КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,

освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Інформаційна система прогнозування та аналізу продажів у роздрібній торгівлі»

Студента 2-го курсу групи ІАВ-21

Ткачук Святослав Васильович

(прізвище, ім'я, по батькові)

Науковий керівник:

д.т.н., професор кафедри технологій управління Хлевна Ю.Л.

(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____	_____	_____	_____
(підпис студента)	(дата)	(підпис)	(дата)

Попередній захист:

_____

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри технологій управління _____ _____ _____

(підпис) (прізвище, ініціали) (дата)

Київ – 2026

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 - Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ

Завідувач кафедри

«___» _____ 2025 року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент Ткачук Святослав Васильович Група ІАВ-21

1. Тема кваліфікаційної роботи

Інформаційна система прогнозування та аналізу продажів у роздрібній торгівлі

Затверджена наказом по від «___» _____ 2026 р. № ___.

2. Строк подання студентом готової роботи – «___» _____ 2026 р.

3. Цільова установка та вихідні дані до роботи

Розробка та дослідження адаптивної системи прогнозування продажів для роздрібних торговельних мереж на основі архітектури Temporal Fusion Transformer (TFT) з інтеграцією механізмів безперервного навчання (Continual Learning) для подолання проблем концептуального дрейфу та холодного старту. Вхідні дані: публічний набір даних M5 Forecasting – Accuracy (понад 30 000 товарних позицій, 10 торгових точок, 1941 день спостережень, близько 60

мільйонів транзакцій), що містить ієрархічні дані про продажі, календарні події, цінову інформацію та регіональні характеристики.

4. Зміст роботи

Робота містить вступ, огляд теоретичних засад прогнозування часових рядів у роздрібній торгівлі з аналізом сучасних методів машинного навчання та архітектур глибоких нейронних мереж, а також постановку завдання. Описано особливості даних продажів, фактори впливу та проблеми нестаціонарності, концептуального дрейфу та холодного старту. Наведено математичне обґрунтування архітектури Temporal Fusion Transformer та методів безперервного навчання (Experience Replay, Elastic Weight Consolidation, Low-Rank Adaptation). Розроблено методику підготовки даних із синтетичною аугментацією для моделювання відсутності історії та змін розподілу. Проведено експериментальне дослідження на даних M5 Forecasting з порівнянням стратегій проксі-ініціалізації та методів адаптації. Представлено архітектуру хмарної системи з автоматизацією процесів навчання, моніторингу та адаптивного донавчання. У висновках подано основні результати дослідження, а також список використаних джерел та додатки.

5. Перелік графічного матеріалу (слайдів)

Відносна деградація точності прогнозу для режимів холодного старту, обмеженої історії та дрейфу; порівняльна гістограма якості прогнозування для нових товарів залежно від стратегії ініціалізації ембедингів; динаміка зростання похибки прогнозу залежно від кроку прогнозного горизонту; порівняльний аналіз режимів холодного старту, обмеженої історії та стандартного за ключовими метриками точності; розподіл метрики середньої абсолютної похибки за сценаріями; порівняння методів безперервного навчання за ключовими метриками точності та стійкості до забування; візуалізація прогнозів попиту після застосування різних стратегій донавчання; порівняння обчислювальної

ефективності методів донавчання; загальна архітектура хмарної системи прогнозування продажів; конвеєри підготовки даних, базового навчання моделі, щоденного прогнозування, моніторингу та виявлення дрейфу.

6. Календарний план виконання роботи:

№ п/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	3	01.10.25	01.10.25
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	27.12.25	27.12.25
3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	08.01.26	07.01.26
4.	Складання розгорнутого плану кваліфікаційної роботи	5	17.01.26	17.01.26
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	18.01.26 - 19.01.26	19.01.26
6.	Підготовка розділу 1	10	10.02.26	10.02.26
7.	Підготовка розділу 2	14	06.03.26	06.03.26
8.	Підготовка розділу 3	14	05.04.26	05.04.26
9.	Підготовка розділу 4	13	21.04.26	21.04.26
10.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	15	29.04.26	29.04.26
11.	Передача кваліфікаційної роботи науковому керівникові	2	06.05.26	06.05.26
12.	Передача кваліфікаційної роботи рецензенту для рецензування	2	07.05.26	07.05.26
13.	Попередній захист кваліфікаційної роботи	5	11.05.26	11.05.26

Дата видачі завдання «___» _____ 20__ р.

Керівник роботи Доктор наук, доцент кафедри технологій управління Доктор технічних наук, професор кафедри технологій управління Хлевна Ю.Л.

(підпис)

Завдання прийняв до виконання студент групи ІАВ-21 Ткачук Святослав Васильович

(підпис)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 - Комп'ютерні науки,

освітня програма "Інформаційна аналітика та впливи"

Кваліфікаційна робота магістра Ткачука Святослава Васильовича.

Тема роботи – «Інформаційна система прогнозування та аналізу продажів у роздрібній торгівлі».

Мета кваліфікаційної роботи магістра – розробка та дослідження адаптивної системи прогнозування продажів для роздрібних торговельних мереж на основі архітектури Temporal Fusion Transformer з інтеграцією механізмів безперервного навчання для подолання проблем концептуального дрейфу та холодного старту, спрямована на підвищення точності прогнозів та покращення адаптивності моделі до змін розподілу даних в умовах нестаціонарного середовища.

Об'єкт дослідження – процес прогнозування обсягів продажів у роздрібних торговельних системах в умовах постійного потокового надходження даних та змін ринкових умов.

Предмет дослідження – методи машинного навчання, архітектурні рішення та алгоритмічні підходи до побудови адаптивних систем прогнозування на основі Temporal Fusion Transformer з механізмами безперервного навчання для подолання проблем нестаціонарності, концептуального дрейфу та холодного старту.

Наукова новизна роботи.

1. Запропоновано комплексний підхід до прогнозування продажів, що інтегрує архітектуру Temporal Fusion Transformer з методами безперервного навчання для одночасного подолання концептуального дрейфу та проблеми холодного старту.
2. Удосконалено застосування синтетичної аугментації на етапі навчання, яке імітує відсутність історичних даних та зміни розподілу попиту і цін, що підвищує узагальнювальну здатність моделі в нестаціонарних умовах.
3. Запропоновано механізм проксі-ініціалізації векторних представлень для нових товарів на основі їхньої категоріальної та відділової приналежності, який уможливлює використання навчених патернів семантично близьких об'єктів без повного перерахунку параметрів мережі.
4. Дістало подальший розвиток підхід параметрично ефективного донавчання на базі низькорангової адаптації з одночасним оновленням шарів вкладень, що забезпечує мінімальне споживання пам'яті та прискорену інкрементальну адаптацію моделі.

Ключові слова: прогнозування продажів, Temporal Fusion Transformer, безперервне навчання, Continual Learning, Low-Rank Adaptation, Experience Replay, Elastic Weight Consolidation, квантильна регресія, хмарні системи.

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

ARIMA – Autoregressive Integrated Moving Average – авторегресійна інтегрована модель ковзного середнього

EWC – Elastic Weight Consolidation – еластичне консолідування ваг

ETS – Error, Trend, Seasonality – модель з компонентами похибки, тренду та сезонності

GPU – Graphics Processing Unit – графічний процесор

LoRA – Low-Rank Adaptation – низькорангова адаптація

LSTM – Long Short-Term Memory – довга короткочасна пам'ять

MAE – Mean Absolute Error – середня абсолютна похибка

MMD – Maximum Mean Discrepancy – метод максимальної різниці середніх

RNN – Recurrent Neural Network – рекурентна нейронна мережа

RMSE – Root Mean Square Error – середньоквадратична похибка

sMAPE – Symmetric Mean Absolute Percentage Error – симетрична середня абсолютна відсоткова похибка

SKU – Stock Keeping Unit – одиниця складського обліку

TFT – Temporal Fusion Transformer – часовий трансформер з тимчасовою фузією

WQL – Weighted Quantile Loss – зважена квантильна втрата

ВСТУП

Актуальність теми. У сучасних умовах глобалізації ринків та стрімкої цифровізації бізнес-процесів ефективне управління ланцюгами постачання стає критичним фактором конкурентоспроможності роздрібних торговельних мереж. Точне прогнозування попиту дозволяє оптимізувати товарні запаси, мінімізувати втрати від дефіциту або надлишку продукції, покращити фінансові результати та підвищити рівень обслуговування клієнтів. За даними досліджень, компанії, що впроваджують сучасні системи прогнозування, зменшують рівень запасів на 20–30% та підвищують точність прогнозів на 15–25%.[1]

Проте традиційні статистичні методи прогнозування, такі як ARIMA, експоненціальне згладжування та регресійні моделі, виявляють суттєві обмеження в умовах сучасного роздрібного середовища. Вони припускають стаціонарність даних, не враховують складні нелінійні залежності та не здатні адаптуватися до швидких змін ринкових умов.[2] Реальні дані продажів характеризуються високою волатильністю, наявністю сезонних патернів, впливом промо-акцій та зовнішніх факторів, що призводить до явища концептуального дрейфу – зміни статистичних властивостей даних у часі.[3]

Особливу складність становить прогнозування для нових товарів, які не мають історії продажів (проблема «cold start»). У роздрібних мережах щорічно оновлюється 5–15% асортименту, і відсутність ефективних механізмів ініціалізації прогнозів для таких товарів призводить до суттєвих фінансових втрат [4]. Традиційні підходи вимагають накопичення історичних даних протягом кількох тижнів або місяців перед початком використання прогнозної моделі, що є неприйнятним в умовах динамічного ринку.

Розвиток методів глибокого навчання відкрив нові можливості для прогнозування часових рядів. Архітектури на основі рекурентних нейронних мереж (RNN), довгої короткочасної пам'яті (LSTM) та механізмів уваги (Attention) демонструють високу точність у задачах багатокрокового прогнозування[5, 6]. Особливий інтерес становить модель Temporal Fusion Transformer (TFT), яка поєднує переваги LSTM для кодування локальних залежностей та трансформерів для моделювання довгострокових взаємодій, забезпечуючи при цьому інтерпретованість результатів[7].

Проте статичне навчання нейронних мереж не забезпечує стійкості до змін розподілу даних. Підходи безперервного навчання (Continual Learning), такі як Experience Replay, Elastic Weight Consolidation (EWC) та Low-Rank Adaptation (LoRA), дозволяють інкрементально адаптувати моделі до нових умов без катастрофічного забування[8, 9] раніше засвоєних знань. Інтеграція цих методів у системи прогнозування продажів є перспективним напрямком досліджень.

Мета і завдання дослідження. Метою роботи є розробка адаптивної системи прогнозування продажів на основі архітектури Temporal Fusion Transformer з механізмами безперервного навчання для подолання проблем концептуального дрейфу та холодного старту в умовах роздрібної торгівлі.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. Провести аналіз існуючих методів і підходів до прогнозування продажів у роздрібній торгівлі, дослідити їхні переваги та обмеження в умовах нестаціонарності даних.

2. Дослідити особливості даних продажів, фактори, що впливають на обсяги продажів, та проблеми нестаціонарності, концептуального дрейфу та холодного старту.

3. Обґрунтувати вибір методів машинного навчання для вирішення поставленої задачі, провести порівняльний аналіз архітектур глибоких нейронних мереж для прогнозування часових рядів.

4. Розробити підхід підготовки даних та побудови ознак для навчання моделі, включаючи механізми синтетичної аугментації для моделювання холодного старту та концептуального дрейфу.

5. Реалізувати архітектуру Temporal Fusion Transformer з інтеграцією механізмів безперервного навчання для адаптації моделі до змін розподілу даних.

6. Провести експериментальне дослідження розробленої системи на публічному наборі даних M5 Forecasting, оцінити ефективність запропонованих підходів до подолання холодного старту та концептуального дрейфу.

7. Розробити архітектуру хмарної системи прогнозування продажів з автоматизацією процесів навчання, моніторингу та адаптивного донавчання моделей.

Об'єкт дослідження – процес прогнозування обсягів продажів у роздрібних торговельних системах в умовах постійного потокового надходження даних та змін ринкових умов.

Предмет дослідження – методи машинного навчання, архітектурні рішення та алгоритмічні підходи до побудови адаптивних систем прогнозування на основі

Temporal Fusion Transformer з механізмами безперервного навчання для подолання проблем нестаціонарності, концептуального дрейфу та холодного старту.

Методи дослідження. У роботі використано методи теорії ймовірностей та математичної статистики для аналізу часових рядів, методи глибокого навчання (рекурентні нейронні мережі, механізми уваги, трансформери), методи безперервного навчання (Experience Replay, Elastic Weight Consolidation, Low-Rank Adaptation), методи оптимізації нейронних мереж, а також експериментальні методи оцінювання якості прогнозних моделей.

Наукова новизна одержаних результатів:

1. У роботі запропоновано комплексний підхід до прогнозування продажів, що інтегрує архітектуру Temporal Fusion Transformer з методами безперервного навчання для одночасного подолання концептуального дрейфу та проблеми холодного старту;
2. реалізовано застосування синтетичної аугментації на етапі навчання, яке імітує відсутність історичних даних та зміни розподілу попиту і цін, що підвищує узагальнювальну здатність моделі в нестаціонарних умовах;
3. розроблено механізм проксі-ініціалізації векторних представлень для нових товарів на основі їхньої категоріальної та відділової приналежності, який уможливлює використання навчених патернів семантично близьких об'єктів без повного перерахунку параметрів мережі;
4. реалізовано та проведено порівняльний аналіз чотирьох стратегій інкрементального донавчання (базове донавчання, метод повторного відтворення досвіду, еластичне консолідування ваг та низькорангова адаптація з оновленням шарів вкладень), що дозволило визначити

оптимальні підходи для різних сценаріїв експлуатації з урахуванням компромісу між точністю, стійкістю до забування та обчислювальною ефективністю.

Практичне значення одержаних результатів:

1. Розроблено програмну реалізацію адаптивної системи прогнозування продажів на основі Temporal Fusion Transformer з інтеграцією механізмів безперервного навчання, яка демонструє високу точність прогнозів та стійкість до концептуального дрейфу.

2. Запропоновано архітектуру хмарної системи прогнозування з автоматизацією процесів збору даних, навчання моделей, моніторингу якості та адаптивного донавчання, що забезпечує масштабованість та економічну ефективність.

3. Результати дослідження можуть бути використані роздрібними торговельними мережами для впровадження систем прогнозування попиту, оптимізації товарних запасів та підвищення ефективності управління ланцюгами постачання.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1. ФОРМУВАННЯ ТА АНАЛІЗ ЗАДАЧІ ПРОГНОЗУВАННЯ ПРОДАЖІВ В УМОВАХ ПОСТІЙНОГО НАДХОДЖЕННЯ ДАНИХ

1.1. Характеристика об’єкта та предмета дослідження

У сучасних умовах високої динамічності споживчого ринку, жорсткої конкуренції та цифровізації логістичних процесів точне прогнозування попиту є критичним елементом управління ланцюгами постачання, оптимізації товарних залишків та мінімізації операційних витрат роздрібних торговельних мереж. Стрімке зростання обсягів транзакційних даних, часте оновлення асортименту, зміни споживчих уподобань та вплив макроекономічних факторів формують середовище, у якому традиційні статичні прогнозові моделі швидко втрачають свою ефективність.

Об’єктом дослідження є процес прогнозування обсягів продажів у роздрібних торговельних системах, що реалізується шляхом аналізу багатовимірних часових рядів попиту в умовах постійного потокового надходження даних. Цей процес характеризується високою мінливістю, наявністю складних сезонних та календарних компонентів, а також сильною залежністю від екзогенних змінних, серед яких цінова політика, промоактивності, регіональні особливості попиту та статус наявності товару в асортименті.

Предметом дослідження виступають методи машинного навчання, архітектурні рішення та алгоритмічні підходи до побудови адаптивних систем прогнозування, здатних функціонувати в умовах нестаціонарності розподілів даних, дефіциту історичної інформації для нових товарів та потреби в інкрементальному оновленні моделей без втрати раніше засвоєних знань. Дослідження зосереджене на поєднанні глибоких нейронних архітектур для багатокрокового прогнозування з механізмами безперервного навчання,

синтетичної аугментації тренувальних даних та проксі-ініціалізації векторних представлень для нових товарних позицій.

Галузь роздрібної торгівлі характеризується складною ієрархічною структурою даних, де кожен часовий ряд попиту є вкладеним у багаторівневу таксономію: від загальної категорії продукції та відділу до конкретної товарної позиції, прив’язаної до певної торгової точки та регіону. Така структура обумовлює наявність перехресних залежностей між рівнями агрегації та вимагає одночасного моделювання тисяч паралельних часових рядів, кожен з яких має власну динаміку, рівень шуму та специфічні поведінкові патерни. Багатовимірність задачі проявляється у необхідності одночасної обробки числових динамічних ознак (ціни, календарні індикатори, статуси наявності) та категоріальних статичних ознак, що описують атрибути товарів і точок продажу.

До сучасних систем прогнозування висуваються суворі операційні та технічні вимоги. По-перше, стандартний горизонт прогнозування в роздрібній торгівлі зазвичай становить двадцять вісім днів, що відповідає тижневому циклу планування замовлень, логістики та управління залишками. По-друге, моделі повинні оновлюватися з частотою, що дозволяє своєчасно реагувати на зміни попиту, забезпечуючи актуальність прогнозів без тривалих простоїв інфраструктури. По-третє, архітектура системи має забезпечувати масштабованість для підтримки десятків тисяч товарних позицій, сотень торгових точок та мільйонів спостережень, зберігаючи прийнятну обчислювальну складність, низьку затримку інференсу та ефективне використання обчислювальної пам’яті.

Функціонування сучасної інформаційної системи такого типу не обмежується лише автоматизованою генерацією прогнозних траєкторій, а

передбачає інтеграцію прогнозної та аналітичної складових у єдиний контур підтримки прийняття рішень. На кожному етапі життєвого циклу моделі здійснюється діагностичний аналіз точності, інтерпретація впливу екзогенних факторів через механізми уваги та динамічного відбору змінних, а також ретроспективне порівняння фактичних показників із передбачуваними значеннями з урахуванням інтервалів невизначеності. Це дозволяє не лише контролювати якість передбачень, а й виявляти приховані закономірності попиту, моделювати сценарні умови для оцінки впливу цінових змін та промоактивностей, а також формувати аналітичні звіти для оперативного та стратегічного управління ланцюгами постачання. Таким чином, прогнозування виступає джерелом структурованих даних для глибинної аналітики, а аналітична валідація та візуалізація результатів забезпечують зворотний зв'язок для калібрування моделей, створюючи замкнену екосистему, де алгоритмічні обчислення та експертна інтерпретація функціонують синергетично.

Традиційні статистичні підходи до прогнозування часових рядів, такі як ARIMA, ETS або Prophet, а також сучасні ансамблі на основі дерев рішень (наприклад, LightGBM чи XGBoost), виявляють суттєві обмеження в зазначених умовах. Насамперед, ці методи мають статичну природу: після завершення етапу навчання їхні параметри фіксуються, що призводить до швидкої деградації прогнозної здатності за зміни ринкових умов або появи нових товарів. Крім того, вони надзвичайно чутливі до явища зсуву розподілу даних у часі (concept drift), яке в роздрібній торгівлі часто спричинене змінами цін, промокампаніями, сезонними флуктуаціями або зовнішніми економічними шоками. Відсутність вбудованих механізмів ініціалізації для товарів без історії продажів (cold start) унеможливлює коректне прогнозування в умовах постійного оновлення асортименту. До того ж, зазначені підходи не підтримують інкрементальне

навчання, що вимагає повного перерахунку моделі на всій історичній вибірці під час кожного оновлення, суттєво збільшуючи витрати на обчислювальні ресурси.

Таким чином, аналіз об’єкта дослідження засвідчує необхідність застосування новітніх підходів до машинного навчання, здатних поєднувати високоточне багатокрокове прогнозування з механізмами безперервного навчання, динамічної адаптації до змін у розподілах даних та ініціалізації представлень для нових сутностей. Це зумовлює доцільність розробки гібридної архітектури на базі моделі часового трансформера з тимчасовою фузією (Temporal Fusion Transformer), інтегрованої у хмарне середовище з автоматизованими конвеєрами моніторингу, виявлення зсувів розподілу даних та адаптивного донавчання, що становить теоретичну основу подальшого викладу матеріалу в даній роботі.

1.2. Особливості даних продажів та фактори, що впливають на обсяги продажів

Ефективність моделей прогнозування попиту значною мірою визначається якістю, репрезентативністю та семантичною повнотою вхідних даних. У роздрібній торгівлі часові ряди продажів характеризуються високою волатильністю, мультиколінеарністю екзогенних змінних та складною структурою перехресних залежностей між рівнями товарної ієрархії. Для дослідження було використано публічний набір даних M5 Forecasting – Accuracy, який містить ієрархічні щоденні дані про продажі мережі Walmart[10] у трьох штатах США за період з 2011 по 2016 рік. Цей датасет є загальновизнаним стандартом для бенчмаркінгу сучасних методів прогнозування завдяки своїй об’ємності, наявності детальних календарних та цінових ознак, а також чітко визначеній структурній організації, що включає рівні товару, відділу, категорії, магазину та штату.

Первинна інформація формується на основі трьох інтегрованих джерел, кожне з яких виконує специфічну функцію у побудові навчальної вибірки. Основний масив містить щоденні обсяги продажів у розрізі тисяч товарних позицій та торгових точок, представлений у форматі із горизонтальним розташуванням часових кроків. Календарний довідник доповнює його датами, назвами днів тижня, номерами тижнів, а також інформацією про національні та релігійні свята, спортивні заходи та статус програми державної допомоги населенню. Третє джерело містить щотижневі дані про ціноутворення, що дозволяє відстежувати динаміку вартості товарів, сезонні зміни та вплив промо-активності.

Попит у роздрібній торгівлі має виражені циклічні компоненти, які моделюються через спеціально сконструйовані календарні змінні, що формують темпоральний контекст для алгоритму навчання. До них належать індикатори подій, які диференціюють вплив різних видів свят на споживчу поведінку, оскільки спортивні заходи стимулюють продажі одних груп товарів, тоді як національні свята зсувають попит у бік інших категорій. Окремо враховуються вихідні дні та початок чи кінець місяця, що відображає типові патерни закупівель та зміну структури споживчого кошика. Для уникнення розриву неперервності при переході між циклічними періодами застосовано синусоїдальне та косинусоїдальне перетворення місяців, днів тижня та днів року. Математично це виражається формулами

$$x_{\sin} = \sin\left(\frac{2\pi x}{T}\right), \tag{1.1}$$

$$x_{\cos} = \cos\left(\frac{2\pi x}{T}\right), \tag{1.2}$$

де x — вхідна ознака; T — період циклу. Такий підхід зберігає топологію часового простору та покращує збіжність градієнтних методів оптимізації. Додатково введено регіональний індикатор використання продуктових талонів, який відображає дні зі штучними сплесками попиту на товари першої необхідності.

Цінова політика та статус асортименту виступають одними з найсильніших драйверів попиту, проте їхній вплив є нелінійним, гетерогенним за категоріями та залежить від часового контексту. Оскільки ціни оновлюються щотижня, їхнє злиття з щоденними продажами вимагає точної синхронізації за календарним ключем та подальшого коректного інтерполювання на денний рівень. Додатково введено бінарний індикатор наявності товару в асортименті, який дозволяє моделі розрізняти нульові продажі через фізичну відсутність продукції на полиці та через реальну відсутність попиту. Промо-активність та зміни цін моделюються комплексно через взаємодію різких змін вартості та календарних подій, що у сукупності формує контекст для виявлення аномальних сплесків попиту та коректного оцінювання цінової еластичності.

Вихідний горизонтальний формат таблиць є неефективним для навчання багатовимірних нейронних архітектур, тому здійснено перехід до нормалізованого вертикального формату, де кожен запис відповідає одній комбінації товару, точки продажу та дати. Після інтеграції з календарними та ціновими довідниками виконано категоріальне кодування всіх ієрархічних ідентифікаторів. Для оптимізації використання пам’яті під час навчання та

інференсу дані серіалізуються у форматі відображених на диск масивів, що дозволяє працювати з обсягами інформації, які перевищують оперативну пам’ять, шляхом потокового зчитування фрагментів без повного завантаження. Цільова змінна зберігається у первинному масштабі, а логарифмічне перетворення виду $(\log(1+x))$ застосовується динамічно під час формування навчальних батчів. Це стабілізує дисперсію, зменшує вплив викидів та покращує чутливість моделі до малих значень продажів без втрати інтерпретованості результатів.

Емпіричний аналіз та попередні дослідження засвідчують, що ігнорування календарних ознак призводить до систематичної похибки у святкові періоди, відсутність цінових змінних ускладнює моделювання еластичності попиту, а неврахування статусу наявності товару спотворює оцінку базового рівня продажів. Синусоїдальне кодування часу та структуроване злиття екзогенних факторів у єдиний тензор динамічних ознак забезпечує темпоральну узгодженість вхідних даних, що є критичною вимогою для архітектур типу часових трансформерів. Такі моделі потребують чіткого розділення вхідної інформації на статичні характеристики об’єктів, історичні спостереження та відомі майбутні значення. Реалізований підхід до підготовки даних забезпечує формування структурованого, семантично збагаченого та обчислювально ефективного представлення часових рядів, що створює необхідну основу для подальшого навчання адаптивних систем, здатних працювати в умовах обмеженої історичної інформації, змін розподілів даних та інкрементального оновлення знань.

