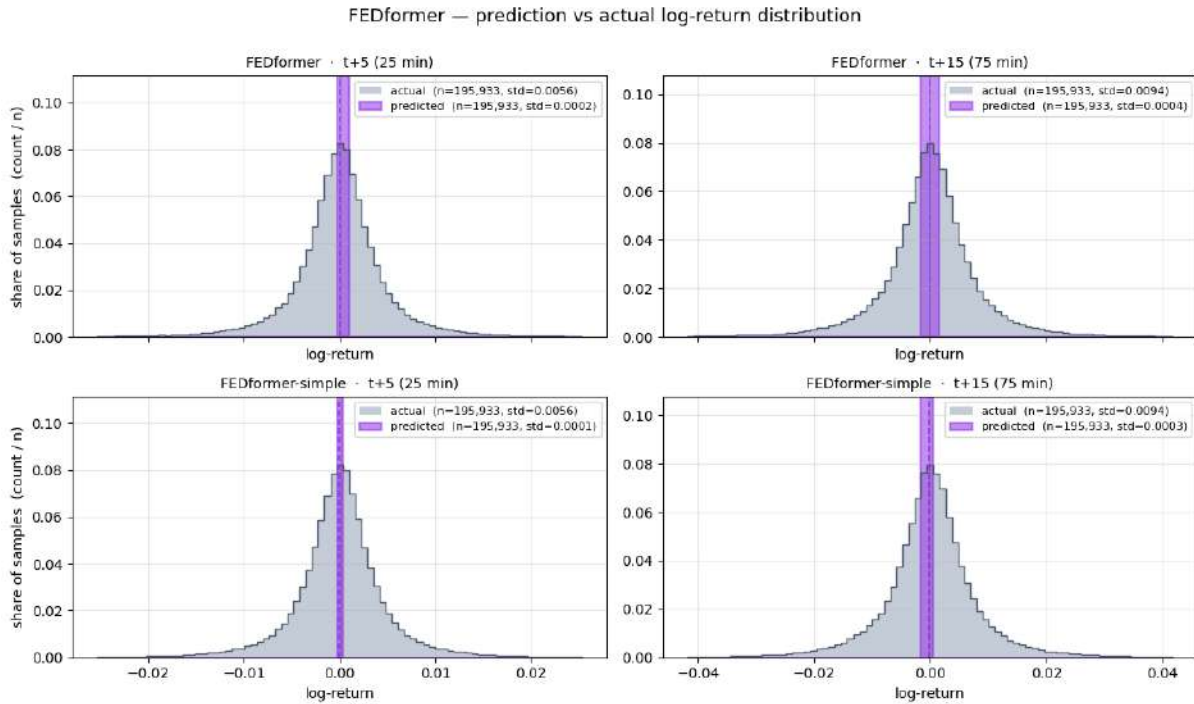


Рисунок 3.10 Розподіл тестових передбачень моделі FedFormer

3.6 Результати моделі FETransformer

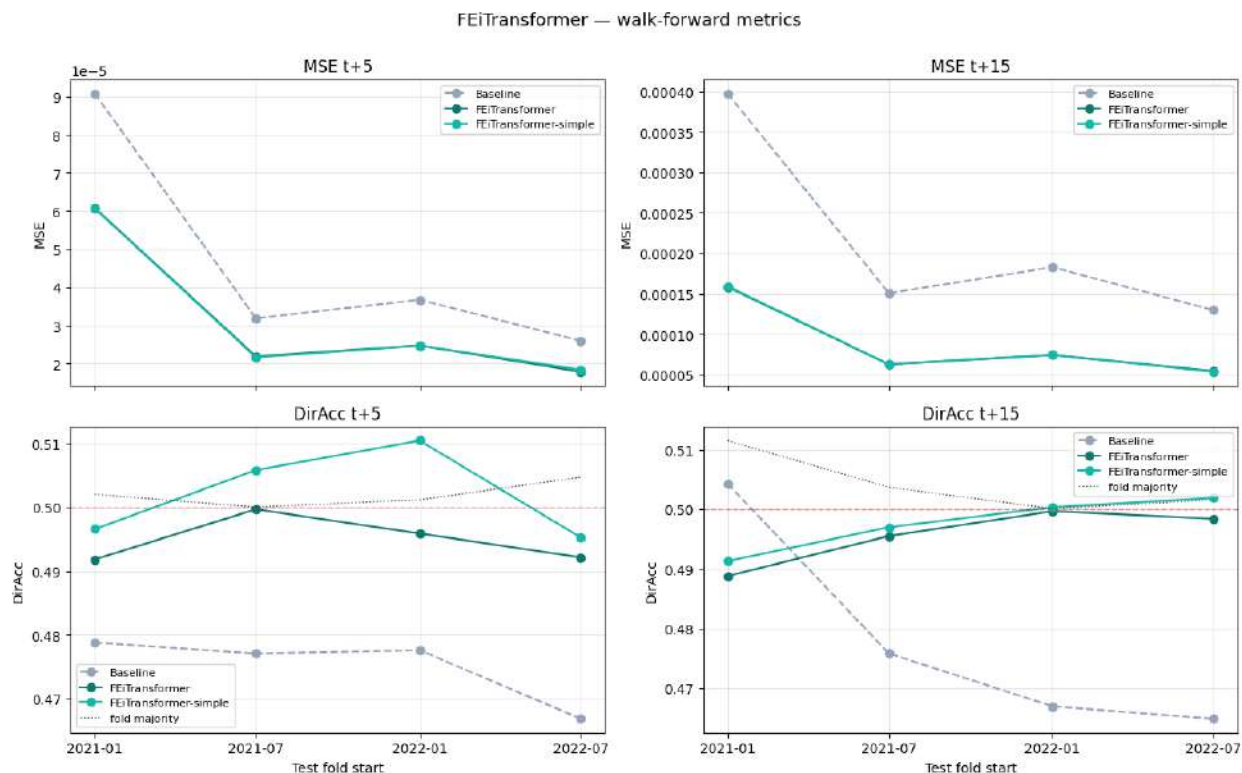


Рисунок 3.11 Walk-forward метрики для моделі FETransformer

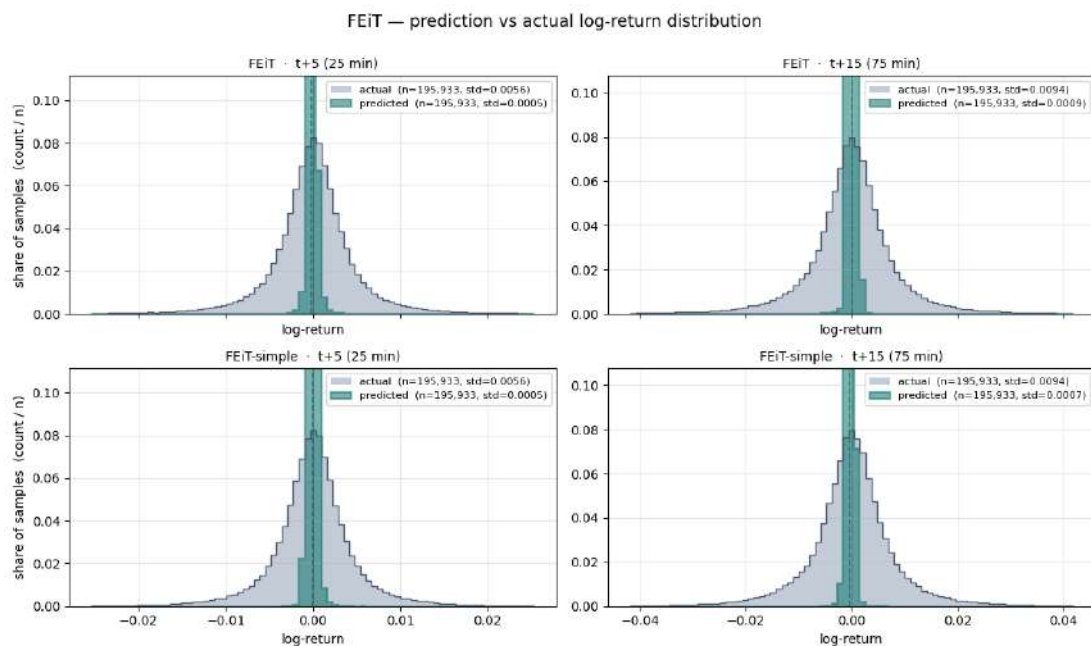


Рисунок 3.12 Розподіл тестових передбачень моделі Transformer

3.7 Остаточне порівняння моделей і ретроспективне тестування

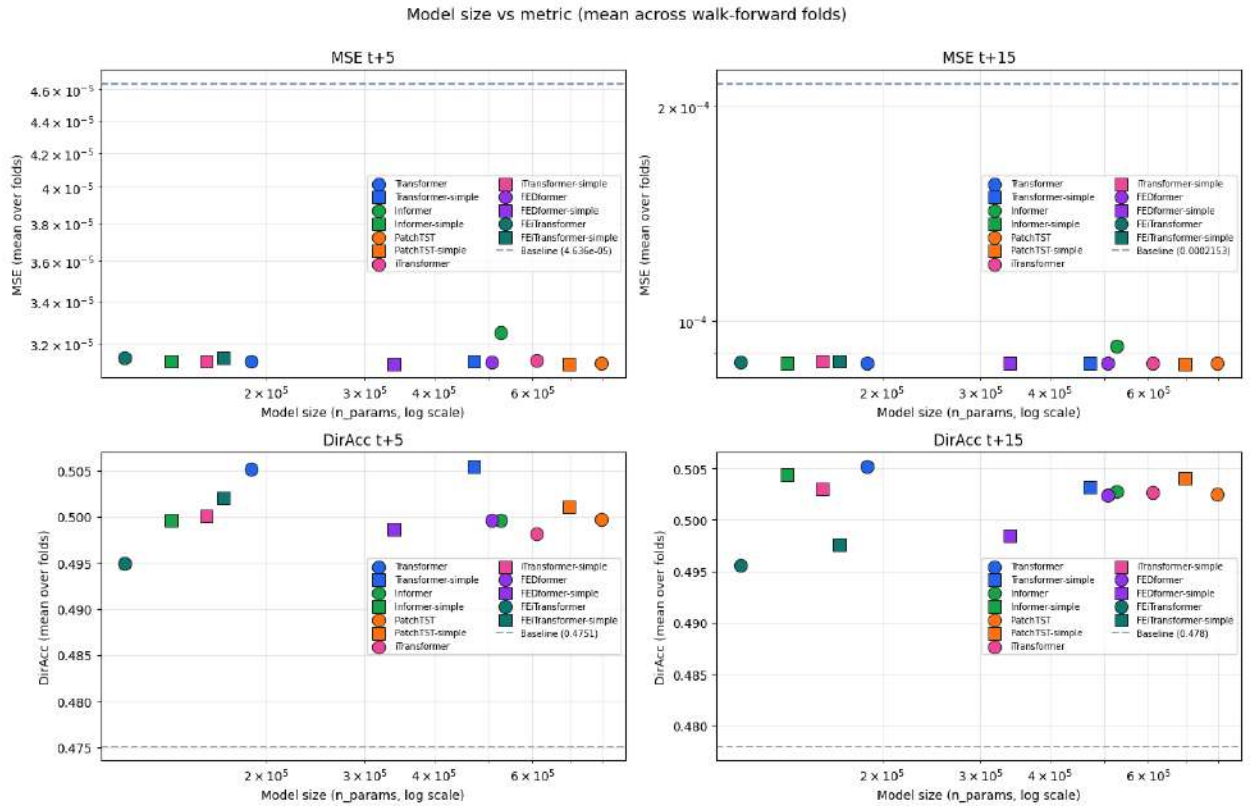


Рисунок 3.13 Візуалізація результатів моделей у відношенні до їх розміру

Зібрані метрики дозволяють порівняти ефективність моделей між собою, але не дають розуміння чи є практична доцільність їх використання для реальних фінансових рішень. Для того щоб оцінити, чи дають моделі прийнятні прогнози було побудовану просту трейдингову стратегію на їх основі.

Стратегія полягає у розміщенні торгової позиції у разі коли модель повертає велике за модулем передбачення. Передбачення представляють собою очікувану логарифмічну різницю між цінами зараз і на горизонті прогнозування, тому коли оцінка моделі відчутно більша/менша нуля, це означає що модель впевнена у сильному зрості або падінні ціни за наступні 5 або 15 часових кроків. Поріг для розміщення позиції обирається для кожної моделі окремо, так, щоб проходив рівно один відсоток найбільших за модулем передбачень.

Стратегії тестуються ретроспективно на всьому тестовому періоді. Для передбачення обирається модель натренована для відповідного тестового проміжка.

Кожна позиція має розмір 100 USDT і закривається по спливанню горизонту прогнозування. Тобто, для обчислення прибутку кожної відкритої позиції береться справжня історична зміна ціни за час позиції і апроксимується прибуток від такої торгової угоди. Далі наведено візуалізації кумулятивних прибутків для стратегії що базується на кожній натренованій моделі.