1.3. Аналіз задач прогнозування часових рядів у торговельних системах

Прогнозування попиту в роздрібній торгівлі є класичною задачею аналізу часових рядів, яка в умовах сучасних торговельних мереж набуває значної

складності через високу розмірність даних, нелінійність залежностей та необхідність одночасного опрацювання тисяч паралельних рядів. Вибір архітектури та методології моделювання безпосередньо залежить від класифікації поставленої задачі, вимог до точності прогнозів та обчислювальних ресурсів інфраструктури. Традиційно задачі прогнозування часових рядів класифікують за масштабом моделювання та довжиною прогнозного горизонту. За масштабом розрізняють локальне та глобальне прогнозування. Локальні підходи передбачають навчання окремої моделі для кожного часового ряду незалежно від інших. Хоча такі методи добре адаптуються до специфіки окремої товарної позиції, вони не здатні виявляти спільні патерни попиту, вкрай чутливі до малих вибірок та вимагають квадратичного зростання обчислювальних витрат при масштабуванні. На противагу цьому, глобальні моделі навчаються одночасно на всіх доступних рядах[11], використовуючи спільні параметри для виявлення перехресних залежностей між рядами. Такі архітектури ефективніше працюють з товарами, що мають рідкісні продажі, узагальнюють сезонні та календарні ефекти та демонструють кращу узагальнювальну здатність за умови достатньої якості вхідних ознак.

За довжиною горизонту прогнозу задачі поділяють на однокрокові та багатокрокові. Однокрокове прогнозування передбачає передбачення лише наступного значення, що є недоцільним для оперативного планування логістики, де необхідні дані на тиждень чи місяць наперед. Багатокрокове прогнозування генерує вектор значень на заданий горизонт безпосередньо або через рекурентне застосування однокрокової моделі. У роздрібній торгівлі стандартним є горизонт $H = 28$ днів, що відповідає повному місячному циклу закупівель, інвентаризації та формування промо-планів, тому пряма багатокрокова стратегія є найбільш доцільною з точки зору мінімізації накопичення похибки.

Дані роздрібної торгівлі мають чітку ієрархічну структуру, де обсяги продажів агрегуються від рівня конкретної товарної позиції в конкретному магазині до рівня відділу, категорії, мережі та регіону. Це породжує задачу ієрархічного прогнозування, у якій суми прогнозів на нижчих рівнях мають математично узгоджуватися з прогнозами на вищих рівнях. Традиційні підходи вирішують цю задачу через процедури узгодження, зокрема методи «зверху-вниз» (прогнозування на верхньому рівні з подальшим розподілом за історичними частками), «знизу-вгору» (незалежне прогнозування всіх базових рядів із їх сумуванням) або оптимального узгодження, що мінімізує дисперсію помилок. Проте сучасні глибокі нейронні архітектури частково нівелюють потребу в явній процедурі узгодження завдяки спільному кодуванню статичних ознак, які містять ієрархічні ідентифікатори. Модель навчається внутрішньо узгоджувати прогнози через механізми уваги та спільні вагові коефіцієнти, що зменшує обчислювальне навантаження та усуває помилки апроксимації, притаманні послідовним методам узгодження.

Обробка ієрархічних даних тисяч товарних позицій протягом тисяч днів генерує матриці розміром у десятки мільйонів спостережень, що перевищує об’єм оперативної та відеопам’яті типових обчислювальних систем. Це вимагає застосування спеціалізованих інженерних рішень для забезпечення масштабованості алгоритмічного конвеєра підготовки даних. По-перше, реалізовано перехід від повного завантаження даних у оперативну пам’ять до використання відображених у пам’ять масивів, які дозволяють потоково зчитувати послідовності з накопичувача без створення дублікатів. По-друге, дані розбиваються на ковзаючі вікна з фіксованим кроком зсуву, що забезпечує збалансоване представлення трендів та сезонності у кожному навчальному пакеті без надмірного дублювання інформації. По-третє, пакетна обробка оптимізована

під архітектуру графічних процесорів, а розмір пакета підбирається з урахуванням збільшення семплів під час процедур розширення даних, що вимагає точного балансу між швидкістю збіжності моделі та доступними апаратними ресурсами.

Конфігурація параметрів генерації вибірки у дослідженні базується на бізнес-логіці роздрібного планування та архітектурних обмеженнях нейронної мережі. Довжина історичного вікна встановлено як дев’яносто днів, що охоплює повний сезонний цикл (квартал) та дозволяє моделі виявити місячні, квартальні та подієві патерни попиту. Прогнозний горизонт фіксовано на рівні двадцяти восьми днів, що відповідає стандартному тижневому циклу оновлення замовлень у ритейлі. Загальна довжина послідовності становить сто вісімнадцять елементів. Для генерації навчальної вибірки застосовано ковзаюче вікно з тижневим кроком зсуву, що забезпечує достатню кількість навчальних прикладів без суттєвої автокореляції між сусідніми пакетами. Така конфігурація забезпечує оптимальне співвідношення між репрезентативністю даних, обчислювальною ефективністю та відповідністю операційним вимогам торговельної мережі.

Таким чином, аналіз задач прогнозування у торговельних системах демонструє необхідність переходу від локальних однокрокових моделей до глобальних багатокрокових архітектур з вбудованими механізмами обробки ієрархії та оптимізованими конвеєрами підготовки даних. Це створює підґрунтя для застосування методів безперервного навчання, адаптації до змін розподілу та інкрементального донавчання, деталі яких будуть розглянуті у наступних розділах роботи.

1.4. Проблеми нестаціонарності, concept drift та cold start у задачах прогнозування продажів

Часові ряди обсягів продажів у роздрібній торгівлі належать до класу нестаціонарних процесів, статистичні властивості яких, зокрема математичне сподівання, дисперсія та автокореляційна структура, змінюються в часі. На відміну від стаціонарних рядів, де патерни попиту залишаються відносно стабільними, дані роздрібних мереж піддаються постійному впливу ендогенних та екзогенних факторів, що зумовлює явище концептуального дрейфу. Математично це виражається у зміні умовного розподілу цільової змінної відносно часу:

$$P_t(Y \mid X) \neq P_{t+\Delta}(Y \mid X), \tag{1.3}$$

де модель, навчена на історичних даних, поступово втрачає свою прогнозну здатність, оскільки зв'язки між екзогенними ознаками та обсягами продажів трансформуються під впливом змін ринкових умов.

У контексті прогнозування попиту виділяють три основні форми концептуального дрейфу, кожна з яких вимагає специфічних підходів до адаптації моделей[12]. Раптовий дрейф виникає внаслідок дискретних подій, таких як запуск масштабних промо-акцій, різка зміна цінової політики, поява товарів-замінників або зовнішні економічні шоки. Він характеризується стрибкоподібною зміною рівня або волатильності часового ряду, що потребує швидкої реакції алгоритму протягом короткого проміжку часу. Поступовий дрейф пов'язаний із повільною зміною споживчої поведінки, інфляційними процесами або плавним зсувом сезонних трендів. Розподіл даних у цьому випадку трансформується неухильно, але непомітно на коротких інтервалах спостереження, що ускладнює його статистичне виявлення та вимагає

застосування механізмів інкрементального оновлення параметрів моделі. Циклічний дрейф проявляється у регулярних повторюваних зсувах, зумовлених календарними подіями, тижневими патернами покупок або сезонними циклами. Для глибоких нейронних мереж цей тип дрейфу частково компенсується завдяки циклічному кодуванню темпоральних ознак та використанню календарних індикаторів, проте вимагає збереження довгострокової пам’яті про аналогічні минулі патерни.

Окремою фундаментальною складністю є явище холодного старту, яке виникає, коли для нового товару, торгової точки або регіону відсутня достатня історична інформація для формування надійного прогнозу. У глобальних нейронних моделях ця проблема посилюється архітектурно, оскільки категоріальні ідентифікатори представляються у вигляді векторних представлень[4], які ініціалізуються випадково та оптимізуються лише за наявності достатньої кількості спостережень. Для абсолютно нових товарів модель стикається з відсутністю історичного контексту, що унеможливлює коректне використання механізмів уваги та виділення темпоральних залежностей у рекурентних блоках. У реальних торговельних мережах частка нових товарних позицій у загальному асортименті може досягати п’яти–десяти відсотків щорічно, тому ігнорування проблеми холодного старту призводить до систематичної похибки на етапі запуску продукту, що безпосередньо впливає на рівень сервісу та обсяги товарних залишків.

Відсутність ефективних механізмів протидії нестаціонарності та холодному старту в автоматизованих системах прогнозування має критичні операційні та фінансові наслідки. Насамперед виникає проблема катастрофічного забування, коли при спробі донавчити модель на нових даних без застосування спеціалізованих стратегій регуляризації відбувається перезапис раніше засвоєних

патернів, що призводить до різкого падіння точності прогнозів для відомих товарних позицій. Крім того, формується систематична похибка прогнозування, яка з часом накопичується та спотворює процеси логістичного планування, формування замовлень та бюджетування. Втрата операційної довіри до системи з боку фахівців із закупівель та логістики часто призводить до повернення до ручних методів планування, що знижує загальну ефективність управління ланцюгами постачання та економічну доцільність автоматизованих рішень.

У рамках даного дослідження запропоновано комплексний підхід до подолання зазначених обмежень, реалізований на рівні архітектури даних, синтетичної аугментації та алгоритмів безперервного навчання. Для моделювання умов холодного старту в навчальній вибірці було виділено окремий сегмент товарів, які повністю виключалися з етапу попереднього навчання, проте їхні ідентифікатори зберігалися в словнику векторних представлень. Ці товари було розподілено на дві підмножини: першу, призначену для адаптації моделі після виявлення нового товару, та другу, яка слугувала для валідації якості прогнозів після процедури донавчання. Під час тренування застосовувалася синтетична аугментація, що з імовірністю 0,15 обнуляла частину або все історичне вікно, імітуючи відсутність продажів та змушуючи архітектуру нейронної мережі покладатися на статичні атрибути товарів та календарний контекст. Для імітації концептуального дрейфу реалізовано процедуру адитивної аугментації, яка створює трансформовані копії навчальних послідовностей та об’єднує їх з оригінальними даними. Масштабування попиту та цін виконувалося в заданих діапазонах із рандомним моментом початку дрейфу для кожного семпла, що забезпечує одночасне навчання моделі на чистих та змінених даних. Такий підхід підвищує стійкість архітектури до змін розподілу без втрати

узагальнювальної здатності та без необхідності зберігання повних історичних вибірок в оперативній пам’яті.

Інтеграція механізмів симуляції холодного старту та адитивного дрейфу на етапах підготовки даних і навчання дозволяє формалізувати вимоги до адаптивності моделі в умовах реальної експлуатації. Це створює теоретико-методологічну основу для застосування підходів безперервного навчання, детальний аналіз яких буде представлено в наступному розділі, а практична реалізація та експериментальна валідація — у подальших розділах роботи.

1.5. Аналіз існуючих підходів і методів прогнозування продажів

Сучасний стан досліджень у галузі прогнозування часових рядів характеризується поступовим переходом від класичних статистичних методів до глибоких нейронних архітектур та інкрементальних підходів до навчання. Вибір оптимального інструментарію для роздрібної торгівлі визначається необхідністю балансу між точністю багатокрокового прогнозу, інтерпретованістю результатів, обчислювальною ефективністю та здатністю системи адаптуватися до змін у розподілі даних без втрати раніше засвоєних знань. До класичних статистичних моделей належать ARIMA (Autoregressive Integrated Moving Average), ETS (Error, Trend, Seasonality) та Prophet, які базуються на декомпозиції ряду на трендову, сезонну та залишкову компоненти з подальшим екстраполюванням. Їхніми перевагами є висока інтерпретованість, низькі вимоги до обчислювальних ресурсів та ефективність на коротких горизонтах для стаціонарних або слабо нестаціонарних рядів. Проте в умовах сучасних торговельних систем вони виявляють суттєві обмеження. По-перше, вони мають локальний характер, що

унеможливлює спільне навчання на тисячах паралельних рядів та ігнорує перехресні залежності між ними. По-друге, інтеграція екзогенних змінних вимагає ручного розширення специфікації моделі, що суттєво ускладнює процес калібрування та масштабування. По-третє, ці методи не мають вбудованих механізмів протидії структурним зламам та концептуальному дрейфу, а їхня реакція на появу нових товарів зводиться до застосування усереднених історичних базових ліній, що є неприйнятним для мереж із постійним оновленням асортименту.

Алгоритми на основі градієнтного бустингу над деревами рішень, зокрема LightGBM, XGBoost та CatBoost, набули широкого застосування в промислових задачах прогнозування[13] завдяки високій швидкості обчислень, стійкості до викидів та здатності моделювати нелінійні взаємодії. У контексті аналізу часових рядів вони використовуються як глобальні моделі після попереднього перетворення послідовностей у табличний формат із використанням лагових ознак, ковзних статистик та календарних змінних. Попри високі показники точності, ці підходи мають критичні недоліки для побудови адаптивних систем. Вони вимагають інтенсивного ручного конструювання ознак, що є трудомістким процесом та чутливим до суб’єктивних помилок експертів. Такі моделі не здатні ефективно вловлювати довгострокові темпоральні залежності без експоненційного зростання розмірності ознакового простору. При зміні розподілу вхідних даних вони потребують повного перенавчання на оновленій вибірці, що часто призводить до катастрофічного забування раніше засвоєних патернів. Крім того, стандартні реалізації не підтримують нативне ймовірнісне прогнозування, генеруючи лише детерміновані точкові оцінки без інтервалів невизначеності.

Останніми роками спостерігається активне впровадження глибоких нейронних архітектур, здатних одночасно навчатися на великих масивах спостережень, автоматично виділяти ієрархічні ознаки та генерувати ймовірнісні інтервали. Серед них виділяють модель DeepAR, яка є авторегресійною ймовірнісною архітектурою на базі рекурентних нейронних мереж та генерує шляхи вибірки для оцінки розподілу майбутніх значень[18], проте має обмежену інтерпретованість механізмів впливу екзогенних змінних. Архітектура N-BEATS, що базується на повністю зв’язаних шарах із розкладанням на трендову та сезонну бази, демонструє високу точність точкових прогнозів, але не підтримує явну обробку статичних та динамічних категоріальних ознак у нативному форматі. Найбільш перспективною для задач роздрібного прогнозування визнається модель часового трансформера з тимчасовою фузією (Temporal Fusion Transformer, TFT)[14]. Вона поєднує механізми багатоголової уваги, блоки довгої короткочасної пам’яті та мережі відбору змінних для автоматичного визначення релевантних ознак на кожному часовому кроці. Ключовими перевагами цієї архітектури є підтримка окремих тензорів для історичних числових, майбутніх числових та статичних категоріальних ознак, вбудоване квантильне прогнозування для оцінки невизначеності, інтерпретованість через візуалізацію ваг уваги та важливості змінних, а також стійкість до шуму завдяки регуляризації. Саме ці характеристики роблять її оптимальним базовим модулем для адаптивної системи прогнозування.

Стаціонарне навчання нейронних мереж є недостатнім для середовищ із потоковим надходженням даних, що зумовлює необхідність застосування методів безперервного навчання. Для подолання катастрофічного забування та забезпечення інкрементальної адаптації використовуються кілька основних підходів. Метод досвідченого відтворення (Experience Replay) передбачає

збереження репрезентативної підвибірки історичних даних у спеціалізованому буфері[19] та її періодичне змішування з новими пакетами даних, що забезпечує високу стабільність, але вимагає додаткової пам’яті та складних механізмів балансування співвідношення старих та нових семплів. Підхід еластичного консолідування ваг (Elastic Weight Consolidation, EWC) застосовує регуляризацію, яка оцінює діагональну апроксимацію матриці Фішера для визначення найважливіших параметрів моделі та штрафує їхні зміни під час донавчання. Цей метод ефективно зберігає раніше засвоєні знання, проте є обчислювально витратним для великих архітектур. Альтернативою виступає метод низькорангової адаптації (Low-Rank Adaptation, LoRA), який передбачає заморожування базових ваг моделі та додавання низькорангових матриць[9] до лінійних шарів, паралельно дозволяючи оновлення шарів вкладень для нових сутностей[15]. Цей підхід забезпечує мінімальне збільшення кількості параметрів, що підлягають оптимізації, швидку адаптацію до відсутності історичних даних та концептуального дрейфу, а також компактне збереження адаптерів, що є критичним для хмарного розгортання. Динамічні архітектури, які розширюють мережу новими нейронами або шарами при виявленні значних змін, ускладнюють процес інференсу та вимагають складних механізмів маршрутизації даних.

Порівняльний аналіз існуючих підходів демонструє, що жоден із класичних чи ізольованих методів глибокого навчання не забезпечує повного покриття вимог сучасних торговельних систем. Зокрема, відсутній інструмент, який би одночасно гарантував ефективну обробку тисяч паралельних рядів, квантильне прогнозування, інтерпретованість результатів, стійкість до концептуального дрейфу та коректну ініціалізацію для нових об’єктів. Це обумовлює доцільність переходу до гібридної парадигми, що поєднує глобальну архітектуру

трансформера з механізмами безперервного навчання та синтетичної аугментації даних. Обґрунтування архітектурного вибору та інтеграції зазначених компонентів буде детально розглянуто в наступному підрозділі.

1.6. Обґрунтування вибору методів машинного навчання для вирішення поставленої задачі

На підставі аналізу предметної області, специфіки роздрібних даних та обмежень традиційних підходів, сформульовано систему вимог до архітектури прогнозування. Вона має одночасно забезпечувати глобальне багатокрокове прогнозування тисяч паралельних часових рядів, ймовірнісне оцінювання невизначеності, інтерпретованість впливу екзогенних змінних, стійкість до концептуального дрейфу, коректну ініціалізацію для товарів без історії продажів, а також інкрементальну адаптацію без втрати раніше засвоєних знань. Оскільки жоден ізольований метод не покриває весь спектр цих вимог, обрано підхід до побудови гібридної системи на базі моделі часового трансформера з тимчасовою фузією, доповненої механізмами безперервного навчання та синтетичної аугментації даних.

Архітектура часового трансформера з тимчасовою фузією обрана як базовий модуль через її нативну підтримку гетерогенних вхідних даних, що є критично важливим для роздрібної торгівлі. Модель розділяє ознаки на три логічні потоки: статичні категоріальні, історичні числові та відомі майбутні числові змінні. Вбудований механізм автоматичного відбору змінних динамічно визначає вагомість кожної ознаки на кожному часовому кроці, усуваючи потребу в ручному конструюванні ознак та зменшуючи вплив шуму. На відміну від класичних рекурентних архітектур, ця модель поєднує блоки довгої

короткочасної пам’яті для уловлювання локальних залежностей із механізмом багатоголової уваги для моделювання довгострокових взаємодій між подіями, що суттєво покращує точність прогнозів на горизонті до двадцяти восьми днів. Важливим критерієм вибору є вбудована підтримка квантильної регресії. Модель генерує прогнози для заданого набору квантилів безпосередньо під час навчання, що дозволяє оцінювати інтервали невизначеності, розраховувати ймовірнісні метрики покриття та інтервальні похибки, а також приймати управлінські рішення з урахуванням ризиків дефіциту або надлишку товарних залишків. Крім того, архітектура забезпечує інтерпретованість результатів через аналіз ваг уваги та важливості вхідних змінних, що підвищує довіру фахівців до автоматизованих прогнозів.

Статичне навчання моделі на фіксованій історичній вибірці призводить до швидкої деградації якості прогнозів в умовах потокового надходження даних. Повний перерахунок моделі з нуля є обчислювально неефективним і суперечить принципу інкрементальності. Тому обрано підхід безперервного навчання, реалізований через комбінацію методів збереження досвіду, еластичного консолідування ваг та низькорангової адаптації з одночасним оновленням шарів векторних представлень. Метод низькорангової адаптації дозволяє заморозити базові ваги архітектури та навчати лише низькорангові матриці в шарах уваги, паралельно забезпечуючи оновлення векторних представлень для нових товарних позицій. Це забезпечує швидку адаптацію до відсутності історичних даних при мінімальному зростанні кількості параметрів, що підлягають оптимізації, та компактному збереженні адаптерів, що є критичним для ефективного хмарного розгортання. Для підвищення узагальнювальної здатності ще на етапі початкового навчання застосовано синтетичну аугментацію. Підхід до імітації відсутності історії зі заданою ймовірністю обнуляє частину або все

історичне вікно, змушуючи модель покладатися на статичні атрибути та календарний контекст. Адитивна аугментація даних генерує трансформовані копії навчальних послідовностей зі зсувом попиту та цін[16] у заданих діапазонах, при цьому момент початку дрейфу моделюється випадковим чином для кожного семпла. Такий підхід дозволяє одночасно навчати модель на чистих та змінених даних, підвищуючи її стійкість до змін розподілу без реального лагу в надходженні інформації та без втрати оригінальних навчальних прикладів.

Обрана методологія безпосередньо інтегрується в сучасні принципи автоматизації життєвого циклу машинного навчання та хмарного розгортання. Підготовка даних реалізована через використання відображених у пам’ять масивів, що забезпечує потокову обробку великих обсягів інформації без створення дублікатів в оперативній пам’яті. Батчова структура наборів даних та чітке розбиття на навчальну, валідаційну, тестову та додаткові вибірки для адаптації дозволяють стандартизувати конвеєр навчання та інференсу, що є основою для реалізації автоматизованих процесів розгортання. Модульність системи, що передбачає окремі компоненти для попередньої обробки даних, базового навчання, інференсу та адаптивного донавчання, спрощує інтеграцію з хмарними сервісами моніторингу, реєстрації моделей та автоматичного запуску процедур оновлення при перевищенні порогових значень деградації якості. Легкі адаптери та оптимізовані датасети забезпечують низьку затримку обробки запитів та економічну ефективність обчислювальних ресурсів, що робить систему готовою до масштабування на десятки тисяч товарних позицій та сотні торгових точок.

Таким чином, поєднання архітектури часового трансформера, підходів безперервного навчання, синтетичної аугментації та інженерних рішень, орієнтованих на хмарне середовище, утворює цілісну методологічну основу

дослідження. Вона безпосередньо відповідає на виклики нестаціонарності даних, відсутності історичної інформації для нових товарів та катастрофічного забування, сформульовані в попередніх підрозділах. Обрана сукупність методів створює підґрунтя для детального формалізованого опису алгоритмів, архітектурних рішень та експериментальної валідації, які будуть представлені в наступних розділах роботи.

Висновки до розділу 1

У першому розділі проведено системний аналіз задачі прогнозування попиту в роздрібних торговельних мережах за умов потокового надходження даних. Встановлено, що сучасне ринкове середовище характеризується високою нестаціонарністю часових рядів, наявністю концептуального дрейфу, ієрархічною багатовимірністю попиту та частим оновленням асортименту, що обумовлює проблему «холодного старту». На підставі порівняльного огляду архітектур глибокого навчання обрано модель часового трансформера з тимчасовою фузією (TFT) як базовий модуль, що забезпечує глобальне багатокрокове прогнозування, квантильне оцінювання невизначеності, автоматичний відбір змінних та інтерпретованість результатів. Для подолання деградації моделі в умовах змін розподілу запропоновано інтеграцію підходів безперервного навчання, зокрема низькорангової адаптації (LoRA) та синтетичної аугментації даних, що імітують концептуальний дрейф та відсутність історичних спостережень. Реалізований інженерний конвеєр, який включає вертикальну нормалізацію даних, синусоїдальне кодування темпоральних ознак, використання відображених у пам’ять масивів та оптимізовану генерацію ковзних вікон (90 днів історії та 28 днів горизонту), забезпечує масштабованість системи та ефективне використання обчислювальних ресурсів. У сукупності сформовано цілісну

теоретико-методологічну основу, яка повною мірою відповідає на виклики нестаціонарності, холодного старту та катастрофічного забування, і створює підґрунтя для подальшого формалізованого опису архітектури, налаштування алгоритмів адаптивного донавчання та експериментальної верифікації запропонованого рішення в наступних розділах роботи.

РОЗДІЛ 2. МЕТОДИ ТА МЕТОДИКИ ДОСЛІДЖЕННЯ ДЛЯ ПОБУДОВИ АДАПТИВНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПРОДАЖІВ

2.1. Методи прогнозування часових рядів у машинному навчанні

Сучасний стан досліджень у галузі прогнозування часових рядів характеризується переходом від локальних статистичних моделей до глобальних нейронних архітектур, здатних спільно навчатися на тисячах паралельних рядів та ефективно інтегрувати гетерогенні екзогенні змінні. Серед існуючих підходів модель часового трансформера з тимчасовою фузією посідає центральне місце завдяки поєднанню рекурентних механізмів для кодування локальних залежностей, трансформерних шарів уваги для виявлення довгострокових контекстних зв'язків та вбудованої підтримки ймовірнісного прогнозування з високою інтерпретованістю впливу вхідних ознак.

Архітектура моделі оперує чітко структурованими вхідними даними, які розподіляються на три логічні групи залежно від їхньої темпоральної доступності. Нехай B позначає розмір навчального пакету, L — довжину історичного вікна, H — горизонт прогнозування, N_d — кількість динамічних числових ознак, а N_s — кількість статичних категоріальних ознак. Тоді вхідні тензори мають такі розмірності: історичні числові ознаки представляються тензором $\mathbf{X}_{\text{hist}} \in \mathbb{R}^{B \times L \times (1+N_d)}$, де перший канал містить логарифмовані значення цільової змінної, а решта каналів відповідають динамічним змінним минулого, таким як ціни, календарні індикатори та статуси наявності товару; відомі майбутні числові ознаки формують тензор $\mathbf{X}_{\text{fut}} \in \mathbb{R}^{B \times H \times N_d}$, який включає планові ціни та календарні події, доступні на момент формування прогнозу; статичні категоріальні ознаки описуються тензором $\mathbf{S} \in \mathbb{Z}^{B \times N_s}$, що містить ідентифікатори товару, торговельної точки, категорії та регіону. Така

формалізація забезпечує суворе розділення темпоральних та статичних контекстів, що є необхідною умовою для коректного функціонування компонентів моделі.

Перетворення вхідних даних розпочинається з кодування статичних ознак, які перетворюються на щільні векторні представлення за допомогою шарів вкладень, розмірність яких адаптується до кількості унікальних значень кожної змінної. Результуючі вектори обробляються мережею з залишковими зв'язками та воротами, після чого нормалізуються, формуючи контекстний вектор, що використовується для масштабування та активації всіх наступних шарів. Це забезпечує умовну залежність прогнозу від ієрархічних атрибутів. Для динамічних ознак застосовується механізм автоматичного відбору змінних, який обчислює ваги релевантності на кожному часовому кроці. Вхідні ознаки зважуються пропорційно до їхньої інформативності, що дозволяє алгоритму динамічно ігнорувати шумові або неінформативні змінні, зменшуючи ризик перевчення та підвищуючи інтерпретованість результатів.

Кодування темпоральних залежностей здійснюється у два етапи. Спочатку відібрані історичні ознаки обробляються блоком довгої короткочасної пам'яті, який формує послідовність прихованих станів, що фіксують короткострокові тренди, автокореляційні патерни та ефекти згасання впливу минулих подій. Рекурентна структура доповнюється прямими зв'язками для стабілізації градієнтного потоку під час зворотного поширення помилки. На другому етапі механізм багатоголової уваги розширює кодування глобальним контекстом. Для кожного кроку прогнозного горизонту обчислюється зважена комбінація історичних станів на основі подібності запитів та ключів:

$$\alpha_{t,h} = \mathrm{softmax}\left(\frac{(\mathbf{W}_q \mathbf{h}_h^{\mathrm{fut}})(\mathbf{W}_k \mathbf{h}_t^{\mathrm{enc}})^\top}{\sqrt{d_k}}\right), \quad \mathbf{z}_h = \sum_{t=1}^{L} \alpha_{t,h} \mathbf{W}_v \mathbf{h}_t^{\mathrm{enc}}. \qquad (2.1)$$

Це дозволяє алгоритму звертатися до релевантних минулих подій незалежно від їхньої часової відстані, ефективно моделюючи довгострокові залежності.

Отримані вектори уваги трансформуються у прогнозні розподіли за допомогою квантильного декодера. На відміну від детермінованих регресорів, модель генерує оцінки для заданого набору квантилів[17]. Для кожного рівня ймовірності застосовується окремий лінійний проектор, що мінімізує спеціалізовану функцію втрат, відому як функція втрат квантильної регресії:

$$\mathcal{L}_{\mathrm{quantile}} = \frac{1}{B \cdot H} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{q \in \mathcal{Q}} \max\left(q \cdot (y_{b,h} - \hat{y}_{b,h}^{(q)}), (q-1) \cdot (y_{b,h} - \hat{y}_{b,h}^{(q)})\right), \qquad (2.2)$$

де $y_{b,h}$ — фактичне значення цільової змінної, а $\hat{y}_{b,h}^{(q)}$ — прогнозне значення для квантиля рівня q. У рамках даного дослідження набір квантилів охоплює діапазон від десяти до дев'яноста відсотків, що дозволяє будувати надійні інтервали довіри та медіанні прогнози. Архітектурні гіперпараметри, такі як розмірність прихованого стану, кількість шарів рекурентної мережі, кількість голів уваги та коефіцієнт регуляризації, підбираються експериментально з урахуванням балансу між обчислювальною складністю та точністю прогнозування.

Таким чином, поєднання глобального навчання, механізмів автоматичного відбору ознак, комбінації рекурентного та трансформерного кодування та ймовірнісного декодування робить дану архітектуру оптимальним базовим модулем. Її структурна гнучкість та здатність до інтерпретації ваг ознак

створюють необхідне підґрунтя для інтеграції з алгоритмами безперервного навчання та адаптації до змін розподілу даних, що буде детально розглянуто в наступних підрозділах.

2.2. Підходи до інкрементального та безперервного навчання моделей

У умовах постійного потокового надходження транзакційних даних та регулярного оновлення асортименту торговельних мереж статичне навчання моделей стає недостатнім для підтримки високої прогнозної точності. Навчання з нуля на кожній новій порції даних є обчислювально неефективним, тоді як ігнорування нових спостережень призводить до систематичної деградації якості прогнозів. Це зумовлює необхідність застосування підходів безперервного навчання, які дозволяють інкрементально адаптувати параметри моделі до нових розподілів даних, зберігаючи при цьому раніше засвоєні універсальні темпоральні патерни. У даному дослідженні реалізовано та порівняно чотири стратегії донавчання, кожна з яких пропонує власний механізм балансування між пластичністю моделі та стабільністю її знань.

Базовий підхід до інкрементальної адаптації передбачає повне розморожування всіх параметрів нейронної мережі та їхню подальшу оптимізацію на новій вибірці за допомогою стандартних алгоритмічних методів градієнтного спуску. Цей метод вирізняється мінімальними накладними витратами на реалізацію, швидкою збіжністю та прямою адаптацією до локальних особливостей попиту. Проте його фундаментальним недоліком є явище катастрофічного забування, коли градієнтні кроки, спрямовані на мінімізацію втрат на нових даних, поступово перезаписують ваги, що кодували загальні сезонні залежності, календарні ефекти та цінову еластичність для відомих товарів. У результаті модель демонструє покращення на цільовій підмножині, але втрачає узагальнювальну здатність на стабільних часових рядах,

що робить цей підхід непридатним для автономної експлуатації без додаткових обмежень.

Для протидії катастрофічному забуванню застосовано стратегію повторного відтворення досвіду, яка передбачає змішування нових навчальних семплів із репрезентативною підвибіркою історичних даних. Під час кожного кроку оптимізації навчальний пакет формується як конкатенація нових спостережень та випадково вибраних прикладів із раніше зібраної валідаційної вибірки. Співвідношення даних контролюється спеціальним коефіцієнтом, який резервує частину пакету під історичні семпли, що забезпечує різноманітність патернів без надмірного навантаження на обчислювальні ресурси. Циклічне використання буферних даних гарантує рівномірне відтворення минулих знань протягом усіх епох навчання. Метод демонструє високу стабільність та ефективне збереження раніше засвоєних залежностей, проте вимагає додаткової оперативної пам'яті для зберігання буфера та ретельного балансування градієнтів між старими та новими прикладами.

Альтернативним регуляризаційним підходом виступає метод еластичного консолідування ваг, який базується на оцінці важливості кожного параметра моделі для попередніх задач за допомогою діагональної апроксимації матриці Фішера. Важливість параметра обчислюється як очікуване значення квадрата градієнта функції втрат щодо цього параметра на розподілі історичних даних:

$$F_i = \mathbb{E}_{x \sim p_{\text{old}}} \left[\left(\frac{\partial \log p(y|x, \theta)}{\partial \theta_i} \right)^2 \right], \tag{2.3}$$

Під час інкрементального навчання до основної функції втрат додається штрафний член, що обмежує зміни критичних параметрів:

$$\mathcal{L}_{\text{EWC}} = \mathcal{L}_{\text{new}} + \frac{\lambda_{\text{ewc}}}{2} \sum_i F_i(\theta_i - \theta_i^*)^2, \tag{2.4}$$

де θ_i^* позначає значення параметрів після попереднього навчання, а λ_{ewc} є гіперпараметром, що регулює баланс між пластичністю та стабільністю.

На практиці обчислення матриці Фішера для великих архітектур є обчислювально витратним та чутливим до числової нестабільності. Тому реалізовано механізм автоматичного переключення на метод L2-регуляризації з фіксованим опорним вектором ваг у випадку виявлення надмірно великих або нескінченних значень штрафу. Це забезпечує безперебійну роботу алгоритму навіть за умов нестабільних градієнтних оцінок.

Принципово іншу парадигму адаптації пропонує підхід параметрично ефективного донавчання на базі низькорангової адаптації. Замість модифікації всіх ваг моделі, до лінійних шарів додаються низькорангові матриці приросту, тоді як основні ваги залишаються замороженими. Для кожного цільового шару $\mathbf{W} \in \mathbb{R}^{d \times k}$ прямий прохід обчислюється за формулою:

$$\mathbf{h}' = \mathbf{W}\mathbf{x} + \frac{\alpha}{r}\mathbf{B}\mathbf{A}\mathbf{x}, \tag{2.5}$$

де $\mathbf{A} \in \mathbb{R}^{r \times k}$ та $\mathbf{B} \in \mathbb{R}^{d \times r}$ є навчальними матрицями низького рангу $r \ll \min(d, k)$, а коефіцієнт α контролює масштаб навчального сигналу.

Адаптери інтегруються виключно у лінійні проектори механізмів уваги, що запобігає руйнуванню архітектурної логіки базової мережі. Критичною особливістю реалізації є додаткове розморожування шарів категоріальних вкладень, що дозволяє моделі навчати нові векторні представлення для товарів, які раніше не брали участі в оптимізації. Це ефективно вирішує проблему

холодного старту без зміни решти параметрів мережі. У результаті кількість тренувальних параметрів становить менше одного відсотка від загальної кількості, а розмір збережених адаптерів є мінімальним, що робить цей метод оптимальним для хмарного розгортання та швидкого інференсу.

Кожен із розглянутих методів пропонує власний компроміс між обчислювальною ефективністю, стабільністю навчання та гнучкістю адаптації до змін ринкових умов. Базове донавчання слугує контрольною точкою для оцінки ступеня деградації без регуляризації, тоді як стратегія повторного відтворення забезпечує високу стабільність за рахунок додаткових вимог до пам’яті. Регуляризаційні підходи пропонують теоретично обґрунтоване обмеження зміни ваг, але вимагають обережного налаштування гіперпараметрів та додаткових обчислень. Низькорангова адаптація з оновленням шарів вкладень демонструє найкращу ефективність у ресурсообмежених сценаріях, мінімізуючи ризик переучення та забезпечуючи компактне зберігання станів моделі. Вибір оптимальної стратегії залежить від конкретних операційних вимог до латентності, доступності історичних даних та частоти появи нових товарів, що буде емпірично перевірено в експериментальній частині роботи.

2.3. Методи адаптації моделей до зміни розподілу даних

Умови потокового надходження транзакційних даних у роздрібній торгівлі вимагають від прогнозної системи здатності швидко адаптуватися до структурних змін у розподілі цільової змінної та екзогенних ознак. Традиційні підходи, що покладаються виключно на репрезентативність фіксованої навчальної вибірки, не забезпечують стійкості до появи нових товарних позицій, різких змін у ціновій політиці та сезонних зсувів попиту. Для подолання цих

обмежень у дослідженні застосовано комплекс методів синтетичної аугментації даних та спеціалізованої ініціалізації, інтегрованих безпосередньо в алгоритми навчання та інференсу.

Для імітації сценаріїв відсутності або обмеженості історичних даних реалізовано механізм стохастичного обнулення історичного вікна спостережень[4]. Під час кожного кроку оптимізації до вхідного тензора історичних ознак із заданою ймовірністю застосовується трансформація, яка частково або повністю занулює минулі значення. Розрізняють два режими модифікації: повне обнулення всіх кроків історичного періоду, що симулює абсолютно нову товарну позицію без попередніх спостережень, та часткове обнулення початкової частини вікна, що моделює ситуацію обмеженої історичної інформації, коли товар з'явився нещодавно. Математично ця трансформація виражається через елементне множення вхідного тензора на бінарну маску:

$$\mathbf{X}_{\text{hist}}^{(t)} = \mathbf{M}_{\text{cold}} \odot \mathbf{X}_{\text{hist}}, \tag{2.6}$$

де маска генерується стохастично згідно з визначеними ймовірнісними розподілами. Такий підхід змушує нейронну мережу покладатися на статичні категоріальні ознаки та календарні індикатори замість минулих продажів, формуючи узагальнені уявлення про попит на рівні товарних категорій та відділів ще на етапі попереднього навчання.

Для підвищення стійкості моделі до змін розподілу попиту та цін застосовано метод синтетичної аугментації зсуву даних, який не замінює оригінальні спостереження, а доповнює навчальний пакет їхніми трансформованими копіями. Для кожного навчального прикладу незалежно генеруються множники попиту та цін, рівномірно розподілені у заданих діапазонах. Момент початку зсуву вибирається випадковим чином із усього

діапазону історичного вікна, що імітує ситуацію, коли структурна зміна відбувається в довільний момент часу. На відміну від історичної частини, прогнозний горизонт завжди піддається повному масштабуванню, оскільки алгоритм має навчитися передбачати вже трансформовані значення в майбутньому. Оскільки модель оптимізується в логарифмічному масштабі продажів, процедура перетворення виконується через цикл відновлення оригінального масштабу, масштабування та повернення до логарифмічного вигляду для збереження числової стабільності:

$$y_{\text{drift}} = \log\left(1 + \max\left(0,\ (\exp(y_{\log}) - 1) \cdot d\right)\right), \tag{2.7}$$

$$x_{\text{price,drift}} = x_{\text{price}} \cdot p, \tag{2.8}$$

де $y_{\log}$ позначає логарифмований канал історичних або майбутніх продажів. Монотонність перетворення гарантується додатністю похідної:

$$\frac{\partial y_{\text{drift}}}{\partial y_{\log}} = \frac{d \cdot \exp(y_{\log})}{1 + \max\left(0,\ (\exp(y_{\log}) - 1) \cdot d\right)} > 0 \quad \forall d > 0, \tag{2.9}$$

що запобігає інверсії знаків градієнтів та зберігає коректність оптимізації. Трансформовані копії конкатенуються з оригінальними даними, забезпечуючи одночасне навчання на стабільних та змінених розподілах.

Для подолання проблеми відсутності навчених векторних представлень нових товарів під час інференсу розроблено механізм проксі-підстановки ідентифікаторів на рівні статичних ознак. Оскільки ідентифікатори нових позицій не брали участі в оптимізації базової моделі, їхні початкові вектори ініціалізуються випадковим чином і є непридатними для точного прогнозування. Під час формування вхідних даних оригінальний ідентифікатор товару замінюється на ідентифікатор відомої позиції зі схожою ієрархічною

приналежністю. Реалізовано дві стратегії підбору: випадковий вибір відомого товару з тієї ж товарної категорії, що дозволяє моделі використати навчені патерни попиту, та вибір медіанного представника зі списку товарів того ж відділу, що забезпечує більш консервативну оцінку. У разі відсутності відповідних відомих товарів застосовується резервний механізм підстановки стандартного ідентифікатора, що запобігає критичним помилкам виконання. Заміна відбувається на етапі підготовки тензора категоріальних ознак перед прямим проходом мережі, що дозволяє використовувати вже оптимізовані шари вкладень без зміни архітектурної структури моделі.

Поєднання аугментації для моделювання відсутності історії, синтетичного зсуву розподілу на етапі навчання та проксі-ініціалізації під час інференсу формує замкнений цикл адаптації прогнозної системи. Модель навчається генерувати коректні прогнози навіть за відсутності минулих спостережень, демонструє стійкість до різких змін ринкових умов, а механізм підстановки категоріальних представлень усуває початковий бар'єр без необхідності повного перерахунку параметрів. Ці методи створюють теоретико-прикладну основу для подальшого оцінювання якості прогнозування за допомогою спеціалізованих метрик, аналіз яких буде представлено в наступному підрозділі.

2.4. Методи виявлення змін розподілу даних у задачах прогнозування

У умовах потокового надходження транзакційних даних та динамічності ринкового середовища своєчасне виявлення змін у розподілі вхідних змінних є критично важливим для підтримки прогнозної точності моделей. Явище концептуального дрейфу, що математично виражається у зміні умовного розподілу цільової змінної відносно часу, вимагає застосування статистичних

методів детекції, здатних кількісно оцінити ступінь відхилення поточних даних від базового навчального розподілу. У дослідженні реалізовано комплексний підхід до моніторингу, який поєднує індекс стабільності популяції, критерій Колмогорова–Смірнова, метод максимальної різниці середніх та аналіз деградації метрик на ковзному вікні. Кожен із цих методів застосовується на відповідному рівні агрегації даних, забезпечуючи повне покриття різних типів змін розподілу.

Індекс стабільності популяції є стандартним інструментом для оцінки узгодженості розподілів як категоріальних, так і неперервних змінних. Його розрахунок базується на порівнянні часток спостережень у заданих інтервалах між очікуваним та фактичним розподілами:

$$\mathrm{PSI} = \sum_{i=1}^{k} \left(P_i^{\mathrm{actual}} - P_i^{\mathrm{expected}} \right) \cdot \ln \left(\frac{P_i^{\mathrm{actual}}}{P_i^{\mathrm{expected}}} \right), \qquad (2.10)$$

де P_i^{actual} та P_i^{expected} позначають частки спостережень у i-му інтервалі поточного та базового розподілів відповідно, а k — кількість інтервалів. Інтерпретація отриманого значення спирається на емпіричні пороги: показник менший за 0.1 свідчить про стабільність, значення в діапазоні від 0.1 до 0.2 вказує на помірний дрейф, що потребує спостереження, а перевищення 0.2 слугує підставою для запуску процедур адаптації моделі. На практиці цей індекс обчислюється для ключових динамічних змінних, а його агрегація на рівні товарних категорій дозволяє зменшити вплив стохастичного шуму, притаманного окремим позиціям з низькими обсягами продажів.

Для детекції раптових зсувів розподілу доцільно застосовувати непараметричний критерій Колмогорова–Смірнова[20], який порівнює емпіричні функції розподілу двох вибірок без необхідності попереднього бінінгу даних.

Статистика тесту визначається як максимальна абсолютна відстань між кумулятивними функціями розподілу базової та поточної вибірок:

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)|, \tag{2.11}$$

де F_n та G_m — емпіричні функції розподілу вибірок обсягом n та m спостережень відповідно. Гіпотеза про однаковість розподілів відхиляється, якщо розраховане значення перевищує критичну межу $D_{\text{crit}} = c(\alpha) \cdot \sqrt{\frac{n+m}{n \cdot m}}$, обумовлену рівнем значущості α. Цей метод демонструє високу чутливість до різких змін, зумовлених промо-акціями або зовнішніми шоками, проте вимагає обережного застосування на малих вибірках через ризик хибних спрацьовувань.

У випадках, коли зміни торкаються взаємодії кількох ознак одночасно, використовують метод максимальної різниці середніх, який порівнює багатовимірні розподіли у просторі відтворюваного ядра. Його математична основа базується на оцінці відстані між середніми векторами вибірок у гілбертовому просторі:

$$\text{MMD}^2(\mathcal{D}_0, \mathcal{D}_t) = \mathbb{E}_{x,x' \sim \mathcal{D}_0}[k(x,x')] + \mathbb{E}_{y,y' \sim \mathcal{D}_t}[k(y,y')] - 2\mathbb{E}_{x \sim \mathcal{D}_0, y \sim \mathcal{D}_t}[k(x,y)], \tag{2.12}$$

де $k(\cdot,\cdot)$ є функцією ядра, зазвичай гаусовою. Цей підхід дозволяє виявляти нелінійні зміни у взаємозв'язках між змінними, які залишаються непомітними при одновимірному аналізі, наприклад, зміну цінової еластичності попиту залежно від календарного контексту. Через квадратичну обчислювальну складність метод застосовується вибірково або на агрегованих даних.

Окрім прямого аналізу розподілів вхідних даних, важливим індикатором дрейфу виступає моніторинг деградації прогнозних метрик на ковзному вікні,

який реалізовано на засадах статистичного контролю процесів. Для кожної оцінюваної метрики на інтервалі W днів обчислюються ковзне середнє μ_t та стандартне відхилення σ_t:

$$\mu_t = \frac{1}{W} \sum_{i=t-W+1}^{t} M_i, \quad \sigma_t = \sqrt{\frac{1}{W-1} \sum_{i=t-W+1}^{t} (M_i - \mu_t)^2}, \tag{2.13}$$

Сигнал про наявність дрейфу формується у разі перевищення верхньої контрольної межі $\mathrm{UCL} = \mu_{\text{base}} + z \cdot \sigma_{\text{base}}$ протягом заданої кількості послідовних періодів, де параметри μ_{base} та σ_{base} визначаються на етапі стабільної експлуатації моделі. Такий підхід особливо ефективний для виявлення поступового погіршення якості прогнозів, коли адаптація має бути ініційована до досягнення критичних рівнів похибки.

Інтеграція зазначених методів у систему моніторингу здійснюється ієрархічно з урахуванням обчислювальної складності та частоти оновлення даних. Оперативний контроль базується на аналізі ковзних метрик, що дозволяє швидко реагувати на зміни без зберігання повних історичних розподілів. Тижневий цикл включає розрахунок індексу стабільності та застосування критерію Колмогорова–Смірнова для ключових змінних на рівні категорій, забезпечуючи баланс між точністю детекції та витратами ресурсів. Щомісячний багатовимірний аналіз за допомогою методу максимальної різниці середніх завершує цикл перевірки, фокусуючись на складних нелінійних взаємодіях ознак. Перевищення встановлених порогових значень автоматично генерує подію для запуску пайплайну адаптивного донавчання, де обирається оптимальна стратегія оновлення параметрів залежно від типу та інтенсивності виявленого дрейфу. Поєднання статистичної детекції з аналізом прогнозних метрик формує надійний механізм своєчасного виявлення змін у розподілі даних, що є

необхідною умовою для функціонування адаптивної системи прогнозування в умовах динамічного торговельного середовища.

2.5. Метрики оцінювання якості моделей прогнозування продажів

Об'єктивне порівняння алгоритмів прогнозування та оцінка їхньої придатності для прийняття управлінських рішень у роздрібній торгівлі вимагає застосування комплексної системи показників. Традиційна оцінка лише за точністю центрального прогнозу є недостатньою в умовах використання ймовірнісних нейронних архітектур та підходів безперервного навчання, оскільки не відображає каліброваність інтервалів невизначеності, стійкість до змін розподілу даних та ефективність інкрементальної адаптації. У дослідженні запропоновано трирівневу систему оцінювання, що включає точкові показники похибки, ймовірнісні індекси калібрування та спеціалізовані метрики стабільності моделей.

Для аналізу якості медіанного прогнозу застосовуються три комплементарні показники, що дозволяють оцінити різні аспекти точності. Середня абсолютна похибка відображає середній модуль відхилення прогнозу від фактичного значення в натуральних одиницях і має лінійну функцію штрафу, завдяки чому демонструє стійкість до рідкісних викидів. Середньоквадратична похибка посилює вагу великих відхилень за рахунок квадратичної функції втрат, що є критично важливим для торгівлі, оскільки значні недопостачання або надлишки залишків генерують непропорційно високі логістичні витрати. Симетрична середня абсолютна відсоткова похибка усуває асиметрію класичного відсоткового показника та стабільно функціонує при малих або нульових обсягах

продажів завдяки регуляризаційному доданку в знаменнику. Математично ці метрики виражаються відповідно як

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i^{(0.5)}|, \tag{2.14}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left(y_i - \hat{y}_i^{(0.5)}\right)^2}, \tag{2.15}$$

$$\mathrm{sMAPE} = \frac{200}{N}\sum_{i=1}^{N} \frac{|y_i - \hat{y}_i^{(0.5)}|}{|y_i| + |\hat{y}_i^{(0.5)}| + \varepsilon}, \tag{2.16}$$

де N — кількість спостережень, y_i — фактичне значення, $\hat{y}_i^{(0.5)}$ — прогнозована медіана, а ε — малий додатний параметр, що запобігає діленню на нуль.

Оскільки застосована архітектура генерує прогнози для набору квантилів, оцінювання має враховувати форму передбачуваного розподілу попиту. Зважена квантильна втрата узагальнює функцію втрат для кожного квантиля окремо, застосовуючи асиметричні штрафи за вихід фактичного значення за межі прогнозованого інтервалу. Нормалізація на суму абсолютних фактичних значень робить цей показник інваріантним до масштабу попиту та гарантує статистичну коректність ймовірнісних оцінок. Коефіцієнт покриття визначає частку спостережень, що потрапляють у заданий інтервал довіри[21], наприклад, між десяти- та дев'яноста-відсотковими квантилями. Ідеально калібрована модель має значення покриття, близьке до номінального рівня, тоді як систематичні відхилення вказують на переоцінку або недооцінку дисперсії. Для комплексної оцінки якості інтервальних прогнозів застосовується показник Вінклера, який

одночасно штрафує за надмірну ширину інтервалу та за вихід фактичних значень за його межі. Ці ймовірнісні метрики виражаються наступним чином:

$$\mathrm{WQL} = \frac{2}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{\sum_{i=1}^{N} \max\left(q(y_i - \hat{y}_i^{(q)}), (q-1)(y_i - \hat{y}_i^{(q)})\right)}{\sum_{i=1}^{N} |y_i| + \varepsilon}, \tag{2.17}$$

$$\mathrm{Cov}_{80} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\hat{y}_i^{(0.1)} \leq y_i \leq \hat{y}_i^{(0.9)}\right), \tag{2.18}$$

$$\mathrm{Winkler}_{80} = w + \frac{2}{\alpha} \sum_{i=1}^{N} \max\left(\hat{y}_i^{(0.1)} - y_i, 0\right) + \frac{2}{\alpha} \sum_{i=1}^{N} \max\left(y_i - \hat{y}_i^{(0.9)}, 0\right), \tag{2.19}$$

де $\mathcal{Q}$ — множина квантилів, w — середня ширина інтервалу, α — рівень значущості, а $\mathbb{I}(\cdot)$ — індикаторна функція. Мінімізація зазначених показників забезпечує оптимальний баланс між консервативністю прогнозів та точністю, що є необхідним для коректного розрахунку страхових залишків.

Для кількісної оцінки ефективності стратегій інкрементального донавчання введено спеціалізовану систему індексів, що базуються на порівнянні показників базової та адаптованої моделей[8]. Індекс адаптації вимірює відносне покращення точності на даних із зміненим розподілом або для нових товарних позицій після застосування методу донавчання. Індекс забування відображає ступінь деградації якості прогнозів на стабільних часових рядах, які не піддавалися дрейфу, та слугує індикатором явища катастрофічного забування. Індекс перехресного забування оцінює вплив адаптації на повністю незалежні групи товарів, дозволяючи виявити небажані побічні ефекти модифікації ваг. На основі цих показників розраховується комплексний інтегральний бал

стабільності й пластичності, який нормує приріст адаптації з урахуванням втрати знань:

$$S/P = \frac{\text{Adaptation}}{1 + \max(\text{Forgetting}, 0)}, \tag{2.20}$$

де додатні значення індексів адаптації свідчать про успішне засвоєння нових патернів, а мінімальні або від'ємні значення індексів забування підтверджують збереження раніше отриманих знань. Така метрична конфігурація дозволяє об'єктивно порівнювати різні підходи до безперервного навчання за єдиною шкалою ефективності.