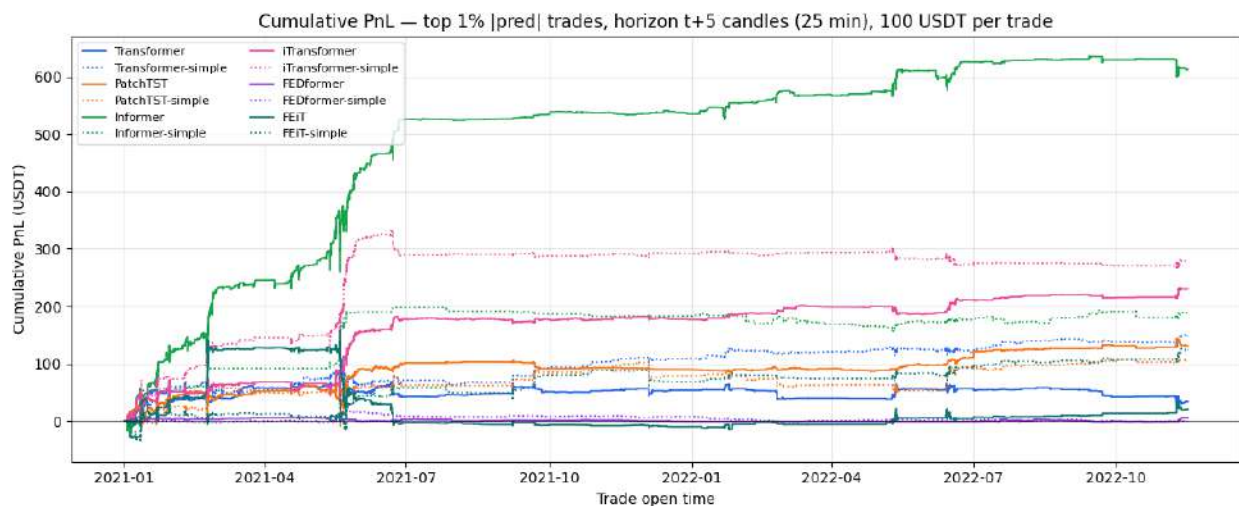


Рисунок 3.14 Кумулятивні прибутки торгових стратегій на горизонті $t+5$

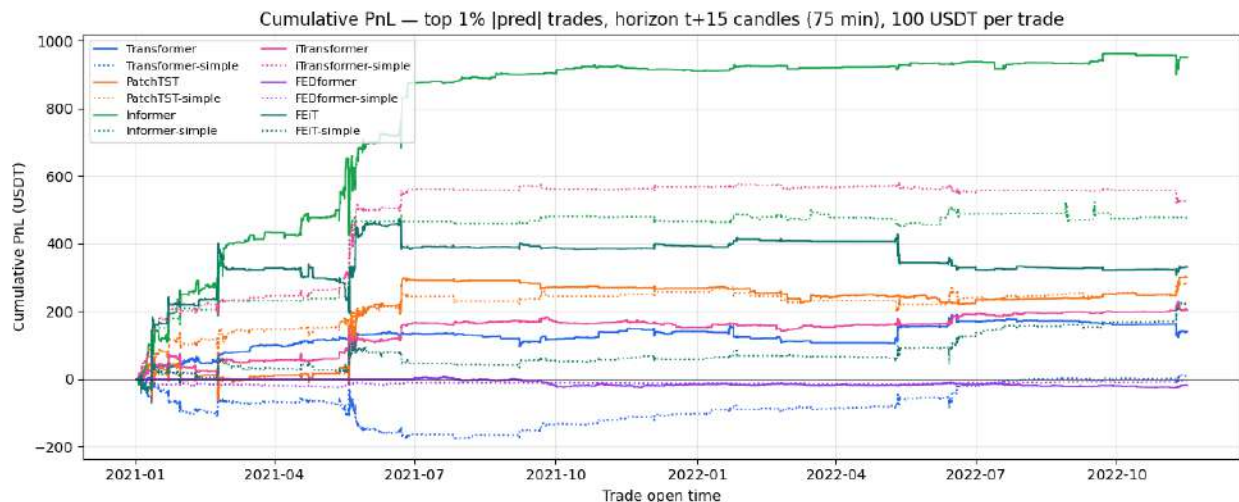


Рисунок 3.15 Кумулятивні прибутки торгових стратегій на горизонті $t+15$

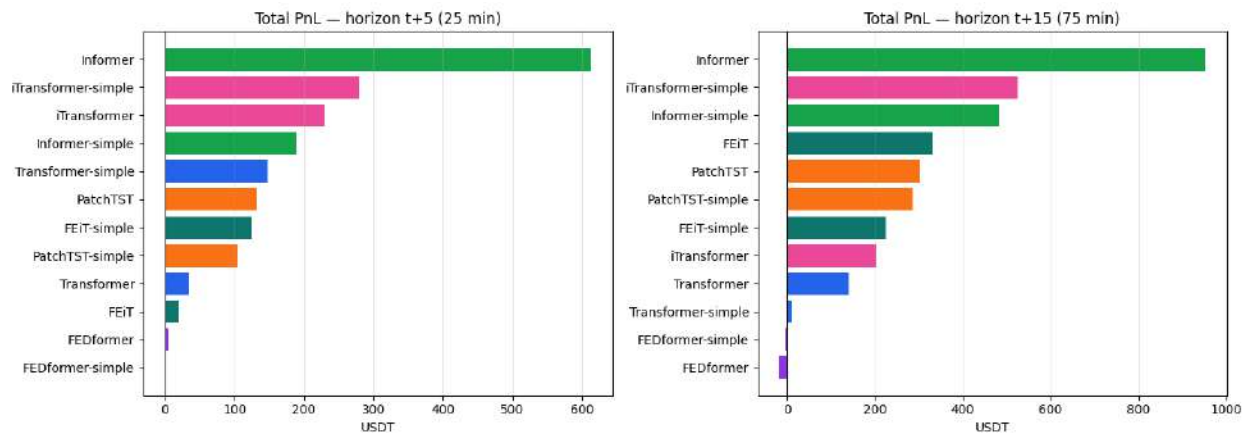


Рисунок 3.16 Прибуток отриманий за весь тестовий період

Висновки

У межах даної роботи було досліджено ефективність моделей на основі механізму уваги для задачі передбачення фінансових часових рядів. Було розглянуто класичну архітектуру Transformer та її спеціалізовані модифікації, адаптовані до роботи з часовими рядами: Informer, PatchTST, iTransformer та FEDFormer.

Окремо було запропоновано й реалізовано гібридну архітектуру FEiTransformer, що поєднує інверсію вхідного багатовимірного представлення з елементами частотної декомпозиції. Для забезпечення коректності порівняння всі моделі були навчені в однакових умовах та оцінені із застосуванням walk-forward підходу, який краще відповідає реальному сценарію використання прогнозних моделей на фінансових ринках.

Отримані результати показали, що досліджені моделі в більшості випадків перевершують просту статистичну модель як за середньоквадратичною похибкою, так і за точністю визначення напрямку майбутнього руху ціни. Це свідчить про те, що Transformer-архітектури здатні виявляти певні закономірності у високочастотних фінансових даних і можуть бути використані як основа