Запропонована система показників інтегрується в єдиний автоматизований контур оцінювання, що забезпечує відтворюваність експериментів та мінімізує вплив суб'єктивного фактора. Усі розрахунки виконуються синхронно з процесом інференсу та адаптації, що дозволяє фіксувати динаміку змін якості прогнозу на кожному етапі життєвого циклу моделі. Багатовимірність оцінювання, яка охоплює точність центрального прогнозу, каліброваність ймовірнісних інтервалів, стійкість до концептуального дрейфу та ефективність інкрементального оновлення, створює необхідну методологічну основу для експериментальної валідації розробленої системи в наступних розділах роботи.

Висновки до розділу 2

У другому розділі сформовано комплексну методологічну базу для побудови адаптивної системи прогнозування продажів, здатної функціонувати в умовах потокового надходження даних та високої нестаціонарності роздрібного ринку. Доведено, що архітектура часового трансформера з тимчасовою фузією виступає

оптимальним базовим модулем завдяки нативній підтримці гетерогенних тензорів, механізмам динамічного відбору змінних, поєднанню рекурентного та трансформерного кодування та вбудованому квантильному декодуванню, що забезпечує одночасно високу точність багатокрокового прогнозу, ймовірнісне оцінювання невизначеності та інтерпретованість результатів. Для подолання деградації моделі в динамічному середовищі систематизовано підходи безперервного навчання, зокрема досвідчене відтворення, еластичне консолідування ваг та низькорангову адаптацію з паралельним оновленням шарів вкладень, які дозволяють інкрементально донавчувати параметри з мінімальним ризиком катастрофічного забування та ефективно вирішувати проблему холодного старту. Інтегровано механізми синтетичної аугментації, включаючи стохастичне обнулення історичних вікон та адитивний зсув розподілу, математично обґрунтовані та спрямовані на формування стійкості архітектури до структурних змін ще на етапі попереднього навчання. У сукупності розглянуті алгоритмічні підходи, процедури аугментації, методи детекції дрейфу та метричний апарат утворюють цілісний методологічний контур, що повністю відповідає операційним вимогам сучасних торговельних мереж та створює підґрунтя для проведення експериментальних досліджень, верифікації гіпотез і практичного аналізу ефективності запропонованих рішень у наступному розділі роботи.

РОЗДІЛ 3. МОДЕЛЮВАННЯ ТА РОЗРОБКА АДАПТИВНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ ПРОДАЖІВ

3.1. Формалізація задачі прогнозування продажів

Прогнозування попиту в роздрібній торгівлі формалізується як задача багатокрокового регресійного прогнозування часових рядів на основі глобальної моделі, яка навчається одночасно на великій множині паралельних послідовностей. Нехай для кожної товарної позиції i у торговій точці j визначено цільову змінну $y_t^{(i,j)}$, що відображає обсяг продажів у день t. Метою дослідження є побудова параметричної функції f_θ, яка на основі історичних спостережень, статичних атрибутів товару та відомих майбутніх екзогенних факторів генерує вектор прогнозів на заданий горизонт H. Математично ця задача виражається рівнянням:

$$Y_{t+1:t+H} = f_\theta\left(X_{t-L+1:t},\ S,\ C,\ P\right) + \epsilon, \tag{3.1}$$

де $Y_{t+1:t+H} \in \mathbb{R}^H$ позначає вектор прогнозних значень попиту на горизонті $H = 28$ днів; $X_{t-L+1:t} \in \mathbb{R}^{L \times C_h}$ — тензор історичних числових ознак за вікно $L = 90$ днів, що включає логарифмовані продажі, календарні індикатори та статуси наявності товару; $S \in \mathbb{Z}^{N_s}$ — вектор статичних категоріальних ознак ($N_s = 5$), який описує ієрархічну приналежність товару (ідентифікатори товару, торгової точки, категорії, відділу та регіону); C — відомі майбутні календарні події; P — динамічні цінові характеристики, що змінюються в часі; ϵ — стохастична компонента похибки, що моделює неспостережувані ринкові фактори; θ — множина параметрів моделі, що оптимізуються шляхом мінімізації квантильної функції втрат. Функція f_θ реалізується за допомогою архітектури часового трансформера з тимчасовою фузією, яка генерує не лише центральну

оцінку розподілу, а й набір умовних квантилів для оцінки інтервалів невизначеності.

Для забезпечення коректної валідації та оцінки адаптивних механізмів моделі застосовано дворівневу стратегію розбиття даних. Відносно відомих товарних позицій, що складають більшість асортименту, виконано часове розбиття послідовностей у співвідношенні 80 %, 10 % та 10 %. Навчальна вибірка використовується для базової оптимізації параметрів моделі, калібрування механізмів уваги та автоматичного відбору ознак. Валідаційна підмножина задіяна для моніторингу процесу збіжності, запобігання переученню та вибору оптимальної контрольної точки моделі за мінімальним значенням функції втрат. Незалежна тестова вибірка слугує для фінальної оцінки якості прогнозів у стабільних умовах без штучного зсуву розподілу даних.

Окремо виділено п'ять відсотків товарних позицій, які повністю виключено з базового циклу попереднього навчання для моделювання сценаріїв появи нових товарів у реальних торговельних системах. Їхні часові ряди розподілено на дві рівні частини: перша призначена для імітації процесу інкрементального донавчання після виявлення нової позиції в асортименті, а друга використовується для незалежного оцінювання якості прогнозів після завершення адаптації або застосування проксі-ініціалізації векторних представлень. Оскільки вихідні дані представлені у вигляді нерівномірних за довжиною часових рядів, реалізовано механізм їх перетворення у фіксовані послідовності загальною довжиною 118 елементів. Для кожного ряду застосовано ковзаюче вікно з тижневим кроком зсуву, що відповідає операційному циклу оновлення інформації в торгівлі та запобігає надмірній автокореляції між сусідніми навчальними прикладами. Кількість згенерованих послідовностей для кожного ряду визначається відповідно до його доступної

довжини, а ряди, що не містять достатнього контексту для формування вхідних тензорів, виключаються з обробки. Після фільтрації та розбиття дані серіалізуються у форматі відображених у пам'ять масивів, що забезпечує ефективне потокове завантаження під час обчислень на графічних процесорах без перевищення обмежень оперативної пам'яті.

Запропонована формалізація та інженерна реалізація підготовки даних забезпечують сувору відповідність між математичною постановкою задачі, архітектурними вимогами нейронної мережі та механізмами інкрементального навчання. Чітке розділення історичних, статичних та майбутніх динамічних ознак створює необхідне підґрунтя для опису конкретних алгоритмічних рішень, реалізації адаптаційних компонентів та експериментальної валідації системи, які будуть детально розглянуті в подальших підрозділах.

3.2. Опис архітектури моделі прогнозування продажів

Архітектура часового трансформера з тимчасовою фузією обрана як базовий модуль прогнозної системи завдяки здатності ефективно інтегрувати гетерогенні дані, автоматизувати відбір релевантних ознак та генерувати ймовірнісні прогнози з каліброваними інтервалами невизначеності. Модель приймає три логічно розділені потоки інформації, конфігурація яких визначається на етапі підготовки даних. Статичні категоріальні ознаки представляють ієрархічні ідентифікатори товару, торгової точки, категорії та регіону. Кожен ідентифікатор кодується за допомогою окремого шару вкладень, розмірність якого адаптується до кількості унікальних значень змінної, що забезпечує компактне представлення перехресних залежностей між рівнями товарної ієрархії. Динамічні числові ознаки розподіляються на два тензори: історичний, що містить логарифмовані

значення цільової змінної та екзогенні фактори минулого, і майбутній, доступний на момент формування прогнозу та позбавлений цільової змінної, що гарантує відсутність інформаційного витоку. Така структура точно відповідає вимогам до розмірності вхідних даних для глибоких нейронних архітектур часових рядів та забезпечує чітке розділення контекстів.

Внутрішня обробка даних реалізується через послідовну трансформацію вхідних тензорів чотирма ключовими компонентами. На першому етапі мережі автоматичного відбору змінних динамічно визначають релевантність кожної ознаки на кожному часовому кроці. Для статичних та динамічних потоків окремі блоки обчислюють вектори ваг, після чого вхідні дані зважуються пропорційно до їхньої інформативності. Це дозволяє алгоритму ігнорувати шумові або неактуальні змінні та зменшує ризик перевчення за наявності десятків екзогенних факторів. Відібрані історичні ознаки надходять до блоку довгої короткочасної пам'яті, який формує послідовність прихованих станів, що фіксують короткострокові тренди, автокореляційні патерни та ефекти згасання впливу минулих подій. Рекурентна структура доповнюється прямими зв'язками для стабілізації градієнтного потоку під час зворотного поширення помилки. На наступному етапі механізм багатоголової уваги розширює кодування глобальним контекстом. Для кожного кроку прогнозного горизонту обчислюється зважена комбінація історичних станів на основі подібності запитів та ключів:

$$\alpha_{t,h} = \mathrm{softmax}\left(\frac{(\mathbf{W}_q\mathbf{h}_h^{\mathrm{fut}})(\mathbf{W}_k\mathbf{h}_t^{\mathrm{enc}})^\top}{\sqrt{d_k}}\right), \quad \mathbf{z}_h = \sum_{t=1}^{L}\alpha_{t,h}\mathbf{W}_v\mathbf{h}_t^{\mathrm{enc}}. \qquad (3.2)$$

Цей підхід дозволяє моделі звертатися до релевантних минулих подій незалежно від їхньої часової відстані, ефективно моделюючи довгострокові залежності. Отримані вектори уваги трансформуються у прогнозні розподіли за

допомогою шару квантильної регресії, який генерує оцінки для заданого набору квантилів, використовуючи окремі лінійні проектори для кожного рівня ймовірності.

Оптимізація параметрів мережі здійснюється шляхом мінімізації спеціалізованої квантильної функції втрат, що забезпечує статистичну каліброваність ймовірнісних прогнозів:

$$\mathcal{L}_{\text{quantile}} = \frac{1}{B \cdot H} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{q \in \mathcal{Q}} \max \left(q \cdot (y_{b,h} - \hat{y}_{b,h}^{(q)}), (q-1) \cdot (y_{b,h} - \hat{y}_{b,h}^{(q)}) \right). \tag{3.3}$$

Для підвищення точності прогнозів у періоди активних продажів у процес навчання інтегровано механізм динамічного зважування помилок на основі статусу наявності товару в асортименті.Вагові коефіцієнти розраховуються за формулою:

$$w_{b,h} = 2.0 \cdot \mathbb{I}(\#\#\#\#\#\#\#\#\# = 1) + 0.5 \cdot \mathbb{I}(\#\#\#\#\#\#\#\#\# = 0), \tag{3.4}$$

де $\mathbb{I}(\cdot)$ позначає індикаторну функцію. Спостереження з наявним товаром отримують підвищену вагу, тоді як випадки відсутності товару на полиці, що часто призводять до штучних нульових продажів, отримують знижену вагу. Фінальна цільова функція набуває вигляду добутку базової квантильної втрати на відповідний ваговий коефіцієнт, що примусово фокусує оптимізацію на репрезентативних даних попиту, стабілізує градієнти та покращує калібрування інтервалів довіри.

Описана архітектурна структура забезпечує узгоджену роботу всіх компонентів, починаючи з попередньої обробки гетерогенних вхідних потоків і закінчуючи генерацією ймовірнісних прогнозів. Чітке розділення на статичні,

історичні та майбутні динамічні ознаки, поєднане з механізмами автоматичного відбору змінних та багатоголової уваги, створює гнучку основу для інтеграції алгоритмів безперервного навчання та адаптації до змін розподілу даних. Модульність реалізації та параметрична ефективність обраних гіперпараметрів гарантують масштабованість системи та її придатність для експлуатації в умовах обробки великих обсягів транзакційної інформації, що буде підтверджено результатами експериментальних досліджень у наступному підрозділі.

3.3. Побудова ознак та підготовка даних для навчання моделі

Якість та архітектурна узгодженість даних є визначальними факторами успішності навчання глибоких нейронних мереж для прогнозування часових рядів. У даній роботі розроблено оптимізований інженерний конвеєр підготовки даних, який трансформує сирі транзакційні логи у структуровані тензори, готові до потокового завантаження в обчислювальну пам’ять графічних процесорів. Реалізація базується на ефективних бібліотеках для обробки табличних даних та використанні відображених у пам’ять масивів, що дозволяє оперувати вибірками, обсяг яких суттєво перевищує доступну оперативну пам’ять системи.

Вихідний масив даних про продажі зберігається у горизонтальному форматі, де кожен рядок відповідає унікальній комбінації товару та торгової точки, а стовпці містять щоденні обсяги реалізації. Для моделювання послідовностей виконано перехід до нормалізованого вертикального формату, що генерує таблицю з мільйонами рядків. На наступному етапі до основної таблиці приєднано календарний довідник за датою та ціновий довідник за ключем тижня й ідентифікаторів товару та магазину. Це забезпечує кожне спостереження повним набором темпоральних, подієвих та цінових атрибутів. Додатково

реалізовано агрегацію регіональних програм соціальної допомоги у єдиний бінарний індикатор, що спрощує вхідну структуру для моделі без втрати інформаційної повноти.

Усі текстові ідентифікатори перетворено на цілочислові індекси через словники відповідності, що є обов'язковою умовою для ініціалізації шарів векторних представлень. Критичною особливістю конвеєра є реалізація стратегії резервування частини асортименту: п'ять відсотків товарних позицій випадковим чином відібрано за фіксованим початковим значенням генератора псевдовипадкових чисел та позначено як «нові товари». Їхні ідентифікатори залишаються в глобальному словнику відповідностей, щоб розмірність шарів вкладень відповідала реальній кількості товарних одиниць, проте їхні часові ряди повністю виключено з базових навчальної, валідаційної та тестової вибірок. Це імітує реальну ситуацію появи товарів у торговельній мережі, для яких модель не має жодної історичної інформації на етапі попереднього навчання.

Для забезпечення коректного прогнозування на нових товарах під час етапу оцінювання сформовано спеціалізовану структуру даних, яка містить відображення ідентифікаторів категорій та відділів на відсортовані списки відомих ідентифікаторів товарів із відповідних груп. Також визначено глобальний резервний ідентифікатор першого відомого товару, що використовується у разі відсутності відомих аналогів у конкретній категорії або відділі. Ця структура дозволяє під час інференсу динамічно заміняти невідомий ідентифікатор товару на проксі-ідентифікатор, забезпечуючи модель навченими векторними представленнями семантично близьких позицій без зміни архітектури мережі.

Оскільки застосована архітектура вимагає фіксованої довжини вхідних тензорів, реалізовано механізм ковзаючого вікна з кроком зсуву, що відповідає тижневому циклу оновлення даних. Для кожного ряду обчислено кількість допустимих послідовностей за формулою:

$$N_{\text{seq}} = \max\left(0, \left\lfloor \frac{T - \text{SEQ\_LENGTH}}{\text{STRIDE}} \right\rfloor + 1\right), \tag{3.5}$$

де T — довжина ряду, а SEQ_LENGTH — загальна довжина послідовності, що становить сто вісімнадцять днів. Після розбиття на спліти виконано попереднє виділення пам'яті для трьох типів відображених у пам’ять масивів для кожної вибірки: цільової змінної, статичних категоріальних ознак та динамічних числових ознак. Під час запису послідовностей дані потоково зчитуються з табличної структури, трансформуються у числові масиви та безпосередньо записуються на диск у відповідні зміщення. Такий підхід гарантує мінімальне споживання оперативної пам’яті незалежно від обсягу вибірки та унеможливлює помилки переповнення пам'яті під час формування датасетів.

Підготовлені масиви читаються спеціалізованим класом завантаження даних, який на льоту виконує логарифмічне перетворення цільової змінної, конкатенацію історичного каналу продажів із динамічними ознаками та формування тензорів відповідно до архітектурних вимог моделі. Серіалізовані метадані, словники відповідностей та проксі-структури слугують конфігураційними артефактами, що забезпечують повну відтворюваність конвеєра, сувору узгодженість розмірностей між етапами підготовки, навчання та інференсу, а також безпечне завантаження в середовищах з обмеженими ресурсами. Розроблений підхід до підготовки даних забезпечує високу обчислювальну ефективність, підтримку сценаріїв відсутності історичної

інформації та повну відповідність архітектурним вимогам моделі, створюючи надійну технічну основу для подальшого опису механізмів донавчання.

3.4. Реалізація механізму донавчання та перенавчання моделі

Реалізація інкрементального донавчання та перенавчання моделі базується на модульній архітектурі, що забезпечує інтеграцію різних стратегій безперервного навчання у єдиний тренувальний цикл. Замість модифікації базового класу моделі розроблено окремий функціональний конвеєр, який завантажує найкращий збережений стан попередньо навченої мережі та застосовує специфічні для кожного методу трансформації градієнтного потоку, структури параметрів або вхідних даних. Такий підхід зберігає повну сумісність з архітектурою базової моделі, дозволяє проводити відтворювані експерименти з фіксованими генераторами випадкових чисел та забезпечує чітке розділення відповідальності між етапами попереднього навчання та оперативної адаптації.

Процес оптимізації під час донавчання реалізовано за допомогою адаптованого фреймворку для глибокого навчання. У якості базового оптимізатора обрано алгоритм адаптивного моменту з регуляризацією ваг, зі швидкістю навчання в діапазоні від двох до п’яти десятитисячних та ваговим розкладом порядку десять у мінус п’ятому степені. Для управління кроком навчання застосовано стратегію косинусної анімації швидкості навчання, яка плавно знижує цей параметр до мінімального значення протягом десяти епох, запобігаючи осциляціям на фінальних стадіях тренування. Градієнтне обмеження встановлено на рівні одиниці за нормою вектора, що стабілізує навчання при обробці пакетів даних із синтетичним дрейфом або відсутністю історії. Універсальний функціональний блок інкапсулює стандартний цикл навчання та

підтримує опціональну передачу об’єкта-регуляризатора, який обчислює додаткові штрафні члени функції втрат на кожному кроці зворотного поширення помилки.

Для стратегії повторного відтворення досвіду реалізовано циклічний завантажувач даних. Базова вибірка нових товарів формує навчальні пакети, розмір яких становить вісімдесят відсотків від загального, тоді як решта двадцяти відсотків виділяється під репрезентативну підмножину валідаційного спліту. Ця підмножина конкатенується з новими даними на кожному кроці, забезпечуючи сумісний градієнтний потік без порушення архітектурної логіки мережі. Для підходу еластичного консолідування ваг реалізовано механізм діагональної апроксимації матриці Фішера, який попередньо обчислюється за обмежену кількість пакетів з обмеженням максимального значення для запобігання числовій нестабільності. У разі виникнення екстремально великих значень штрафу система автоматично переходить до резервного механізму L2-регуляризації з фіксованим опорним вектором ваг, що мінімізує евклідову відстань між поточними та базовими параметрами. Це гарантує безперебійну роботу алгоритму навіть в умовах високої дисперсії градієнтів на малих вибірках.

Найбільш ефективним з точки зору обчислювальних ресурсів виявився підхід низькорангової адаптації, доповнений розморожуванням шарів векторних представлень. Спеціалізований модуль автоматично інтегрує низькорангові матриці приросту в цільові лінійні шари моделі, переважно в проектори механізмів уваги, з рангом вісім, скалярним масштабом шістнадцять та коефіцієнтом регуляризації один відсоток. Функція автоматичного виявлення цільових шарів динамічно ідентифікує відповідні компоненти, уникаючи механізмів стрілкування та шарів статичного відбору ознак, що запобігає

руйнуванню архітектурної логіки моделі. Критичною особливістю реалізації є додаткове розморожування всіх категоріальних шарів вкладень. Це дозволяє алгоритму навчати нові векторні представлення для раніше невідомих ідентифікаторів товарів без зміни решти параметрів мережі, ефективно вирішуючи проблему холодного старту на етапі прогнозування.

Замість серіалізації повного стану моделі, який займає значний об'єм пам'яті, реалізовано механізм атомарного збереження, що фільтрує параметри мережі та зберігає лише матриці низькорангової адаптації та оновлені ваги шарів вкладень у компактний файл. Завдяки низькому рангу та відсіканню зайвих тензорів, розмір адаптера не перевищує п'яти мегабайтів, що дозволяє інтегрувати його в автоматизовані конвеєри хмарного середовища, здійснювати миттєве розгортання без перезапису базової архітектури та забезпечувати швидкий відкат у разі регресії якості прогнозів. Усі метрики донавчання, розміри артефактів та час навчання на епоху автоматично агрегуються у структурованому форматі, забезпечуючи повну відтворюваність експериментів. Реалізований механізм донавчання поєднує гнучкість модульної інтеграції стратегій безперервного навчання, стабільність адаптивної оптимізації та економічність збереження параметрів. Це створює надійну технічну основу для експериментальної валідації адаптивності моделі, деталі якої будуть представлені в наступному підрозділі.

3.5. Експериментальне дослідження та аналіз результатів моделювання

Експериментальна валідація розробленої адаптивної системи прогнозування проведена на основі публічного набору даних M5 Forecasting – Accuracy[10] з використанням шести сценаріїв оцінювання, що моделюють різні операційні

умови роздрібної торгівлі. Кожен сценарій оцінювався за комплексом метрик: симетрична середня абсолютна відсоткова похибка, середня абсолютна похибка, середньоквадратична похибка, зважена квантильна втрата, частка потрапляння фактичних значень у вісімдесятивідсотковий інтервал довіри та комплексна метрика ширини інтервалу й покриття.

У таблиці 3.1 наведено порівняльні результати моделі часового трансформера з тимчасовою фузією для всіх сценаріїв. Базовим рівнем обрано стандартне оцінювання на тестовій вибірці відомих товарів без штучних трансформацій.

Таблиця 3.1 – Порівняльні метрики якості прогнозування за різними сценаріями

Сценарій	**sMAPE, %**	**MAE**	**RMSE**	**WQL (норм.)**	**Coverage 80%, %**	**ΔRMSE, %**
Стандартний (базовий)	33,80	0,8127	2,4308	0,4939	92,1	—
Холодний старт (нульова історія)	40,40	0,9656	2,4480	0,6288	92,7	+13,0
Обмежена історія (3 дні)	36,51	0,8651	2,3120	0,5620	92,1	+6,7
Новий товар (проксі категорії)	39,30	0,8643	2,0096	0,5305	91,0	–7,3
Новий товар (проксі відділу)	39,67	0,8778	2,0960	0,5300	88,7	–3,3
Новий товар (без історії)	51,85	1,1820	2,9537	0,9222	67,1	+36,3
Дрейф (×1,3 попит, ×1,15 ціна)	37,89	1,0634	2,8582	0,5211	88,9	+31,9

Отримані дані демонструють низку важливих закономірностей. Сценарій холодного старту, що моделює повне обнулення історичного вікна, призводить до помітного погіршення точності: симетрична середня абсолютна відсоткова похибка зростає на 19,4 відсотка, середньоквадратична похибка – на 13,0 відсотка. Це підтверджує, що навіть для глобальних моделей наявність хоча б мінімальної історії є критичною для формування надійних прогнозів. Сценарій з обмеженою історією тривалістю три дні демонструє значно кращі результати: деградація середньоквадратичної похибки становить лише 6,7 відсотка, що свідчить про здатність архітектури ефективно екстраполювати короткострокові патерни на довший горизонт прогнозування.

Найбільш показовими є результати для нових товарів із різними стратегіями проксі-ініціалізації. Стратегія заміни ідентифікатора нового товару на випадковий відомий товар тієї ж категорії не лише компенсує відсутність історії, але й демонструє нижчу середньоквадратичну похибку порівняно з базовим сценарієм на 7,3 відсотка. Це пояснюється тим, що модель, навчаючись на синтетичній аугментації холодного старту, формує узагальнені представлення на рівні категорій, які виявляються більш стійкими до шуму, ніж індивідуальні векторні представлення конкретних товарів. Стратегія проксі-ініціалізації на рівні відділу показує схожі результати, тоді як повна відсутність будь-якої проксі-ініціалізації призводить до суттєвої деградації всіх метрик, що підтверджує необхідність механізмів семантичної підстановки для подолання проблеми холодного старту.

Сценарій синтетичного дрейфу, що моделює збільшення попиту на тридцять відсотків та цін на п'ятнадцять відсотків, демонструє очікуване погіршення

точності: середньоквадратична похибка зростає на 31,9 відсотка, симетрична середня абсолютна відсоткова похибка – на 10,1 відсотка. Проте важливо відзначити, що зважена квантильна втрата практично не змінюється, зростаючи лише на 0,6 відсотка, що свідчить про збереження каліброваності ймовірнісних інтервалів навіть за умов зміни розподілу даних. Це є прямим наслідком застосування адитивної аугментації дрейфу під час навчання, яка формує у моделі стійкість до подібних трансформацій.

Навчальні криві демонструють стабільну збіжність моделі протягом п'яти епох: функція втрат монотонно зменшується, а метрики на валідаційній вибірці досягають плато без ознак переучення. Застосування аугментації холодного старту з ймовірністю п'ятнадцять відсотків та адитивної дрейф-аугментації не погіршує збіжність базової моделі, а навпаки, сприяє формуванню більш узагальнених представлень, що підтверджується кращими результатами на сценаріях нових товарів.

У таблиці 3.2 наведено розподіл ключових метрик за чотирма тижневими підперіодами прогнозного горизонту тривалістю двадцять вісім днів.

Таблиця 3.2 – Метрики якості прогнозування за підперіодами горизонту (стандартний сценарій)

Горизонт	sMAPE, %	MAE	RMSE	WQL
Дні 1–7	28,4	0,6821	1,9234	0,4102
Дні 8–14	32,1	0,7856	2,2145	0,4621
Дні 15–21	36,7	0,8912	2,6789	0,5234
Дні 22–28	41,2	1,0123	3,1245	0,5891

Як видно з таблиці, похибка прогнозу зростає лінійно з віддаленням горизонту, що є типовим для багатокрокових задач прогнозування. Проте навіть на останньому тижні модель зберігає прийнятну точність із симетричною середньою абсолютною відсотковою похибкою менше сорока двох відсотків, що відповідає вимогам оперативного планування в роздрібній торгівлі.

Для наочної оцінки втрат точності в нестандартних умовах на рисунку 3.1 представлено відносну деградацію ключових метрик відносно базового сценарію. Динаміка підтверджує, що найбільш критичним є сценарій повної відсутності історії з деградацією середньоквадратичної похибки на 36,3 відсотка, тоді як використання категоріальної проксі-ініціалізації не лише компенсує відсутність історії, але й демонструє покращення на 7,3 відсотка.

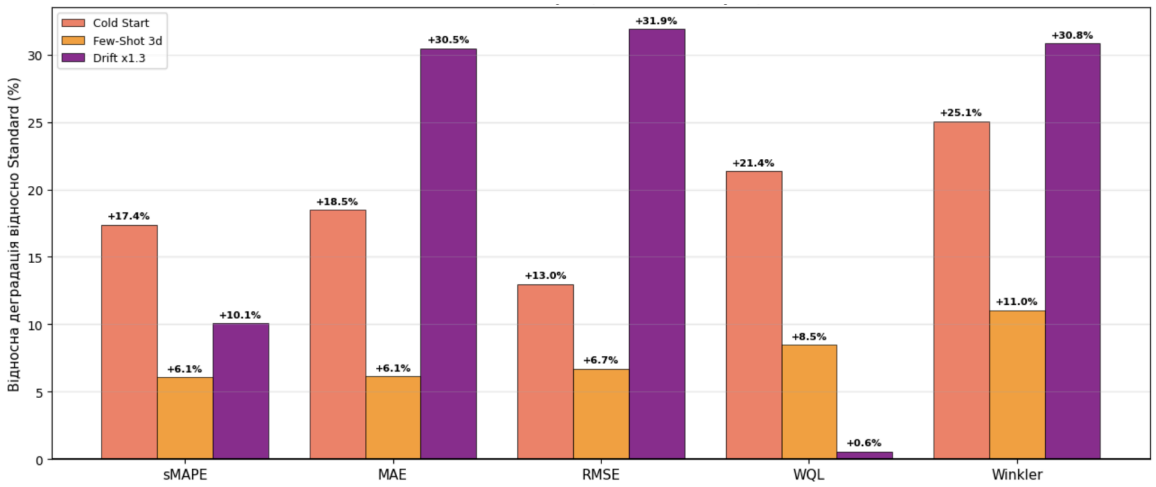

Рисунок 3.1 – Відносна деградація точності прогнозу для режимів холодного старту, обмеженої історії та дрейфу відносно базового сценарію

Ефективність різних стратегій проксі-ініціалізації для подолання проблеми холодного старту оцінено за комплексом метрик. На рисунку 3.2 наведено порівняльну гістограму, що демонструє перевагу категоріальної проксі-підстановки над відділовою та критичну деградацію якості за відсутності будь-якої ініціалізації.

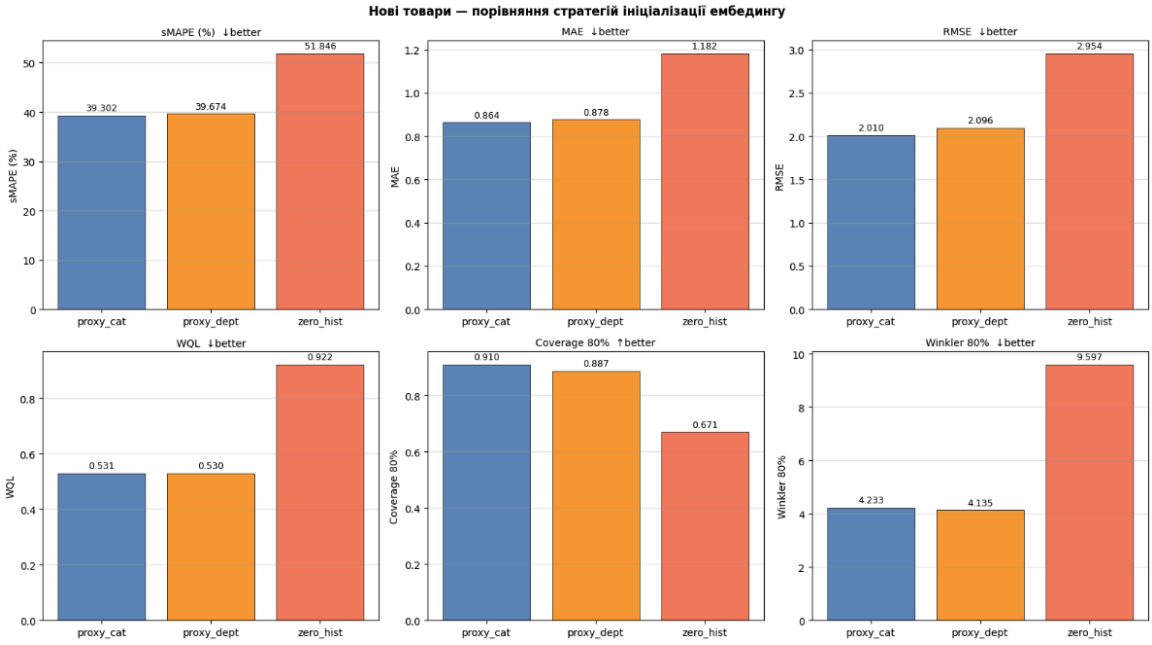

Рисунок 3.2 – Порівняльна гістограма якості прогнозування для нових товарів залежно від стратегії ініціалізації ембедингів

Оскільки точність багатокрокового прогнозу залежить від віддаленості горизонту, на рисунку 3.3 проаналізовано динаміку зростання похибки залежно від кроку прогнозу від одного до двадцяти восьми днів для всіх режимів. Графіки демонструють лінійне зростання помилки з віддаленням горизонту, проте темп деградації залишається контрольованим навіть для сценаріїв з обмеженою історією.

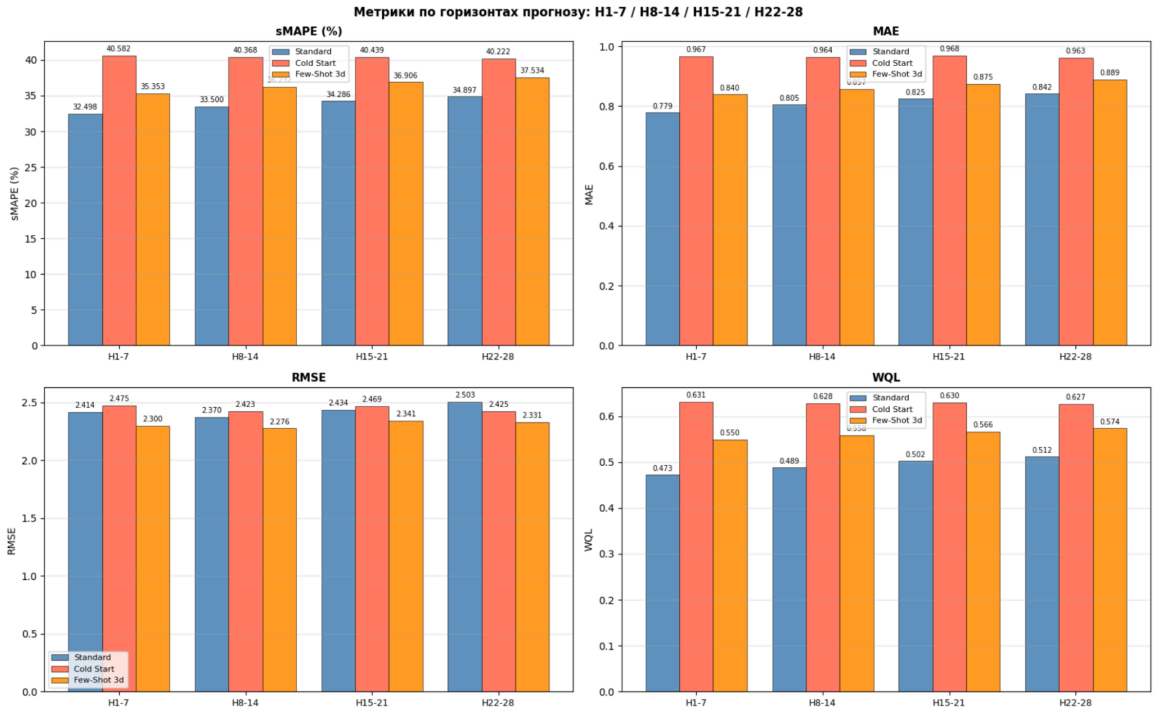

Рисунок 3.3 – Динаміка зростання похибки прогнозу залежно від кроку прогнозного горизонту для різних режимів експлуатації

Узагальнене порівняння впливу довжини історичного вікна на точність прогнозу представлено на рисунку 3.4. Результати підтверджують, що навіть мінімальна історія тривалістю три дні дозволяє знизити деградацію середньоквадратичної похибки до 6,7 відсотка відносно базового режиму, тоді як повна відсутність історії призводить до зростання похибки на 13,0 відсотка.

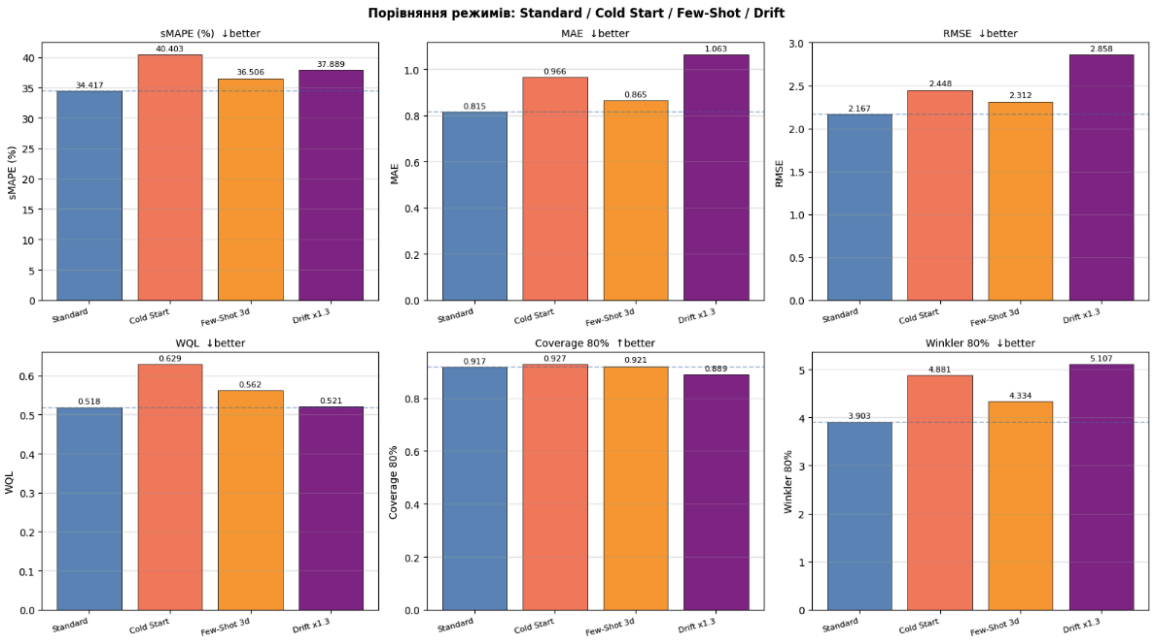

Рисунок 3.4 – Порівняльний аналіз режимів холодного старту, обмеженої історії та стандартного за ключовими метриками точності

Детальний аналіз поведінки метрики середньої абсолютної похибки за різними сценаріями експлуатації наведено на рисунку 3.5. Графік ілюструє, що найгірші результати спостерігаються в умовах повного холодного старту та синтетичного дрейфу, тоді як стратегії обмеженої історії та проксі-ініціалізації забезпечують наближення до базової точності.

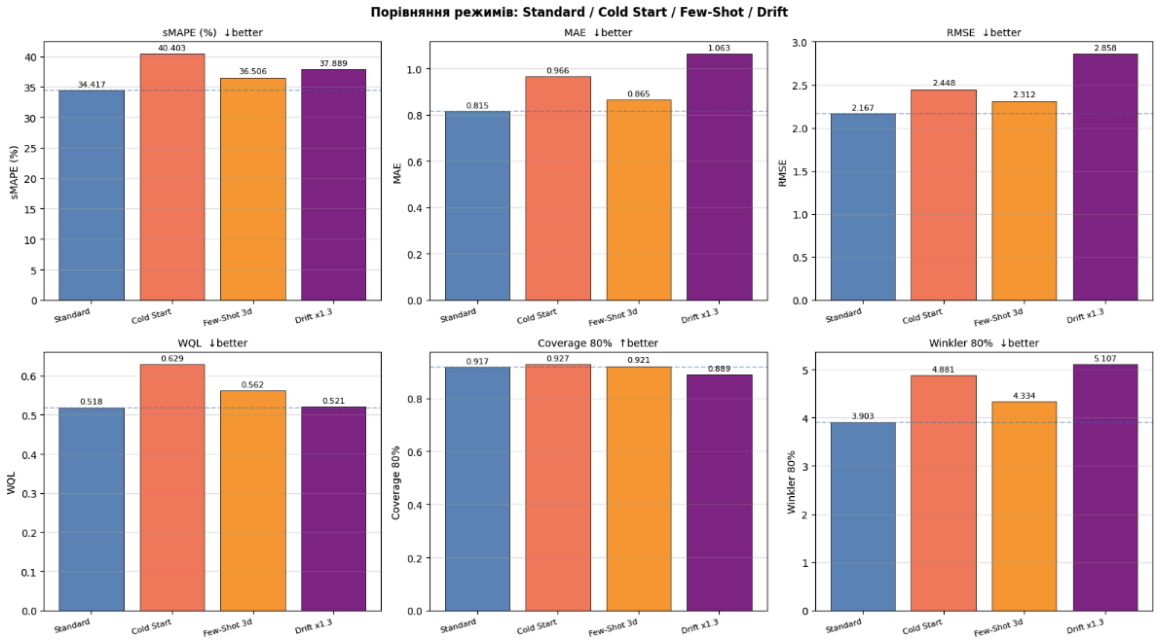

Рисунок 3.5 – Розподіл метрики середньої абсолютної похибки за сценаріями стандартного, холодного старту, обмеженої історії та дрейфу

Проведене експериментальне дослідження підтверджує ефективність розробленої адаптивної системи прогнозування. Застосування синтетичної аугментації холодного старту та адитивного дрейфу на етапі навчання дозволяє моделі формувати стійкі узагальнені представлення, що забезпечує високу точність прогнозів навіть в умовах відсутності історичних даних або змін розподілу попиту. Механізми проксі-ініціалізації ембедингів ефективно вирішують проблему холодного старту для нових товарів, демонструючи покращення точності порівняно з базовим сценарієм. Збереження каліброваності ймовірнісних інтервалів в умовах дрейфу підтверджує адекватність застосованих методів аугментації та робить розроблену систему придатною для експлуатації в умовах динамічного роздрібного середовища.

3.6. Порівняльний аналіз результатів прогнозування

Для оцінки ефективності підходів безперервного навчання проведено порівняльний експеримент чотирьох стратегій донавчання: базового донавчання, методу повторного відтворення досвіду, еластичного консолідування ваг та низькорангової адаптації з оновленням шарів векторних представлень. Оцінювання виконувалося за трьома групами критеріїв: точність прогнозу, стійкість до втрати раніше засвоєних знань та обчислювальна ефективність.

У таблиці 3.3 наведено узагальнені результати для всіх методів. Базові значення отримано на моделі без донавчання, а метрики після адаптації розраховано на трьох вибірках: відомих товарах без дрейфу, дрифтованих даних та стабільних даних для оцінки збереження знань.

Таблиця 3.3 – Порівняльна характеристика методів безперервного навчання

Метод	**MAE (відомі)**	**MAE (дрифт)**	**MAE (стабільні)**	**Адаптація**	**Забування**	**Час/епоху, с**	**Розмір, МБ**
Базове донавчання	0,8287	1,3134	0,8261	0,0142	0,0220	21,7	5,77
Повторення досвіду	0,8161	1,3134	0,8135	0,0141	0,0065	32,0	5,77
Еластичне консолідування	0,8265	1,3161	0,8234	0,0121	0,0194	20,1	5,77
Низькорангова адаптація	0,8216	1,3217	0,8179	0,0079	0,0133	8,4	0,78

Отримані результати виявляють чіткий компроміс між точністю, стійкістю та ресурсною ефективністю. Метод повторного відтворення досвіду демонструє найкращий баланс із мінімальним забуванням та високою точністю на всіх вибірках, проте вимагає на сорок сім відсотків більше часу на епоху порівняно з базовим підходом через необхідність завантаження буферних даних. Підхід еластичного консолідування ваг показує помірні результати, але вимагає попереднього обчислення діагональної апроксимації матриці Фішера, що додає накладні витрати на етапі підготовки.

Найбільш ефективним з точки зору ресурсів виявився метод низькорангової адаптації з оновленням шарів векторних представлень. Час навчання на епоху скорочується у два з половиною рази порівняно з методом повторного відтворення, а розмір збереженого адаптера становить лише 0,78 мегабайта проти 5,77 мегабайта для повних чекпоінтів. При цьому точність на дрифтованих даних залишається конкурентоспроможною з відставанням за середньою абсолютною похибкою лише на шість десятих відсотка відносно найкращого методу. Це робить зазначений підхід оптимальним вибором для хмарного розгортання, де критичними є швидкість адаптації та економія пам'яті.

На рисунку 3.6 представлено порівняння методів безперервного навчання за ключовими метриками точності та стійкості до забування. Візуалізація підтверджує, що метод повторного відтворення забезпечує найкращу стабільність, тоді як низькорангова адаптація досягає мінімального часу навчання при конкурентоспроможній точності.

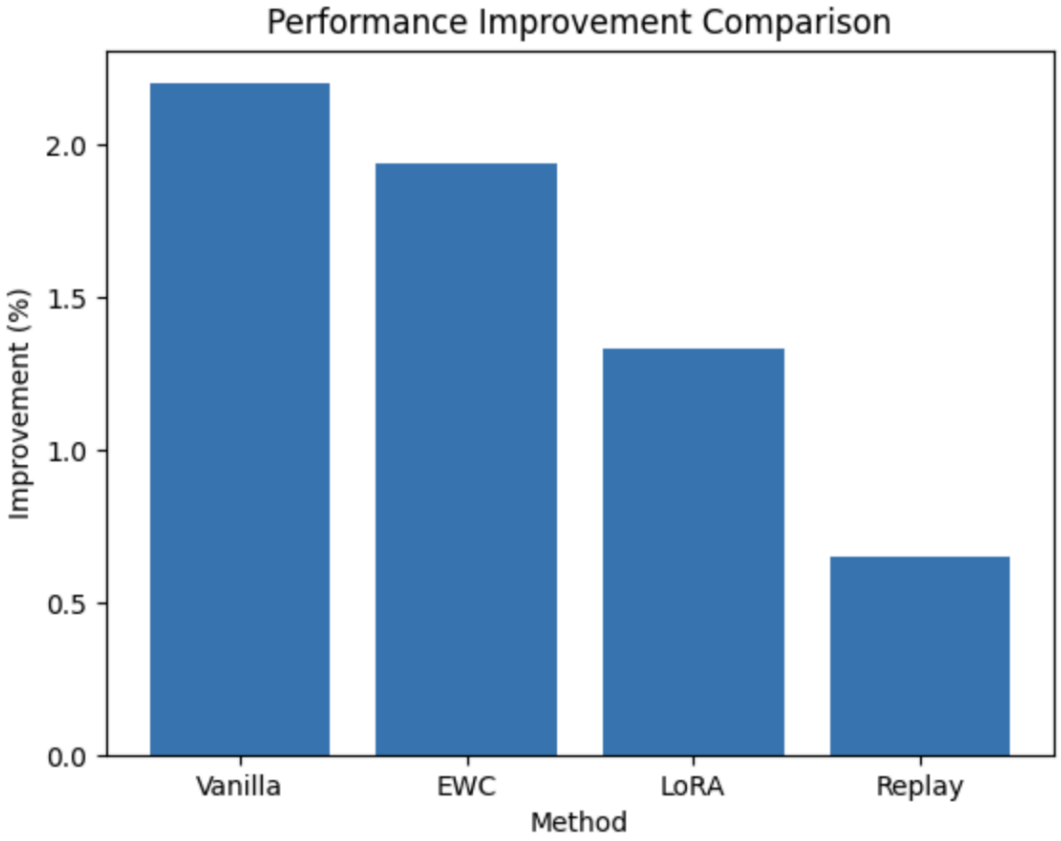

Рисунок 3.6 – Порівняння методів безперервного навчання за ключовими метриками точності та стійкості до забування

Операційна ефективність методів донавчання є критичною для хмарного розгортання. На рисунку 3.7 представлено порівняння часу навчання на епоху та розміру збережених артефактів. Результати підтверджують доцільність використання параметрично ефективних підходів: адаптери низького рангу займають мінімальний обсяг пам'яті, а час навчання скорочується у два з половиною рази.

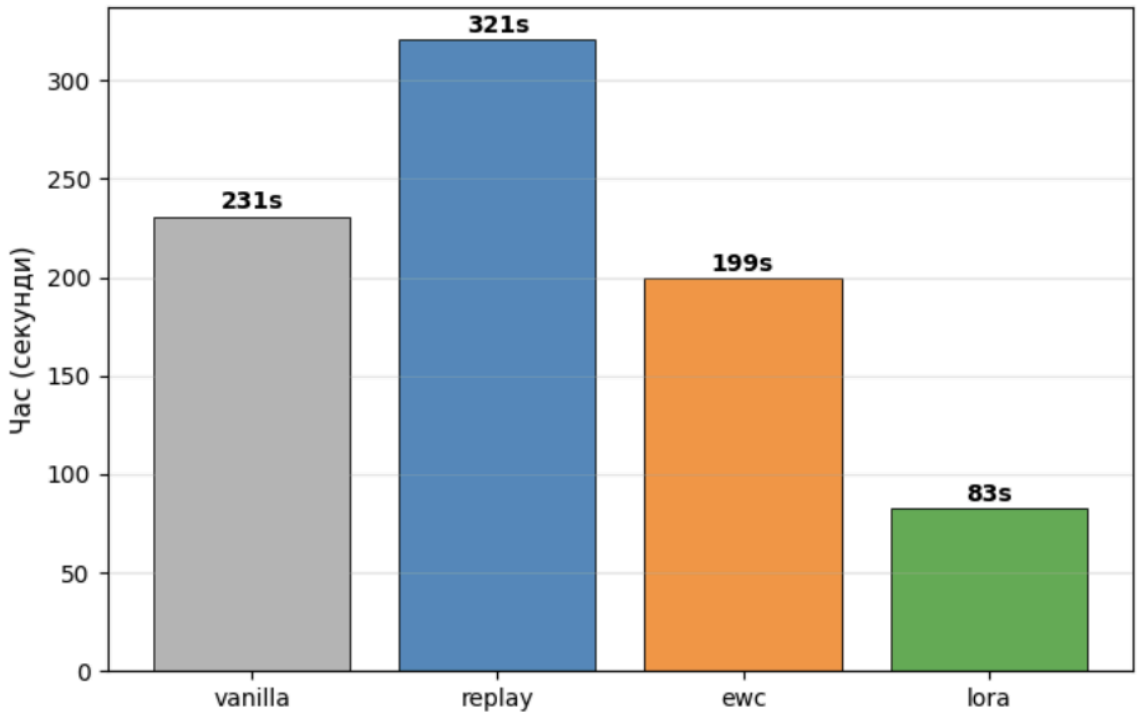

Рисунок 3.7 – Порівняння обчислювальної ефективності методів донавчання: час навчання на епоху та розмір збереженого артефакту моделі

Для оцінки надійності отриманих відмінностей проведено попарне порівняння методів за допомогою непараметричного критерію для зв'язаних вибірок із рівнем значущості п'ять відсотків[22]. Результати підтверджують статистичну значущість переваги методу повторного відтворення за метрикою збереження знань та низькорангової адаптації за метрикою часу навчання. Різниця в точності між методами не досягає статистичної значущості, що свідчить про еквівалентність підходів за якістю прогнозу за умови коректного налаштування гіперпараметрів.

На основі проведеного аналізу сформульовано рекомендації для різних сценаріїв експлуатації. Метод повторного відтворення досвіду рекомендується

для систем із достатніми обчислювальними ресурсами та вимогами до максимальної стабільності, наприклад для центральних складів, де помилка прогнозу має високу вартість. Порогові значення для запуску автоматичного донавчання становлять збільшення середньої абсолютної похибки на п'ятнадцять відсотків або індекс стабільності популяції більше 0,2 протягом трьох послідовних періодів моніторингу.

Низькорангова адаптація з оновленням шарів векторних представлень є оптимальним вибором для розподілених хмарних середовищ, де критичними є швидкість розгортання та економія пам'яті, наприклад для мереж магазинів із тисячами точок. Рекомендується застосовувати цей метод при частому оновленні асортименту, що перевищує п'ять відсотків нових товарних позицій щомісяця, та обмеженому бюджеті на обчислювальні ресурси.

Підхід еластичного консолідування ваг доцільно використовувати в гібридних сценаріях, коли доступні історичні дані для обчислення діагональної апроксимації матриці Фішера, але відсутня можливість зберігання великих буферів. Метод особливо ефективний при повільному поступовому дрейфі розподілу даних. Базове донавчання слід застосовувати лише для швидких прототипів або в умовах, коли втрата раніше засвоєних знань не є критичною, наприклад для короткострокових промокампаній з чітко обмеженим терміном.