для побудови прогнозних систем. Водночас результати не дають підстав стверджувати, що такі моделі здатні стабільно та безпосередньо розв'язувати задачу прибуткового прогнозування ринку. Фінансові часові ряди залишаються надзвичайно шумними, нестационарними та залежними від зовнішніх факторів, які не завжди присутні у вхідних історичних даних.

Важливим спостереженням стало те, що розподіли передбачень моделей мають суттєво менше стандартне відхилення порівняно з розподілом справжніх значень цільової змінної. Іншими словами, моделі схильні повертати занадто обережні прогнози, зосереджені близько до нуля. Така поведінка є очікуваною для моделей, оптимізованих за середньоквадратичною похибкою в умовах високого рівня шуму: найменш ризикованою стратегією для зменшення помилки стає прогнозування згладжених значень, а не рідкісних, але важливих сильних рухів ринку. Саме ця особливість обмежує практичну цінність моделей, оскільки для торгової стратегії важливо не лише правильно визначати середню динаміку, а й виявляти моменти, коли очікуваний рух ціни є достатньо великим для відкриття позиції.

З практичної точки зору найкращою виявилась модель Informer, яка найменше страждає від проблеми заниженої амплітуди прогнозів. Це показує, що для фінансового застосування важливо оцінювати моделі не лише за класичними метриками похибки, але й за характером самих передбачень: їхнім розподілом, варіативністю, здатністю