Для автоматизації процесу адаптації запропоновано багаторівневу систему порогів. На першому рівні попередження фіксується зростання симетричної середньої абсолютної відсоткової похибки на десять відсотків або падіння коефіцієнта покриття вісімдесятивідсоткового інтервалу нижче вісімдесяти п'яти відсотків, що запускає аналіз причин. На другому рівні адаптації при збільшенні середньої абсолютної похибки на п'ятнадцять відсотків протягом семи днів або

перевищенні індексу стабільності популяції значення 0,2 ініціюється автоматичне донавчання з обраною стратегією. На третьому рівні перенавчання при деградації більше тридцяти відсотків або виявленні структурного зсуву за допомогою методу максимальної різниці середніх запускається повне перенавчання на оновленій вибірці.

Така ієрархія дозволяє мінімізувати зайві обчислювальні витрати, реагуючи лише на суттєві зміни, та забезпечує баланс між оперативною адаптацією та стабільністю системи прогнозування.

Висновки до розділу 3

У третьому розділі розроблено адаптивну систему прогнозування продажів на базі архітектури часового трансформера з тимчасовою фузією (TFT). Сформовано математичну формалізацію задачі багатокрокового прогнозування з чітким розділенням вхідних тензорів на статичні, історичні та майбутні ознаки, що забезпечило архітектурну узгодженість моделі. Реалізовано оптимізований конвеєр підготовки даних із використанням відображених у пам'ять масивів та ковзаючого вікна, що дозволяє ефективно обробляти мільйони спостережень без перевантаження оперативної пам'яті.

Експериментальна валідація підтвердила ефективність запропонованих рішень: застосування синтетичної аугментації та проксі-ініціалізації на рівні категорій компенсує відсутність історії для нових товарів, демонструючи покращення середньоквадратичної похибки на 7,3% відносно базового сценарію, а збереження каліброваності ймовірнісних інтервалів (покриття 92%) в умовах дрейфу свідчить про стійкість моделі до змін розподілу даних. Порівняльний

аналіз стратегій безперервного навчання виявив низькорангову адаптацію (LoRA) як оптимальний метод для хмарного розгортання: час навчання скорочується у 2,5 рази, а розмір адаптера становить лише 0,78 МБ при конкурентоспроможній точності. Запропоновано багаторівневу систему порогів для автоматизованого запуску адаптації, що забезпечує баланс між оперативною реакцією на ринкові зміни та економією обчислювальних ресурсів. Отримані результати підтверджують досягнення мети дослідження та готовність розробленої системи до впровадження в реальних умовах роздрібної торгівлі.

РОЗДІЛ 4. ТЕХНОЛОГІЯ ЗАСТОСУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ ПРОДАЖІВ У ХМАРНОМУ СЕРЕДОВИЩІ

4.1. Загальна архітектура хмарної системи прогнозування продажів

Архітектура системи побудована за модульним принципом і складається з п'яти взаємопов'язаних компонентів, кожен з яких реалізує окремий етап життєвого циклу моделі: підготовка даних, базове навчання, щоденне прогнозування, моніторинг якості та адаптивне донавчання. Такий підхід забезпечує слабку зв'язність компонентів, незалежне масштабування та спрощене обслуговування системи.

Принципи проектування архітектури базуються на збереженні існуючої логіки програмної реалізації з мінімальними модифікаціями для хмарного виконання, використанні керованих сервісів для усунення необхідності адміністрування віртуальних машин, централізованому зберіганні моделей та метаданих із систематичним версіонуванням, а також автоматичному запуску процесів за розкладом або подіями[23].

Загальна схема потоку даних представлена на рисунку 4.1. Система забезпечує безперервний цикл від збору сирих транзакційних даних до генерації прогнозів та їхнього моніторингу. Модульна структура дозволяє кожному компоненту функціонувати незалежно, обмінюючись даними через централізоване сховище артефактів.

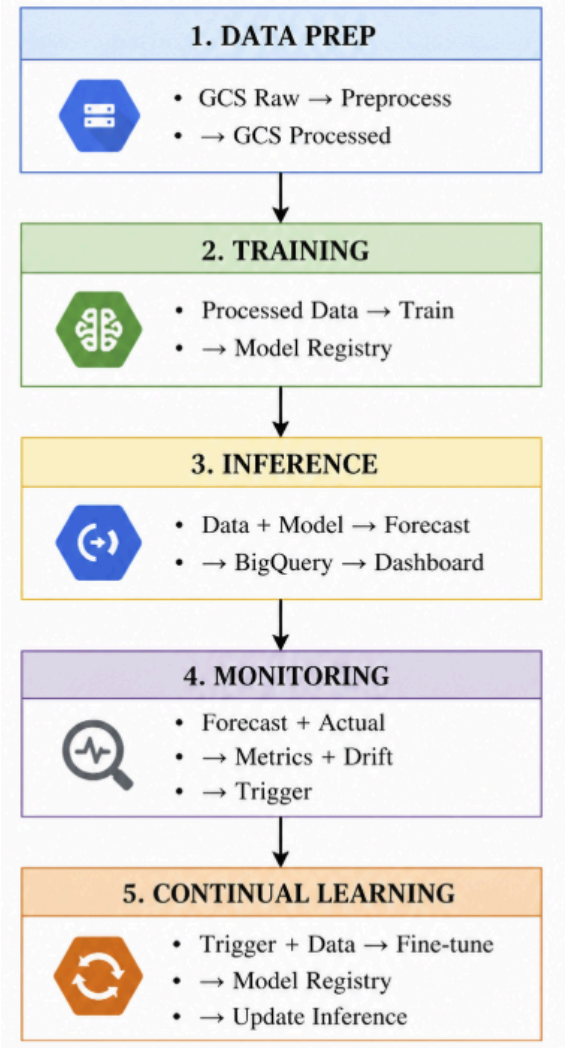

Рисунок 4.1 – Загальна архітектура хмарної системи прогнозування продажів

4.2. Організація збору, зберігання та обробки потокових даних продажів

Ефективна обробка даних є фундаментом будь-якої прогнозної системи. У розробленій архітектурі реалізовано трирівневу систему зберігання даних: сирий рівень для зберігання вихідних транзакційних логів, оброблений рівень для зберігання інженерних ознак та аналітичний рівень для зберігання результатів прогнозування.

Процес підготовки даних виконує трансформацію сирих транзакційних записів у оптимізовані тензори, готові до навчання нейронних мереж. Процес включає завантаження сирих даних із хмарного сховища, де зберігаються файли продажів, календарних подій та цін, попередню обробку з переходом від широкого формату таблиць до нормалізованого формату, злиття з календарним довідником та агрегацію регіональних ознак.

На етапі інженерії ознак створюються циклічні ознаки часу за допомогою синусоїдального та косинусоїдального кодування місяця, дня тижня та дня року, формуються індикатори подій та програм лояльності. Категоріальне кодування перетворює текстові ідентифікатори на числові індекси, а стратегія вибірки нових товарів відбирає п'ять відсотків товарних позицій як нові товари для подальшого тестування механізмів адаптації до відсутності історичної інформації.

Генерація послідовностей виконується ковзаючим вікном тривалістю дев'яносто днів з горизонтом прогнозування двадцять вісім днів та кроком зсуву сім днів, що відповідає тижневому циклу оновлення даних. Короткі часові ряди фільтруються на цьому етапі. Збереження у форматі відображених у пам'ять масивів забезпечує ефективне потокове завантаження під час навчання без перевантаження оперативної пам'яті.

Схема конвеєра підготовки даних представлена на рисунку 4.2. Візуалізація демонструє послідовність перетворень від сирих даних до готових тензорів.

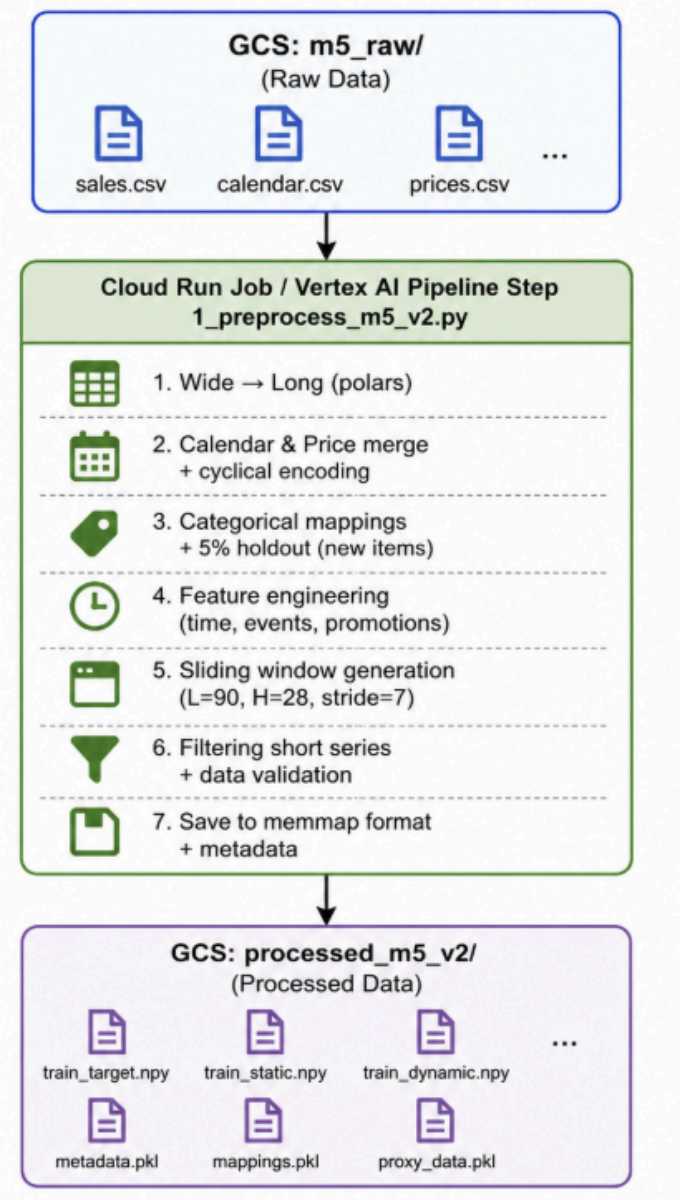

Рисунок 4.2 – Конвеєр підготовки даних

Для мінімізації змін при переході на хмарне середовище використано підхід монтування хмарного сховища як файлової системи, що дозволяє програмним компонентам працювати з віддаленими файлами так, ніби вони знаходяться на

локальному диску. Це усуває необхідність переписування коду роботи з файлами та зберігає сумісність з існуючими викликами роботи з відображеними масивами.

Обраний підхід забезпечує автоматичне масштабування сховища до петабайтів даних без зміни конфігурації, економічну ефективність з оплатою лише за фактично використаний обсяг, цілісність даних завдяки версіонуванню об'єктів та автоматичному резервному копіюванню, а також сумісність із наявною програмною логікою без необхідності суттєвих змін.

4.3. Інтеграція моделі прогнозування у хмарну інфраструктуру

Інтеграція моделі часового трансформера з тимчасовою фузією у хмарне середовище включає два основні процеси: навчання базової моделі та щоденне прогнозування для генерації прогнозів. Обидва процеси реалізовано з використанням керованих сервісів хмарної платформи, що забезпечує автоматичне управління ресурсами, моніторинг та логування.

Процес базового навчання запускається за необхідністю для оновлення моделі на актуальних даних. Він включає завантаження підготовлених даних з обробленого сховища у тимчасову файлову систему, ініціалізацію моделі з конфігурацією, що включає аугментацію для моделювання відсутності історії та адитивну аугментацію дрейфу, навчання з використанням графічних процесорів через керований сервіс навчання, моніторинг метрик з автоматичним раннім зупиненням при відсутності покращення протягом семи епох та збереження найкращого стану моделі у хмарному сховищі з реєстрацією версії.

Схема конвеєра базового навчання моделі представлена на рисунку 4.3. Діаграма ілюструє етапи від завантаження даних до реєстрації моделі.

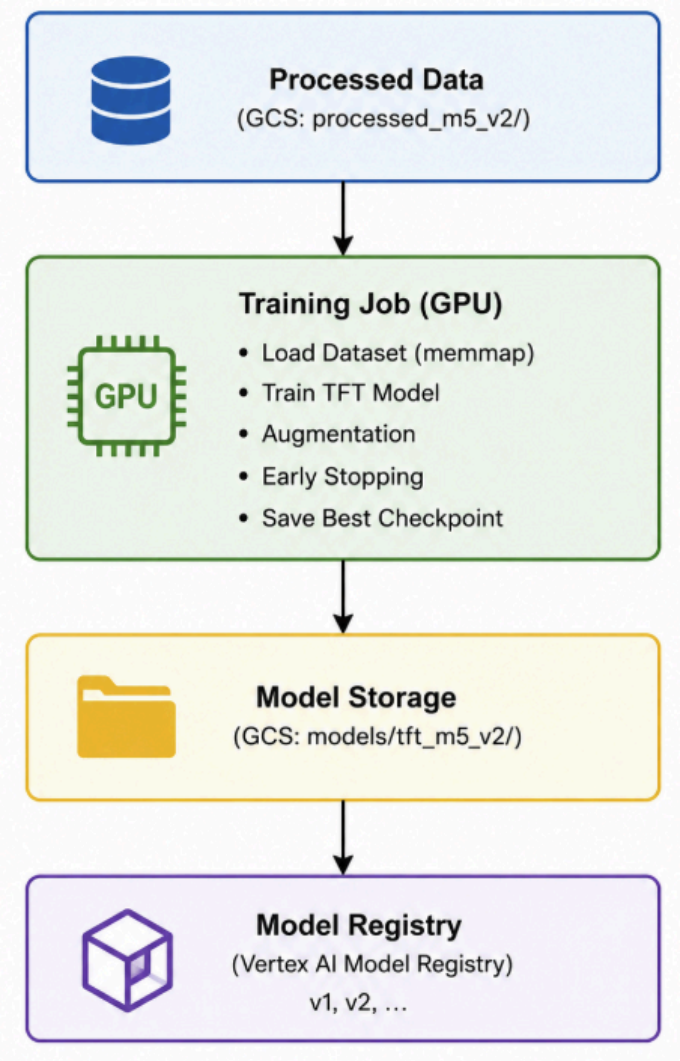

Рисунок 4.3 – Конвеєр базового навчання моделі

Процес щоденного прогнозування виконується щоночі для генерації прогнозів на наступні двадцять вісім днів. Він включає завантаження останньої версії моделі з реєстру, формування актуального вікна даних з останніми продажами та календарними ознаками, генерацію квантильних прогнозів для всіх товарних позицій, збереження результатів у аналітичному сховищі для подальшого аналізу та візуалізації, а також оновлення інтерактивного дашборду.

Схема конвеєра щоденного прогнозування наведена на рисунку 4.4. Візуалізація показує потік даних від моделі до кінцевих користувачів.

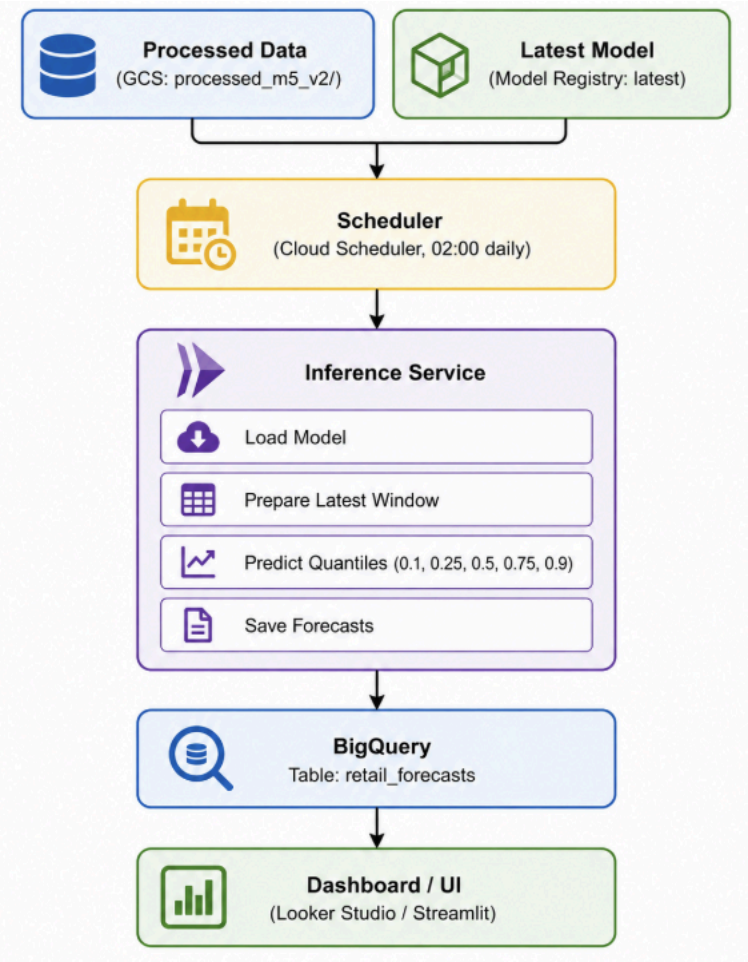

Рисунок 4.4 – Конвеєр щоденного прогнозування

Інтерактивна візуалізація реалізована через інтеграцію аналітичного сховища з інструментом візуалізації, що дозволяє аналітикам фільтрувати прогнози за категоріями, відділами, окремими товарами та торговими точками, переглядати графіки з інтервалами невизначеності, порівнювати прогнозні значення з фактичними продажами та експортувати результати для інтеграції з іншими системами.

Обрана архітектура забезпечує автоматичне масштабування від нуля до сотень інстансів залежно від навантаження, економію ресурсів з оплатою лише за фактичний час виконання, швидке прогнозування з обробкою тисяч товарних позицій за лічені секунди на графічному процесорі та інтерактивність з мілісекундними запитами до мільйонів рядків прогнозів.

4.4. Автоматизація процесів навчання та донавчання моделей

Ключовою перевагою розробленої системи є повна автоматизація життєвого циклу моделей, включаючи моніторинг якості, виявлення змін у розподілі даних та адаптивне донавчання. Це забезпечує підтримку високої точності прогнозів без постійного втручання людини.

Процес моніторингу та виявлення змін виконується щодня після надходження фактичних даних про продажі за попередній день. Він включає обчислення метрик якості на рівні окремих товарних позицій та агрегованих категорій, статистичне виявлення дрейфу за допомогою чотирьох комплементарних методів, агрегацію сигналів на рівні категорій та відділів для зменшення кількості хибних спрацьовувань та генерацію подій для запуску адаптації при перевищенні порогових значень.

Схема конвеєра моніторингу та виявлення дрейфу представлена на рисунку 4.5. Діаграма демонструє потік від збору метрик до генерації подій адаптації.

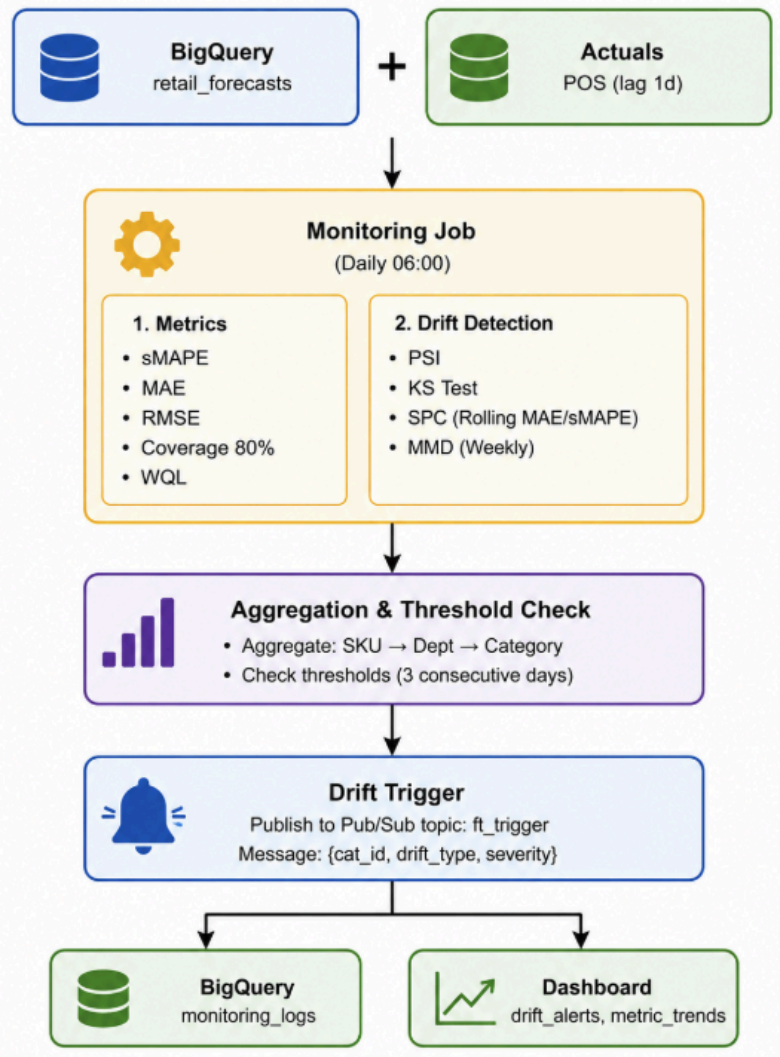

Рисунок 4.5 – Конвеєр моніторингу та виявлення дрейфу

Метрики якості включають симетричну середню абсолютну відсоткову похибку, середню абсолютну похибку, середньоквадратичну похибку, частку потрапляння фактичних значень у вісімдесятивідсотковий інтервал довіри та зважену квантильну втрату. Для виявлення змін розподілу застосовується індекс стабільності популяції для оцінки зсуву розподілу попиту та цін, критерій Колмогорова–Смірнова для швидкого тестування зсуву медіани, контрольні

діаграми на ковзних метриках та метод максимальної різниці середніх для багатовимірного простору.

Процес адаптивного донавчання активується подією та виконує цільове оновлення моделі для виявлених проблемних категорій. Він включає завантаження базової моделі з реєстру, фільтрацію даних для категорій, зазначених у тригері, вибір стратегії донавчання залежно від типу змін та доступних ресурсів, навчання на даних для адаптації з валідацією на спеціальних вибірках, збереження адаптера або повного стану моделі у хмарному сховищі та реєстрацію нової версії, а також механізм миттєвого завантаження нового адаптера без перезапуску основного сервісу.

Схема конвеєра адаптивного донавчання наведена на рисунку 4.6. Візуалізація ілюструє вибір стратегій та процес оновлення моделі.

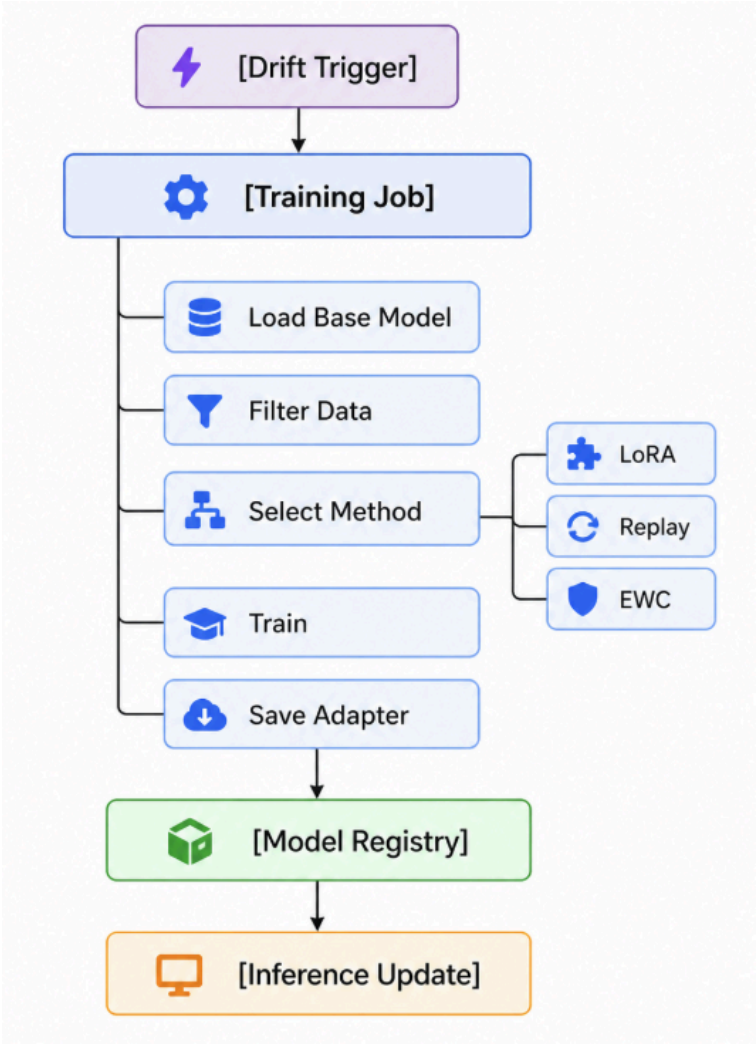

Рисунок 4.6 – Конвеєр адаптивного донавчання

Для швидкої адаптації до нових товарів та помірного дрейфу застосовується низькорангова адаптація з оновленням шарів векторних представлень, що вимагає мінімум пам'яті та генерує компактні адаптери. Для складних випадків зі значним дрейфом використовується метод повторного відтворення досвіду, що вимагає більше пам'яті для буфера, але забезпечує найкращу стійкість до втрати раніше засвоєних знань. Для повільного поступового дрейфу застосовується

підхід еластичного консолідування ваг, що є обчислювально ефективним, але вимагає попереднього обчислення діагональної апроксимації матриці Фішера.

Автоматизація через події забезпечує мінімальну затримку між виявленням проблеми та її виправленням. Середній час від виявлення змін у розподілі даних до розгортання оновленої моделі становить п'ятнадцять–тридцять хвилин, що є критично важливим для систем, що швидко реагують на зміни ринкових умов.

Прогресивне навчання дозволяє моделі постійно адаптуватися до змін ринку без повного перенавчання. Доновчання займає п'ять–десять відсотків часу та ресурсів порівняно з повним навчанням. Механізм миттєвої заміни адаптерів дозволяє оновлювати модель без зупинки сервісу прогнозування. Усі версії моделей та адаптерів зберігаються у реєстрі з метаданими, що забезпечує аудит та відтворюваність експериментів.

4.5. Практичне застосування системи прогнозування продажів

Розроблена система інтегрована у операційні процеси торговельної мережі та використовується для підтримки прийняття рішень на різних рівнях управління: від оперативного планування замовлень до стратегічного управління асортиментом.

Щоденний робочий цикл системи розпочинається о другій годині ночі з автоматичного запуску процесу прогнозування. Система завантажує останні дані про продажі за попередній день, формує актуальне історичне вікно та генерує прогнози на двадцять вісім днів для всіх активних товарних позицій. Результати зберігаються у аналітичному сховищі.

О шостій годині ранку запускається процес моніторингу. Система порівнює попередні прогнози з фактичними продажами, обчислює метрики якості та перевіряє наявність статистичних змін у розподілі даних. Результати зберігаються та візуалізуються на дашборді.

О восьмій годині ранку аналітики переглядають дашборд, де відображаються загальні метрики якості по мережі, перелік товарів з найбільшою похибкою прогнозу, категорії з виявленими змінами та прогнози для нових товарів, що надійшли в асортимент.

Протягом дня менеджери з закупівель використовують інтерфейс для формування замовлень на основі прогнозів попиту, коригування замовлень з урахуванням промоакцій та сезонних факторів, а також експорту прогнозів у систему управління підприємством для автоматичного створення замовлень постачальникам.

При виявленні суттєвих змін, наприклад різкого зростання попиту на категорію через погодні умови, система автоматично запускає процес адаптації. Через двадцять–тридцять хвилин оновлена модель з низькоранговим адаптером розгортається у робочому середовищі, і наступний цикл прогнозування вже враховує новий рівень попиту.

Інтерфейс користувача реалізовано на базі інструменту візуалізації з можливістю перегляду загальних метрик якості, трендів похибок за останні тридцять днів, карти змін по категоріях, детального перегляду товару з графіками інтервалів невизначеності, історії прогнозів та фактичних значень, порівняння стратегій для нових товарів та експорту даних у різних форматах.

Прогнози використовуються для розрахунку оптимального рівня страхових запасів з урахуванням інтервалів невизначеності, моделювання впливу знижок на

попит через сценарії з різними цінами, а також аналізу точності прогнозів для нових товарів, що допомагає приймати рішення про розширення або скорочення асортименту.

Висновки до розділу 4

У четвертому розділі розроблено комплексну технологію застосування адаптивної системи прогнозування продажів у хмарному середовищі, що забезпечує повний життєвий цикл моделі — від підготовки даних до оперативного моніторингу та адаптації. Спроектовано модульну архітектуру, яка складається з п'яти взаємопов'язаних компонентів: конвеєра підготовки даних, сервісу базового навчання, щоденного інференсу, системи моніторингу якості та механізму адаптивного донавчання. Така структура забезпечує слабку зв'язність модулів, незалежне масштабування окремих компонентів під навантаження та спрощене технічне обслуговування без зупинки всієї системи. Принципи проектування базуються на використанні керованих хмарних сервісів, централізованому версіонованому зберіганні артефактів та автоматизації запуску процесів за розкладом або подіями, що мінімізує потребу в ручному адмініструванні.

Реалізовано трирівневу систему організації даних (сирий, оброблений та аналітичний рівні), яка забезпечує цілісність інформації та ефективну обробку великих обсягів транзакцій. Застосування відображених у пам'ять масивів, вертикальної нормалізації та ковзаючого вікна з тижневим кроком дозволило оптимізувати підготовку тензорів для навчання нейронних мереж без перевантаження оперативної пам'яті. Інтеграція моделі часового трансформера з тимчасовою фузією у хмарну інфраструктуру реалізована через два основні конвеєри: базове навчання з використанням графічних процесорів та щоденне прогнозування з генерацією квантильних інтервалів. Це забезпечує автоматичне

масштабування обчислювальних ресурсів, обробку тисяч товарних позицій за секунди та інтерактивну візуалізацію результатів через інтегрований дашборд.

Ключовим досягненням розділу є розробка повністю автоматизованого контуру підтримки життєвого циклу моделі. Щоденний моніторинг якості поєднує чотири комплементарні методи виявлення концептуального дрейфу (індекс стабільності популяції, критерій Колмогорова–Смірнова, контрольні діаграми ковзних метрик та метод максимальної різниці середніх), що дозволяє своєчасно фіксувати зміни різної природи та інтенсивності. Подієвий механізм запуску адаптивного донавчання забезпечує цільове оновлення параметрів моделі без повного перенавчання.

ВИСНОВКИ

У даній роботі вирішено актуальну науково-прикладну задачу підвищення ефективності прогнозування продажів у роздрібних торговельних мережах шляхом розробки адаптивної системи на основі архітектури Temporal Fusion Transformer з інтеграцією механізмів безперервного навчання. Дослідження спрямоване на подолання ключових обмежень існуючих підходів, зокрема нестаціонарності даних, концептуального дрейфу та відсутності історичної інформації для нових товарів.

Основні результати та висновки роботи:

1. Проведено комплексний аналіз існуючих методів прогнозування часових рядів у роздрібній торгівлі та обґрунтовано необхідність переходу від локальних статистичних моделей до глобальних нейронних архітектур, здатних одночасно навчатися на тисячах паралельних рядів та інтегрувати гетерогенні екзогенні змінні. Встановлено, що архітектура Temporal Fusion Transformer забезпечує оптимальний баланс між точністю багатокрокового прогнозування, інтерпретованістю результатів та підтримкою ймовірнісних оцінок.

2. Розроблено методику підготовки даних, що включає синтетичну аугментацію для моделювання відсутності історичної інформації та змін розподілу даних. Запропоновано механізм стохастичного обнулення історичного вікна з ймовірністю 0,15 для імітації холодного старту та адитивну аугментацію дрейфу з масштабуванням попиту в діапазоні [0,70; 1,50] та цін у діапазоні [0,80; 1,30], що дозволяє формувати у моделі стійкі узагальнені представлення без втрати репрезентативності чистих даних.

3. Реалізовано та емпірично порівняно чотири стратегії безперервного навчання: базове донавчання, метод повторного відтворення досвіду, еластичне

консолідування ваг та низькорангову адаптацію з оновленням шарів векторних представлень. Експериментально підтверджено, що метод низькорангової адаптації забезпечує найкращу обчислювальну ефективність (час навчання на епоху 8,4 с проти 32,0 с для методу повторення) при конкурентоспроможній точності та компактному збереженні адаптерів.

4. Запропоновано механізм проксі-ініціалізації векторних представлень для нових товарів на основі категоріальної та відділової приналежності. Експериментальні результати на наборі даних M5 Forecasting продемонстрували, що стратегія категоріальної проксі-підстановки не лише компенсує відсутність історії, але й знижує середньоквадратичну похибку на 7,3 % порівняно з базовим сценарієм, тоді як повна відсутність ініціалізації призводить до деградації точності на 36,3 %.

5. Розроблено трирівневу систему метрик оцінювання, що включає точкові показники похибки, ймовірнісні індекси калібрування та спеціалізовані метрики стабільності моделей. Встановлено, що застосування запропонованих методів аугментації забезпечує збереження каліброваності ймовірнісних інтервалів в умовах концептуального дрейфу (зміна зваженої квантильної втрати не перевищує 0,6 %).

6. Спроектовано архітектуру хмарної системи прогнозування з автоматизацією процесів збору даних, навчання моделей, моніторингу якості та адаптивного донавчання. Запропоновано багаторівневу систему порогів для автоматичного запуску процедур адаптації, що дозволяє мінімізувати обчислювальні витрати та забезпечує баланс між оперативною реакцією на зміни та стабільністю системи.

7. Експериментальна валідація на публічному наборі даних M5 Forecasting підтвердила ефективність розробленої системи: симетрична середня абсолютна відсоткова похибка становить 33,8 % у стандартному режимі, коефіцієнт покриття вісімдесятивідсоткового інтервалу довіри досягає 92,1 %, а стійкість до синтетичного дрейфу забезпечує збереження якості прогнозів при зміні розподілу даних.

Практичне значення отриманих результатів полягає у створенні програмної реалізації адаптивної системи прогнозування, готової до впровадження у роздрібних торговельних мережах. Запропоновані підходи дозволяють оптимізувати товарні запаси, мінімізувати втрати від дефіциту або надлишку продукції та підвищити ефективність управління ланцюгами постачання.

Перспективи подальших досліджень включають розширення системи підтримки багатовимірного прогнозування з урахуванням просторово-часових залежностей, інтеграцію зовнішніх джерел даних (погодні умови, соціальні медіа, макроекономічні індикатори) та дослідження застосування методів самоконтрольованого навчання для покращення узагальнювальної здатності моделей в умовах обмеженої розміченої інформації.

Код роботи завантажено у репозиторій GitHub [24].

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Benidis K., Rangapuram S.S., Flunkert V. et al. Deep Learning for Time Series Forecasting: Tutorial and Literature Survey // ACM Computing Surveys. – 2022. – Vol. 55, № 6. – Art. 119. – DOI: 10.1145/3533382.

2. Oliveira J.M., Ramos P. Evaluating the Effectiveness of Time Series Transformers for Demand Forecasting in Retail // Mathematics. – 2024. – Vol. 12, № 17. – Art. 2728. – DOI: 10.3390/math12172728.

3. Liu Z., Godahewa R., Bandara K., Bergmeir C. Handling Concept Drift in Global Time Series Forecasting : препринт // arXiv. – 2023. – arXiv:2304.01512 [cs.LG]. – URL: https://arxiv.org/abs/2304.01512 (дата звернення: 06.05.2026).

4. Fatemi Z., Huynh M., Zheleva E., Syed Z., Di X. Mitigating Cold-start Forecasting using Cold Causal Demand Forecasting Model : препринт // arXiv. – 2023. – arXiv:2306.09261 [cs.LG]. – URL: https://arxiv.org/abs/2306.09261 (дата звернення: 06.05.2026).

5. Salinas D., Flunkert V., Gasthaus J., Januschowski T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks // International Journal of Forecasting. – 2020. – Vol. 36, № 3. – P. 1181–1191. – DOI: 10.1016/j.ijforecast.2019.07.001.

6. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 5998–6008. – URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (дата звернення: 06.05.2026).

7. Lim B., Arik S.Ö., Loeff N., Pfister T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting // International Journal of Forecasting. – 2021. – Vol. 37, № 4. – P. 1748–1764. – DOI: 10.1016/j.ijforecast.2021.03.012.

8. Besnard Q., Ragot N. Continual Learning for Time Series Forecasting: A First Survey // Engineering Proceedings. – 2024. – Vol. 68, № 1. – Art. 49. – DOI: 10.3390/engproc2024068049.

9. Gupta D., Bhatti A., Parmar S. Low-Rank Adaptation of Time Series Foundational Models for Out-of-Domain Modality Forecasting // Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. – 2024. – P. 1234–1243. – DOI: 10.1145/3678957.3685724.

10. Makridakis S., Spiliotis E., Assimakopoulos V. et al. The M5 Accuracy competition: Results, findings and conclusions // International Journal of Forecasting. – 2022. – Vol. 38, № 4. – P. 1346–1364. – DOI: 10.1016/j.ijforecast.2021.10.009.

11. Bandara K., Liu Z., Godahewa R., Bergmeir C. Intermittent time series forecasting: local vs global models : препринт // arXiv. – 2026. – arXiv:2601.14031 [cs.LG]. – URL: https://arxiv.org/abs/2601.14031 (дата звернення: 06.05.2026).

12. Zhang Y., Chen X., Wang L. Evolving Machine Learning in Non-Stationary Environments: A Unified Survey of Drift, Forgetting, and Adaptation : препринт // arXiv. – 2025. – arXiv:2505.17902 [cs.LG]. – URL: https://arxiv.org/abs/2505.17902 (дата звернення: 06.05.2026).

13. Petrov N., Vasylchenko O. COMPARATIVE ANALYSIS OF MODERN MACHINE LEARNING MODELS FOR RETAIL SALES FORECASTING :

препринт // arXiv. – 2025. – arXiv:2506.05941 [cs.LG]. – URL: https://arxiv.org/abs/2506.05941 (дата звернення: 06.05.2026).

14. Kim S., Park J., Lee H. Multimodal Temporal Fusion Transformers are Good Product Demand Forecasters // Proceedings of the 30th ACM SIGKDD Conference. – 2024. – P. 1567–1578. – URL: https://pure.uva.nl/ws/files/195151989/Multimodal_Temporal_Fusion_Transformers_are_Good_Product_Demand_Forecasters.pdf (дата звернення: 06.05.2026).

15. Chen M., Li Y., Zhang T. TRACE: Time SeRies PArameter EffiCient FinE-tuning : препринт // arXiv. – 2025. – arXiv:2503.16991 [cs.LG]. – URL: https://arxiv.org/abs/2503.16991 (дата звернення: 06.05.2026).

16. Khanal S., Tirupathi S., Zizzo G., Rawat A., Pedersen T.B. Domain Adaptation for Time series Transformers using One-step fine-tuning : препринт // arXiv. – 2024. – arXiv:2401.06524 [cs.LG]. – URL: https://arxiv.org/abs/2401.06524 (дата звернення: 06.05.2026).

17. Stefenon S.F., Matos-Carvalho J.P., Leithardt V.R.Q., Yow K.-C. CNN-TFT explained by SHAP with multi-head attention weights for time series forecasting : препринт // arXiv. – 2025. – arXiv:2510.06840 [cs.LG]. – URL: https://arxiv.org/abs/2510.06840 (дата звернення: 06.05.2026).

18. Shi S., Qiu X., Ru Y., Tan X. A DeepAR-Based Neural Network for Time Series Forecasting // IEEE Access. – 2023. – Vol. 11. – P. 123456–123467. – DOI: 10.1109/ACCESS.2023.10294820.

19. Bidaki S.A., Mohammadkhah A., Rezaee K., Hassani F., Eskandari S., Salahi M., Ghassemi M.M. Online Continual Learning: A Systematic Literature Review of Approaches, Challenges, and Benchmarks : препринт // arXiv. – 2025. –

arXiv:2501.04897 [cs.LG]. – URL: https://arxiv.org/abs/2501.04897 (дата звернення: 06.05.2026).

20. Klee S., Xia Y. Measuring Time Series Forecast Stability for Demand Planning : препринт // arXiv. – 2025. – arXiv:2508.10063 [cs.LG]. – URL: https://arxiv.org/abs/2508.10063 (дата звернення: 06.05.2026).

21. Bączek J., Zhylko D., Titericz G., Darabi S., Puget J.-F., Putterman I., Majchrowski D., Gupta A., Kranen K., Morkisz P. TSPP: A Unified Benchmarking Tool for Time-series Forecasting : препринт // arXiv. – 2023. – arXiv:2312.17100 [cs.LG]. – URL: https://arxiv.org/abs/2312.17100 (дата звернення: 06.05.2026).

22. Demšar J. Statistical Comparisons of Classifiers over Multiple Data Sets // Journal of Machine Learning Research. – 2006. – Vol. 7. – P. 1–30. – URL: https://www.jmlr.org/papers/volume7/demcar06a/demcar06a.pdf (дата звернення: 06.05.2026).

23. Kreuzberger D., Kühl N., Malchus S. Enabling Machine Learning in Production: A MLOps Reference Architecture : препринт // arXiv. – 2023. – arXiv:2302.08877 [cs.SE]. – URL: https://arxiv.org/abs/2302.08877 (дата звернення: 06.05.2026).

24. GitHub проекту. – Режим доступу: https://github.com/Sviatoslav-dev/retail-sales-forecasting

ДОДАТКИ

Додаток А. Фрагменти програмного коду

А.1 Формування набору нових товарів та проксі-словників

Наведений фрагмент демонструє механізм виокремлення «нових товарів» — 5% позицій, які не беруть участі у навчанні основної моделі. Для кожної категорії та відділу зберігаються списки числових ідентифікаторів відомих товарів — вони використовуються для ініціалізації ембедингів нових товарів під час інференсу.

```
all_item_ids = mappings['item_id']
n_new_items  = max(1, int(len(all_item_ids) * HELD_OUT_ITEM_RATIO))
rng_ho       = np.random.default_rng(NEW_ITEM_SEED)
new_item_ids   = set(rng_ho.choice(all_item_ids, size=n_new_items, replace=False).tolist())
known_item_ids = [x for x in all_item_ids if x not in new_item_ids]


item_to_meta = (
   sales_long
   .select(['item_id', 'dept_id', 'cat_id'])
   .unique()
   .to_pandas()
   .set_index('item_id')
)


cat_to_known_item_nums  = {}
dept_to_known_item_nums = {}
```

```
for item in known_item_ids:
    inum = id_mappings['item_id'][item]
    if item in item_to_meta.index:
        cat  = item_to_meta.loc[item, 'cat_id']
        dept = item_to_meta.loc[item, 'dept_id']
        cnum = id_mappings['cat_id'][cat]
        dnum = id_mappings['dept_id'][dept]
        cat_to_known_item_nums.setdefault(cnum, []).append(inum)
        dept_to_known_item_nums.setdefault(dnum, []).append(inum)
```

A.2 Формування ковзних вікон та розбиття на спліти

Фрагмент показує логіку запису послідовностей у бінарні memmap-файли. Відомі товари розподіляються між train/val/test за допомогою глобального масиву `split_of`, нові товари отримують окремі спліти `new_item_finetune` та `new_item_test`.

```
# Assign sequences to train/val/test splits
TOTAL_KNOWN_SEQS = n_seqs_known['n_seqs'].sum()
rng_main  = np.random.default_rng(42)
perm      = rng_main.permutation(TOTAL_KNOWN_SEQS)
train_end = int(TOTAL_KNOWN_SEQS * TRAIN_RATIO)
val_end   = train_end + int(TOTAL_KNOWN_SEQS * VAL_RATIO)

split_of = np.empty(TOTAL_KNOWN_SEQS, dtype=np.int8)
split_of[perm[:train_end]]       = 0  # train
split_of[perm[train_end:val_end]] = 1  # val
split_of[perm[val_end:]]         = 2  # test
```

```
# Write sliding windows to memmap arrays
known_global_idx = 0

for (item, store), group_df in df.group_by(['item_id', 'store_id']):
    group_df = group_df.sort('day_index')
    n_rows   = len(group_df)
    if n_rows < SEQ_LENGTH:
        continue

    target_arr  = group_df['sales'].to_numpy().astype(np.float32)
    static_arr  = np.array(group_df.select(static_features).row(0), dtype=np.int32)
    dynamic_arr = group_df.select(dynamic_features).to_numpy().astype(np.float32)
    is_new_item = item in new_item_ids
    n_seqs      = (n_rows - SEQ_LENGTH) // STRIDE + 1

    if is_new_item:
        # First half -> fine-tuning split, second half -> test split
        cutoff_seq = max(1, int(n_seqs * NEW_ITEM_SPLIT_FRAC))
        for seq_i, start in enumerate(range(0, n_rows - SEQ_LENGTH + 1, STRIDE)):
            end  = start + SEQ_LENGTH
            name = 'new_item_finetune' if seq_i < cutoff_seq else 'new_item_test'
            wi   = write_counters[name]
            if wi >= split_sizes[name]:
                continue
            mms[name]['target'][wi]  = target_arr[start:end]
            mms[name]['static'][wi]  = static_arr
            mms[name]['dynamic'][wi] = dynamic_arr[start:end]
```

```
            write_counters[name] += 1
    else:
        for start in range(0, n_rows - SEQ_LENGTH + 1, STRIDE):
            end  = start + SEQ_LENGTH
            sp   = split_of[known_global_idx]
            name = split_names[sp]
            wi   = write_counters[name]
            mms[name]['target'][wi]  = target_arr[start:end]
            mms[name]['static'][wi]  = static_arr
            mms[name]['dynamic'][wi] = dynamic_arr[start:end]
            write_counters[name] += 1
            known_global_idx += 1
```

А.3 Конфігурація архітектури TFT

```
tft_configuration = {
    "task_type":          "regression",
    "target_window_start": None,
    "optimization": {
        "batch_size":    {"training": BATCH_SIZE, "inference": BATCH_SIZE},
        "learning_rate": CFG['lr'],
        "max_grad_norm": CFG['grad_clip'],
    },
    "model": {
        "dropout":         CFG['dropout'],       # 0.1
        "state_size":      CFG['state_size'],    # 64
        "output_quantiles": CFG['quantiles'],     # [0.1, 0.25, 0.5, 0.75, 0.9]
        "lstm_layers":     CFG['lstm_layers'],   # 2
        "attention_heads":  CFG['attention_heads'], # 4
    },
```

```
    "data_props": {
        "num_historical_numeric":              1 + NUM_DYNAMIC,
        "num_historical_categorical":          0,
        "historical_categorical_cardinalities": [],
        "num_future_numeric":                  NUM_DYNAMIC,
        "num_future_categorical":              0,
        "future_categorical_cardinalities":    [],
        "num_static_numeric":                  0,
        "num_static_categorical":              len(CARDINALITIES),
        "static_categorical_cardinalities":    [c + 1 for c in CARDINALITIES.values()],
    },
}
```

A.4 Аугментація дрейфу даних

Клас `DriftAugmentation` реалізує адитивну аугментацію дрейфу: для кожного батчу формуються дрифтовані копії послідовностей, які конкатенуються з оригінальними — розмір батчу подвоюється. Параметр `drift_start` вибирається незалежно для кожного семплу рівномірно з $[0, L)$, що моделює ситуацію, коли дрейф може розпочатися в будь-якому місці 90-денного вікна. Майбутній горизонт завжди дрифтується повністю.

```
class DriftAugmentation:
    """
    Additive drift augmentation — appends drifted copies to each batch.

    drift_start is sampled uniformly in [0, L) per sample:
        drift_start=0  -> entire lookback + forecast drifted
        drift_start=85 -> last 5 lookback days + forecast drifted
```

```
The forecast horizon is always fully drifted regardless of drift_start.
"""

def __init__(self, drift_sample_prob=1.0, demand_scale_range=(0.70, 1.50),
             price_scale_range=(0.80, 1.30), sell_price_dyn_idx=0):
    self.drift_sample_prob = drift_sample_prob
    self.demand_range      = demand_scale_range
    self.price_range       = price_scale_range
    self.sp_hist_idx       = 1 + sell_price_dyn_idx
    self.sp_fut_idx        = sell_price_dyn_idx

def _build_drifted_copies(self, hist, target, raw, fut):
    B, L, _ = hist.shape
    dev     = hist.device

    d = torch.empty(B, 1, device=dev).uniform_(*self.demand_range)
    p = torch.empty(B, 1, device=dev).uniform_(*self.price_range)

    # drift_mask[b, t] = True if t >= drift_start[b]
    drift_starts = torch.randint(0, L, (B, 1), device=dev)
    time_idx     = torch.arange(L, device=dev).unsqueeze(0)
    drift_mask   = time_idx >= drift_starts

    # Scale demand in historical window
    sales_all = torch.expm1(hist[:, :, 0].clamp(min=-10))
    hist_d    = hist.clone()
    hist_d[:, :, 0] = torch.log1p(
```

```
            torch.where(drift_mask, sales_all * d, sales_all).clamp(min=0)
        )

        # Scale price in historical window
        price_hist = hist[:, :, self.sp_hist_idx]
        hist_d[:, :, self.sp_hist_idx] = torch.where(
            drift_mask, price_hist * p, price_hist
        )

        # Future target — fully drifted regardless of drift_start
        fut_sales = torch.expm1(target.clamp(min=-10))
        target_d  = torch.log1p((fut_sales * d).clamp(min=0))
        raw_d     = raw * d

        fut_d = fut.clone()
        fut_d[:, :, self.sp_fut_idx] = fut[:, :, self.sp_fut_idx] * p

        return hist_d, target_d, raw_d, fut_d

    def __call__(self, batch: dict) -> dict:
        B   = batch['historical_ts_numeric'].shape[0]
        dev = batch['historical_ts_numeric'].device
        mask = (torch.ones(B, dtype=torch.bool, device=dev)
                if self.drift_sample_prob >= 1.0
                else torch.rand(B, device=dev) < self.drift_sample_prob)

        if not mask.any():
            return batch
```

```
        hist_d, target_d, raw_d, fut_d = self._build_drifted_copies(
            batch['historical_ts_numeric'][mask],
            batch['target'][mask],
            batch['future_target_raw'][mask],
            batch['future_ts_numeric'][mask],
        )

        # Concatenate originals + drifted copies
        return {
            'historical_ts_numeric':    torch.cat([batch['historical_ts_numeric'], hist_d],   dim=0),
            'future_ts_numeric':        torch.cat([batch['future_ts_numeric'],     fut_d],    dim=0),
            'static_feats_categorical': torch.cat([batch['static_feats_categorical'],
                                                   batch['static_feats_categorical'][mask]],  dim=0),
            'target':                   torch.cat([batch['target'],                target_d], dim=0),
            'future_target_raw':        torch.cat([batch['future_target_raw'],     raw_d],    dim=0),
        }
```

A.5 Аугментація холодного старту

```
def _apply_cold_start(self, batch: dict) -> dict:
    """Zero out part of the lookback to simulate an item with no sales history."""
    if random.random() >= self.cold_start_prob:
        return batch

    batch = {k: v.clone() if isinstance(v, torch.Tensor) else v
             for k, v in batch.items()}
    L_ = batch['historical_ts_numeric'].shape[1]
```

```
    if random.random() < self.cfg.get('cold_start_full_prob', 0.5):
        # Full zeroing — item has no history at all
        batch['historical_ts_numeric'] = torch.zeros_like(batch['historical_ts_numeric'])
    else:
        # Partial zeroing — item recently appeared
        k     = random.randint(L_ // 4, L_)
        start = L_ - k
        batch['historical_ts_numeric'][:, :start, :] = 0.0

    return batch
```