відокремлювати сильні сигнали від слабких та придатністю до побудови торгових правил. У цьому контексті найціннішою є не обов'язково модель з мінімальною середньою помилкою, а модель, яка формує більш виразні та інформативні прогнози, що можуть бути використані для прийняття рішень.

Таким чином, мету роботи можна вважати досягнутою частково. З одного боку, було реалізовано та експериментально порівняно низку сучасних архітектур на основі механізму уваги, досліджено їхню поведінку на багатовимірних фінансових даних, а також запропоновано власну гібридну модель. З іншого боку, результати показали, що сама по собі архітектурна складність не гарантує практично корисного прогнозу. Основним обмеженням залишається не лише здатність моделі мінімізувати похибку, а й здатність формувати достатньо сильні за модулем передбачення у моменти, коли ринок справді демонструє значний рух.

Подальший розвиток роботи доцільно спрямувати у кількох напрямках. По-перше, перспективною є оркестрація кількох моделей у межах єдиної прогнозної системи. Різні архітектури можуть краще працювати в різних ринкових режимах: одні - під час трендових рухів, інші - в умовах високої волатильності або бокового ринку. Тому ансамблювання, динамічний вибір моделі або побудова метамоделі, що агрегує прогнози кількох архітектур, може підвищити надійність стратегії та зменшити залежність від слабких сторін окремої моделі.