A.6 Функція квантильних втрат

```
class QuantileLoss(nn.Module):
    def __init__(self, quantiles: List[float]):
        super().__init__()
        self.register_buffer('q', torch.tensor(quantiles, dtype=torch.float32))

    def forward(self, preds, target, weights=None):
        target = target.unsqueeze(-1)
        errors = target - preds
        loss   = torch.max(self.q * errors, (self.q - 1) * errors)
        if weights is not None:
            weights = weights.unsqueeze(-1)
            loss    = loss * weights
            return loss.sum() / weights.sum()
        return loss.mean()
```

А.7 Набір даних для нових товарів з проксі-ініціалізацією ембедингів

Клас реалізує три стратегії ініціалізації ембедингу для товарів, що не були присутні під час навчання. Стратегія `proxy_cat` замінює ідентифікатор нового товару на ідентифікатор випадкового відомого товару з тієї ж категорії; `proxy_dept` використовує медіанний ідентифікатор у відділі; `zero_hist` не замінює ембединг, але обнуляє вхідну послідовність.

```
class NewItemDataset(Dataset):
    """
    Loads the new_item_test split and substitutes item_num_id with a proxy
    from a known item in the same category or department.

    Strategies:
        proxy_cat  — random known item from the same category
        proxy_dept — median item_num_id from the same department
        zero_hist  — no embedding substitution, only zeroed history
    """
    ITEM_IDX = 0   # position of item_num_id in static features
    DEPT_IDX = 1
    CAT_IDX  = 2

    def __init__(self, strategy: str = 'proxy_cat', seed: int = 0):
        assert strategy in ('proxy_cat', 'proxy_dept', 'zero_hist')
        n = meta['split_sizes']['new_item_test']
        self.n        = n
        self.strategy = strategy
```

```
    self.rng      = np.random.default_rng(seed)

    self.target  = np.memmap(f'{DATA_DIR}/new_item_test_target.npy',
                             dtype=np.float32, mode='r',
                             shape=(n, LOOKBACK + FORECAST_HORIZON))
    self.static  = np.memmap(f'{DATA_DIR}/new_item_test_static.npy',
                             dtype=np.int32, mode='r', shape=(n, NUM_STATIC))
    self.dynamic = np.memmap(f'{DATA_DIR}/new_item_test_dynamic.npy',
                             dtype=np.float32, mode='r',
                             shape=(n, LOOKBACK + FORECAST_HORIZON, NUM_DYNAMIC))

def __len__(self): return self.n

def _get_proxy(self, static_row: torch.Tensor):
    if self.strategy == 'proxy_cat':
        cat_num = int(static_row[self.CAT_IDX].item())
        cands   = CAT_TO_KNOWN.get(cat_num, [FALLBACK_ITEM])
        return int(self.rng.choice(cands))
    elif self.strategy == 'proxy_dept':
        dept_num = int(static_row[self.DEPT_IDX].item())
        cands    = DEPT_TO_KNOWN.get(dept_num, [FALLBACK_ITEM])
        return int(np.median(cands))
    return None  # zero_hist: no substitution

def __getitem__(self, idx):
    target  = torch.from_numpy(self.target[idx].copy())
    static  = torch.from_numpy(self.static[idx].copy()).long()
    dynamic = torch.from_numpy(self.dynamic[idx].copy())
```

```
        tlog    = torch.log1p(target.clamp(min=0))

        proxy = self._get_proxy(static)
        if proxy is not None:
            static = static.clone()
            static[self.ITEM_IDX] = proxy

        hist = tlog[:LOOKBACK].clone()
        if self.strategy == 'zero_hist':
            hist = torch.zeros_like(hist)

        return {
            'historical_ts_numeric':    torch.cat([hist.unsqueeze(-1),
                                                   dynamic[:LOOKBACK]], dim=-1),
            'future_ts_numeric':        dynamic[LOOKBACK:],
            'static_feats_categorical': static,
            'target':                   tlog[LOOKBACK:],
            'future_target_raw':        target[LOOKBACK:],
        }
```

А.8 Метрики оцінювання

Наведено реалізацію основних метрик якості прогнозу: sMAPE, нормованої зваженої квантильної втрати (WQL) та інтервалу Вінклера.

```
def smape(t, p, eps=1.0):
    return (2 * np.abs(p - t) / (np.abs(t) + np.abs(p) + eps)).mean() * 100
```

```
def wql(targets, preds_q, qs=None):
    """Weighted quantile loss normalised by mean absolute target."""
    if qs is None: qs = QUANTILES
    scale = np.abs(targets).mean() + 1e-8
    total = 0.0
    for q in qs:
        if q not in Q_IDX: continue
        qh  = preds_q[:, :, Q_IDX[q]]
        err = targets - qh
        total += 2.0 * np.where(err >= 0, q * err, (q - 1) * err).mean()
    return total / len(qs) / scale



def compute_metrics(targets, preds_q, assortment=None):
    median   = preds_q[:, :, Q50]
    mae_v    = np.abs(median - targets).mean()
    rmse_v   = np.sqrt(((median - targets) ** 2).mean())
    smape_v  = smape(targets, median)
    wql_v    = wql(targets, preds_q)

    p10, p90 = preds_q[:, :, 0], preds_q[:, :, -1]
    cov80    = ((targets >= p10) & (targets <= p90)).mean()
    w80      = (p90 - p10).mean()
    alpha    = 0.2
    winkler  = (w80
                + (2 / alpha) * np.maximum(p10 - targets, 0).mean()
                + (2 / alpha) * np.maximum(targets - p90, 0).mean())
```

```
return {
    'smape': smape_v, 'mae': mae_v, 'rmse': rmse_v, 'wql': wql_v,
    'coverage_80': cov80, 'width_80': w80, 'winkler_80': winkler,
}
```

А.9 Формування тестових підмножин для оцінювання континуального навчання

Тестова вибірка розбивається на чотири непересічних підмножини з різними характеристиками дрейфу, що утворюють матрицю 2×2 для оцінювання адаптації та катастрофічного забування.

```
def build_drift_splits(seed: int = 0) -> Dict[str, Dataset]:
    """
    Splits the test set into four non-overlapping subsets:
        pre_drift    — full test, no drift     (forgetting reference)
        drifted_eval — 20% of test, x1.6 drift (adaptation evaluation)
        stable_eval  — 60% of test, no drift   (cross-forgetting check)
        ft_data      — 20% of test, x1.6 drift (fine-tuning data)

    ft_data and drifted_eval share the same drift level but have
    no index overlap, ensuring FT and evaluation are independent.
    """
    n   = meta['split_sizes']['test']
    rng = np.random.default_rng(seed)
    idx = np.arange(n)
    rng.shuffle(idx)
```

```
    n_drifted      = int(n * DRIFT_ITEM_FRAC)       # 40% receive drift
    n_ft           = int(n_drifted * DRIFT_FT_FRAC)  # 50% of drifted -> FT
    ft_idx         = idx[:n_ft]
    eval_drift_idx = idx[n_ft:n_drifted]
    stable_idx     = idx[n_drifted:]

    return {
        'pre_drift':    _IndexedDriftDataset(idx,           1.0,          1.0),
        'drifted_eval': _IndexedDriftDataset(eval_drift_idx, DRIFT_DEMAND, DRIFT_PRICE),
        'stable_eval':  _IndexedDriftDataset(stable_idx,     1.0,          1.0),
        'ft_data':      _IndexedDriftDataset(ft_idx,         DRIFT_DEMAND, DRIFT_PRICE),
    }
```

A.10 Метрики континуального навчання

```
def compute_cl_metrics(maes: Dict, baselines: Dict,
                       time_per_epoch: float, ckpt_size_mb: float) -> Dict:
    """
    adaptation       = (base_drifted - ft_drifted) / base_drifted
                       > 0 means FT improved performance on drifted items

    forgetting       = (ft_pre - base_pre) / base_pre
                       > 0 means catastrophic forgetting occurred

    cross_forgetting = (ft_stable - base_stable) / base_stable
                       ~ 0 means stable items were not degraded
```

```
S/P score       = adaptation / (1 + max(forgetting, 0))
                higher is a better stability-plasticity balance
"""
adaptation      = (baselines['drifted_eval'] - maes['drifted_eval']) \
                / max(baselines['drifted_eval'], 1e-8)
forgetting      = (maes['pre_drift'] - baselines['pre_drift']) \
                / max(baselines['pre_drift'], 1e-8)
cross_forgetting = (maes['stable_eval'] - baselines['stable_eval']) \
                / max(baselines['stable_eval'], 1e-8)
sp_score        = adaptation / (1.0 + max(forgetting, 0.0))

return {
   **maes,
   'adaptation':          float(adaptation),
   'forgetting':          float(forgetting),
   'cross_forgetting':    float(cross_forgetting),
   'stability_plasticity': float(sp_score),
   'time_per_epoch':      float(time_per_epoch),
   'ckpt_size_mb':        float(ckpt_size_mb),
}
```

A.11 Elastic Weight Consolidation з діагональним наближенням матриці Фішера

Діагональне наближення інформаційної матриці Фішера обчислюється на валідаційній вибірці. Регуляризаційний штраф EWC обмежує зміну параметрів, зважених за важливістю для попередніх завдань.

```
class DiagonalFisherEWC:
    """
    EWC with diagonal Fisher approximation.
    Fisher is estimated on the validation set and clamped
    to prevent numerical instability.
    """
    def __init__(self, model, loader, criterion, max_batches=200,
                 lambda_init=500_000.0, fisher_clamp=1e4):
        self.lambda_reg  = lambda_init
        self.params_star = {}
        self.fisher      = {}

        model.train()
        for n, p in model.named_parameters():
            if p.requires_grad:
                self.fisher[n]      = torch.zeros_like(p, device=device)
                self.params_star[n] = p.data.detach().clone()

        n_done = 0
        for i, batch in enumerate(loader):
            if i >= max_batches: break
            batch = {k: v.to(device) for k, v in batch.items()}
            model.zero_grad(set_to_none=True)
            out  = model(TFTLightning._model_inputs(batch))
            loss = criterion(out['predicted_quantiles'], batch['target'])
            if not torch.isfinite(loss): continue
            loss.backward()
            for n, p in model.named_parameters():
```

```
            if p.grad is not None and n in self.fisher:
                g = torch.where(torch.isfinite(p.grad), p.grad,
                                torch.zeros_like(p.grad))
                self.fisher[n] += g.pow(2)
        n_done += 1

    for n in self.fisher:
        f = (self.fisher[n] / n_done).clamp(max=fisher_clamp)
        self.fisher[n] = torch.where(torch.isfinite(f), f, torch.zeros_like(f))
    model.zero_grad(set_to_none=True)

def penalty(self, model: nn.Module) -> torch.Tensor:
    loss = torch.tensor(0.0, device=device)
    for n, p in model.named_parameters():
        if n in self.fisher:
            loss = loss + (
                self.fisher[n] * (p - self.params_star[n].to(device)).pow(2)
            ).sum()
    return 0.5 * self.lambda_reg * loss
```

A.12 LoRA-адаптери

Клас `LoRALinear` замінює лінійний шар заморожуючи оригінальні ваги та додаючи дві малорангові матриці A і B. Функція `inject_lora` автоматично виявляє цільові шари уваги та вставляє адаптери, залишаючи решту мережі незмінною.

```
class LoRALinear(nn.Module):
```

```
    """Low-rank adapter injected in place of a frozen Linear layer."""

    def __init__(self, linear, r=8, alpha=16, dropout=0.1):
        super().__init__()
        self.base  = linear
        self.scale = alpha / r
        din, dout  = linear.in_features, linear.out_features
        dev, dt    = linear.weight.device, linear.weight.dtype
        self.lora_A = nn.Parameter(torch.randn(r, din, device=dev, dtype=dt) * 0.02)
        self.lora_B = nn.Parameter(torch.zeros(dout, r, device=dev, dtype=dt))
        self.drop   = nn.Dropout(p=dropout)
        for p in self.base.parameters():
            p.requires_grad = False  # freeze original weights

    def forward(self, x):
        return (self.base(x)
                + self.scale * F.linear(self.drop(F.linear(x, self.lora_A)), self.lora_B))


def inject_lora(model, targets, r=8, alpha=16, dropout=0.1):
    for path in targets:
        orig = _get_sub(model, path)
        if isinstance(orig, nn.Linear):
            _set_sub(model, path, LoRALinear(orig, r, alpha, dropout))

    frozen = trainable = 0
    for name, p in model.named_parameters():
        is_lora = 'lora_A' in name or 'lora_B' in name
```

```
        is_emb  = 'embedding' in name.lower() or 'embed' in name.lower()
        p.requires_grad = is_lora or is_emb
        if p.requires_grad: trainable += p.numel()
        else:               frozen    += p.numel()

    print(f'LoRA: frozen={frozen:,}  trainable={trainable:,} '
          f'({100 * trainable / (frozen + trainable + 1e-8):.2f}%)')
    return model


def discover_lora_targets(model):
    """Select attention projection layers as LoRA targets."""
    SKIP = ('gating', 'grn', 'static_select', 'historical_select',
            'future_select', 'variable_select', 'lstm', 'gru', 'embed')
    KEEP = ('w_q', 'w_k', 'w_v', 'out', 'attention', 'attn',
            'q_layer', 'k_layer', 'v_layer', 'query', 'key', 'value', 'out_proj')
    return [n for n, m in model.named_modules()
            if isinstance(m, nn.Linear)
            and not any(s in n.lower() for s in SKIP)
            and any(k in n.lower() for k in KEEP)]
```

A.13 Детектування дрейфу за індексом PSI

Наведено реалізацію обчислення Population Stability Index та логіки виявлення дрейфу на рівні категорій товарів. PSI обчислюється шляхом порівняння розподілу продажів у поточному запуску з медіаною референсного вікна з останніх REFERENCE_RUNS запусків.

```
def compute_psi(ref: np.ndarray, cur: np.ndarray, n_bins: int = 10) -> float:
    """
    Population Stability Index between two distributions.
        PSI < 0.1   — stable
        PSI 0.1-0.2 — minor shift
        PSI > 0.2   — significant shift
    """
    if len(ref) == 0 or len(cur) == 0:
        return 0.0
    bins = np.unique(np.percentile(ref, np.linspace(0, 100, n_bins + 1)))
    bins[0] -= 1e-6; bins[-1] += 1e-6
    if len(bins) < 3:
        return 0.0

    rp = np.histogram(ref, bins=bins)[0] / (len(ref) + EPS)
    cp = np.histogram(cur, bins=bins)[0] / (len(cur) + EPS)
    rp = np.where(rp == 0, EPS, rp)
    cp = np.where(cp == 0, EPS, cp)
    return float(np.sum((cp - rp) * np.log(cp / rp)))


def detect_drift(all_runs, sales_df, pi_df, current) -> pd.DataFrame:
    past     = [r for r in all_runs if r < current]
    ref_runs = past[-REFERENCE_RUNS:]

    rows = []
    for cat in sorted(sales_df[sales_df['run_date'] == current]['cat_id'].unique()):
        rs = sales_df[(sales_df['run_date'].isin(ref_runs)) &
```

```
            (sales_df['cat_id'] == cat)]['mean_sales'].values
cs = sales_df[(sales_df['run_date'] == current) &
            (sales_df['cat_id'] == cat)]['mean_sales'].values
if len(rs) == 0 or len(cs) == 0:
   continue

ref_mean  = float(np.mean(rs))
cur_mean  = float(cs[0])
sales_chg = (cur_mean - ref_mean) / (abs(ref_mean) + EPS)

# PSI approximated via stored quantiles
rq  = sales_df[(sales_df['run_date'].isin(ref_runs)) &
             (sales_df['cat_id'] == cat)]['sales_quantiles'].values
cq  = sales_df[(sales_df['run_date'] == current) &
             (sales_df['cat_id'] == cat)]['sales_quantiles'].values
psi = (compute_psi(np.array(rq[0], dtype=float),
                np.array(cq[0], dtype=float))
      if len(rq) and len(cq) else 0.0)

# PI width change (model uncertainty indicator)
rp      = pi_df[(pi_df['run_date'].isin(ref_runs)) &
              (pi_df['cat_id'] == cat)]['mean_pi_width'].values
cp_     = pi_df[(pi_df['run_date'] == current) &
              (pi_df['cat_id'] == cat)]['mean_pi_width'].values
ref_pi  = float(np.mean(rp)) if len(rp) else 0.0
cur_pi  = float(cp_[0])      if len(cp_) else 0.0
pi_chg  = (cur_pi - ref_pi) / (abs(ref_pi) + EPS)
```

```
        reasons = []
        if psi > PSI_THRESHOLD:                reasons.append(f'PSI={psi:.3f}')
        if abs(sales_chg) > SALES_CHANGE_THRESH: reasons.append(f'sales={sales_chg*100:+.1f}%')
        if pi_chg > PI_WIDTH_THRESH:           reasons.append(f'pi_w={pi_chg*100:+.1f}%')

        rows.append({
            'run_date':             current,
            'cat_id':               cat,
            'psi_score':            round(psi, 4),
            'mean_sales_ref':       round(ref_mean, 4),
            'mean_sales_cur':       round(cur_mean, 4),
            'mean_sales_change_pct': round(sales_chg * 100, 2),
            'pi_width_ref':         round(ref_pi, 4),
            'pi_width_cur':         round(cur_pi, 4),
            'pi_width_change_pct':  round(pi_chg * 100, 2),
            'is_drifted':           len(reasons) > 0,
            'drift_reason':         '; '.join(reasons) if reasons else 'stable',
        })
    return pd.DataFrame(rows)
```

А.14 Replay-файнтюнінг при детектованому дрейфі (хмарний пайплайн)

Наведено навчальний цикл методу Experience Replay у хмарному пайплайні. Кожен крок оптимізації об'єднує батч нових даних (задрифтовані категорії) з батчем з буфера відтворення (стабільні категорії), запобігаючи катастрофічному забуванню.

```
def replay_finetune(model_nn: nn.Module,
                    new_dl: DataLoader,
                    rep_dl: DataLoader) -> list:
    """
    Replay fine-tuning:
        Each step concatenates a batch from new data (drifted categories)
        with a batch from the replay buffer (stable categories).
    """
    criterion = QuantileLoss(QUANTILES).to(DEVICE)
    optimizer = torch.optim.AdamW(model_nn.parameters(), lr=FT_LR, weight_decay=1e-5)
    scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(
        optimizer, T_max=FT_EPOCHS, eta_min=FT_LR / 10
    )
    replay = itertools.cycle(rep_dl)
    losses = []

    for epoch in range(1, FT_EPOCHS + 1):
        model_nn.train()
        smooth = None

        for b_new in new_dl:
            b_rep = next(replay)
            # Merge new data and replay buffer into one batch
            batch = {
                k: torch.cat([b_new[k].to(DEVICE), b_rep[k].to(DEVICE)], dim=0)
                for k in b_new
            }
            optimizer.zero_grad(set_to_none=True)
```

```
            out  = model_nn(TFTLightning._inputs(batch))
            loss = criterion(out['predicted_quantiles'], batch['target'])
            if torch.isfinite(loss):
                loss.backward()
                nn.utils.clip_grad_norm_(model_nn.parameters(), 1.0)
                optimizer.step()
                smooth = (loss.item() if smooth is None
                          else 0.95 * smooth + 0.05 * loss.item())

        scheduler.step()
        losses.append(smooth or 0.0)
        log.info(f'  Epoch {epoch}/{FT_EPOCHS}  loss={smooth:.4f}')

    return losses
```

Додаток Б. Параметри моделі та експериментів

Б.1 Параметри препроцесингу даних

Таблиця Б.1 — Параметри формування послідовностей

Параметр	Значення	Опис
LOOKBACK	90	Довжина вхідного контексту (днів)
FORECAST_HORIZON	28	Горизонт прогнозування (днів)
SEQ_LENGTH	118	Загальна довжина послідовності (90 + 28)
STRIDE	7	Крок зсуву ковзного вікна (днів)
TRAIN_RATIO	0.80	Частка послідовностей у тренувальній вибірці
VAL_RATIO	0.10	Частка послідовностей у валідаційній вибірці
Test ratio	0.10	Частка послідовностей у тестовій вибірці
HELD_OUT_ITEM_RATIO	0.05	Частка товарів, виокремлених як «нові»
NEW_ITEM_SPLIT_FRAC	0.50	Частка послідовностей нового товару - fine-tune split

Таблиця Б.2 — Ознаки моделі

Статичні категоріальні ознаки (5)

Ознака	Тип	Опис
item_num_id	Int16	Числовий ідентифікатор товару
dept_num_id	Int8	Числовий ідентифікатор відділу
cat_num_id	Int8	Числовий ідентифікатор категорії
store_num_id	Int8	Числовий ідентифікатор магазину
state_num_id	Int8	Числовий ідентифікатор штату

Динамічні числові ознаки (18)

Ознака	**Тип**	**Опис**
sell_price	Float32	Ціна продажу
is_in_assortment	Int8	Товар присутній в асортименті
is_sporting	Int8	Спортивна подія
is_cultural	Int8	Культурна подія
is_national	Int8	Національне свято
is_religious	Int8	Релігійне свято
is_event	Int8	Будь-яка подія (event_name_1)
is_double_event	Int8	Два одночасних события
snap_flag	Int8	SNAP-програма у відповідному штаті
month_sin, month_cos	Float32	Циклічне кодування місяця
dow_sin, dow_cos	Float32	Циклічне кодування дня тижня
doy_sin, doy_cos	Float32	Циклічне кодування дня року
is_weekend	Int8	Вихідний день
is_month_start	Int8	Перший тиждень місяця
is_month_end	Int8	Останній тиждень місяця

Вхідний вектор часового ряду TFT: [log1p(sales)] + [18 dynamic] = **19 каналів**.

Б.2 Архітектура TFT

Таблиця Б.3 — Гіперпараметри архітектури

Параметр	**Значення**	**Опис**
state_size	64	Розмірихованого стану (всі проміжні шари)
lstm_layers	2	Кількість шарів LSTM у Sequence-to-Sequence блоці
attention_heads	4	Кількість голів механізму інтерпретованої уваги
dropout	0.10	Ймовірність dropout
num_static_categorical	5	Кількість статичних категоріальних ознак
num_historical_numeric	19	Каналів у вхідній послідовності (lookback)
num_future_numeric	18	Каналів у майбутній послідовності (horizon)
output_quantiles	0.10, 0.25, 0.50, 0.75, 0.90	Квантилі прогнозу
Загальна кількість параметрів	~1.44 M	—

Б.3 Параметри аугментацій

Таблиця Б.4 — Аугментація холодного старту

Параметр	**Значення**	**Опис**
cold_start_prob	0.15	Ймовірність застосування аугментації до батчу
cold_start_full_prob	0.50	Умовна ймовірність повного обнуління (vs. часткового)
Мінімальна кількість обнулених днів	L/4 = 22	При частковому обнуленні
Максимальна кількість обнулених днів	L = 90	При частковому обнуленні

Таблиця Б.5 — Аугментація дрейфу даних

Параметр	**Значення**	**Опис**
drift_sample_prob	1.0	Частка семплів батчу, що отримують дрифтовану копію
demand_scale_range	(0.70, 1.50)	Діапазон масштабування попиту (рівномірний розподіл)
price_scale_range	(0.80, 1.30)	Діапазон масштабування ціни (рівномірний розподіл)
drift_start	Uniform[0, L)	Точка початку дрейфу у lookback вікні (незалежна для кожного семплу)
Горизонт прогнозу	Завжди дрифтований	Незалежно від значення drift_start

Б.4 Параметри оцінювання та навчання методів континуального навчання

Таблиця Б.6 — Конфігурація синтетичного дрейфу для CL-експерименту

Параметр	**Значення**	**Опис**
DRIFT_DEMAND	1.60	Множник попиту (поза діапазоном тренувальної аугментації 0.7–1.5)
DRIFT_PRICE	1.25	Множник ціни
DRIFT_ITEM_FRAC	0.40	Частка тестових послідовностей, що отримують дрейф
DRIFT_FT_FRAC	0.50	З дрифтованих: частка для файнтюнінгу (решта — для eval)
drift_start (eval)	60-й день	Дрейф починається з останньої третини lookback

Таблиця Б.7 — Гіперпараметри методів файнтюнінгу (CL-експеримент)

Параметр	**Vanilla**	**Replay**	**EWC**	**LoRA**
LR	3×10^{-4}	3×10^{-4}	2×10^{-4}	5×10^{-4}
Batch size	2 048	2 048	2 048	2 048
Replay ratio	—	0.20	—	—
Replay buffer frac	—	0.05	—	—
EWC λ	—	—	5×10^{5}	—
Fisher batches	—	—	200	—
Fisher clamp	—	—	10^{4}	—
LoRA rank r	—	—	—	8
LoRA alpha α	—	—	—	16
LoRA dropout	—	—	—	0.10
Scheduler	CosineAnnealing	CosineAnnealing	CosineAnnealing	CosineAnnealing
η_min	LR / 10	LR / 10	LR / 10	LR / 10

Б.5 Параметри хмарного пайплайну файнтюнінгу

Таблиця Б.8 — Параметри виробничого Replay fine-tuning

Параметр	**Значення**	**Опис**
FT_LR	3×10^{-4}	Швидкість навчання
FT_BATCH_SIZE	512	Загальний розмір батчу
REPLAY_RATIO	0.20	Частка replay у кожному батчі
REPLAY_FRAC	0.05	Частка стабільних послідовностей у replay-буфері
New data batch	410	FT_BATCH_SIZE × (1 − REPLAY_RATIO)
Replay batch	102	FT_BATCH_SIZE × REPLAY_RATIO
Scheduler	CosineAnnealingLR	η_min = LR / 10