По-друге, важливим напрямом є подальша робота з функцією втрат. Використання стандартної середньоквадратичної похибки мотивує модель зменшувати середню помилку, але не стимулює її робити прогнози більшої амплітуди у випадках потенційно сильного руху ціни. Для покращення практичної придатності моделей варто дослідити модифіковані функції втрат, які додатково враховують напрям руху, величину прогнозу, помилки на великих за модулем значеннях або прибутковість умовної торгової дії. Такий підхід може змістити оптимізацію від суто статистичної точності до більш прикладної мети - формування прогнозів, корисних для прийняття фінансових рішень.

Отже, результати дослідження підтверджують доцільність застосування моделей на основі механізму уваги для аналізу фінансових часових рядів, але також демонструють необхідність їх подальшої адаптації до специфіки фінансових задач. Найбільш перспективним є поєднання архітектурних модифікацій, багатовимірного представлення даних, спеціалізованих функцій втрат та ансамблевих підходів. Саме такий напрям може забезпечити перехід від моделей, що демонструють прийнятні статистичні метрики, до моделей, здатних бути корисними в реальних системах прогнозування та алгоритмічної торгівлі.

Під час виконання роботи було використано засоби генеративного ШІ. Для пошуку пов'язаних наукових досліджень та фінансових датасетів було використано продукт **OpenAI DeepResearch** (GPT-5.5 з доступом до веб-пошуку). Попередній аналіз і фільтрація джерел були виконані за допомогою **NotebookLM**. Агентні засоби **Codex** і **Claude Code** було використано для аналізу і адаптації програмного коду із репозиторіїв [6], [7], [8] та [9], а також для пошуку і виправлення помилок у власному програмному коді.

Список літератури

1. Zhou, Y. Why Do Transformers Fail to Forecast Time Series In-Context? [Електронний ресурс] / Y. Zhou, Y. Wang, S. Goel, A. R. Zhang // arXiv. – 2025. – arXiv:2510.09776. – Режим доступу: <https://arxiv.org/abs/2510.09776>, вільний. – Назва з екрана.
2. Nie, Y. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers [Електронний ресурс] / Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam // arXiv. – 2022. – arXiv:2211.14730. – Режим доступу: <https://arxiv.org/abs/2211.14730>, вільний. – Назва з екрана.
3. Zhou, H. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting / H. Zhou, S. Zhang, J. Peng [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. – 2021. – Vol. 35, № 12. – P. 11106–11115.
4. Zhou, T. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting / T. Zhou, Z. Ma, Q. Wen [et al.] // Proceedings of Machine Learning Research. – 2022. – Vol. 162. – P. 27268–27286.
5. Liu, Y. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting [Електронний ресурс] / Y. Liu, T. Hu, H. Zhang [et al.] // International Conference on Learning Representations. – 2024. – Режим доступу: <https://openreview.net/forum?id=JePfAI8fah>, вільний. – Назва з екрана.
6. Informer2020 [Електронний ресурс] : репозиторій вихідного коду. – Режим доступу: <https://github.com/zhouhaoyi/Informer2020>, вільний. – Назва з екрана.
7. PatchTST [Електронний ресурс] : репозиторій вихідного коду. – Режим доступу: <https://github.com/yuqinie98/PatchTST>, вільний. – Назва з екрана.
8. ICML2022-FEDformer [Електронний ресурс] : репозиторій вихідного коду. – Режим доступу: <https://github.com/MAZiqing/FEDformer>, вільний. – Назва з екрана.
9. iTransformer [Електронний ресурс] : репозиторій вихідного коду. – Режим доступу: <https://github.com/thuml/iTransformer>, вільний. – Назва з екрана.