Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та використання методів
прогнозування в умовах концептуального дрейфу
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-2
Вадим ПРАЦИБУДА
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Костянтин ПЕТРОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Працибуді Вадиму Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та використання методів прогнозування в умовах концептуального дрейфу

затверджена наказом університету від 6 квітня 20 26 р. № 269Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 21 травня 20 26 р.
3. Вихідні дані до роботи Науково-технічні публікації, набір датасетів, документація по роботі із мовою Python, документація по роботі із середовищем PyTorch

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі і постановка задачі дослідження
2) Методи та моделі прогнозування в умовах концептуального дрейфу
3) Розробка комбінованого методу прогнозування
4) Експериментальна перевірка роботи розробленого комбінованого методу

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	06.04.2026	виконано
2	Аналіз предметної галузі	10.04.2026	виконано
3	Огляд методів прогнозування в умовах	15.04.2026	виконано
	концептуального дрейфу		
4	Постановка задачі дослідження та вибір	20.04.2026	виконано
	оптимальних методів аналізу		
5	Розробка комбінованого методу прогнозування	27.04.2026	виконано
	в умовах концептуального дрейфу		
6	Написання пояснювальної записки	01.05.2026	виконано
7	Перевірка на академічний плагіат	06.05.2026	виконано
8	Нормоконтроль	07.05.2026	виконано
9	Підготовка презентації та доповіді	10.05.2026	виконано
10	Попередній захист	15.05.2026	виконано
11	Рецензування	17.05.2026	виконано
12	Захист перед ЕК	21.05.2026	

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Костянтин ПЕТРОВ
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 67 с., 5 рис., 4 табл., 1 дод., 20 джерел.

ГЛИБОКІ МЕРЕЖІ, КОНЦЕПТУАЛЬНИЙ ДРЕЙФ, НЕЙРО-НЕЧІТКІ МОДЕЛІ, ОНЛАЙН НАВЧАННЯ, ПРОГНОЗУВАННЯ.

Об’єкт дослідження – процес прогнозування в умовах концептуального дрейфу.

Предмет дослідження – методи прогнозування в умовах концептуального дрейфу.

Метою кваліфікаційної роботи є аналіз та порівняння сучасних методів рішення задачі прогнозування в умовах концептуального дрейфу, комбінація досліджених методів для вирішення задач різної складності, створення власного комбінованого методу на базі досліджуваних методів.

Дослідження, що проведені в роботі, базуються на використанні методів системного аналізу задач прогнозування зі концептуальним дрейфом, методів машинного навчання, методів математичної статистики, методів математичного та комп’ютерного моделювання для верифікації точності та оцінки ефективності запропонованого підходу.

Результатами дослідження є розробка комбінованого методу для вирішення задачі прогнозування в умовах концептуального дрейфу; синтез та навчання моделей для вирішення задачі; дослідження датасетів та їх властивостей; оцінка працездатності та ефективності розробленого методу.

ABSTRACT

Master’s thesis contains: 67 pp., 5 fig., 4 tabl., 1 ann., 20 references.

CONCEPT DRIFT, DEEP NETWORKS, FORECASTING, NEURO-FUZZY MODELS, ONLINE LEARNING.

The object of the research is the process of forecasting under concept drift.

The subject of the study is forecasting methods under concept drift.

The purpose of this thesis is to analyse and compare modern methods for solving forecasting problems under conditions of conceptual drift, to combine the methods studied to solve problems of varying complexity, and to develop an original combined method based on the methods studied.

The research carried out in this paper is based on the use of methods of systems analysis for forecasting problems involving concept drift, machine learning methods, methods of mathematical statistics, and methods of mathematical and computer modelling to verify the accuracy and assess the effectiveness of the proposed approach.

The results of the study include the development of a method for solving the forecasting problem under conditions of concept drift; the synthesis and training of models to solve the problem; an analysis of datasets and their properties; and an assessment of the performance and effectiveness of the developed method.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 8
1 Аналіз предметної галузі і постановка задачі дослідження 10
1.1 Опис предметної галузі 10
1.2 Огляд сучасних методів прогнозування 12
1.2.1 Фундаментальні моделі 12
1.2.2 Нейромережеві методи 13
1.2.3 Моделі мета-навчання 15
1.2.4 Моделі з навчання з підкріпленням 16
1.2.5 Статистичні методи 18
1.3 Постановка задачі 19
2 Методи та моделі прогнозування в умовах концептуального дрейфу 20
2.1 Статистичні методи та моделі 20
2.2 Методи класичного машинного навчання 22
2.3 Глибинні нейронні мережі 25
2.4 Нейро-нечіткі мережі 29
2.5 Методика проведення дослідження 32
3 Розробка комбінованого методу прогнозування 34
3.1 Опис методу 34
3.2 Інструменти та параметри реалізації 41
4 Експериментальна перевірка роботи розробленого комбінованого методу 44
4.1 Опис умов проведення експерименту 44
4.2 Оцінка отриманих результатів 56
Висновки 62
Перелік джерел посилання 64
Додаток А Відомість кваліфікаційної роботи 67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Концептуальний дрейф – процес зміни залежності між вхідними та вихідними даними;

Трансформер – архітектура машинного навчання представлена у роботі «Attention is all you need» у 2017 році, яка використовується для обробки послідовних даних, використовуючи механізм уваги, замість послідовної рекурентності;

CNN – Convolutional Neural Network – згорткові нейронні мережі;

DLinear – Decomposed Linear – декомпозована лінійна модель;

GRU – Gated Recurrent Unit – вентильний рекурентний вузол;

LoRA – Low-Rank Adaptation – низькоранговий адаптація;

LSTM – Long-Short Term Memory – довга короткочасна пам’ять;

MSE – Mean Squared Error – середньоквадратична помилка;

NFNN – Neo-fuzzy Neural Network – нео-нечітка нейронна мережа;

NLinear – Normalized Linear – нормалізована лінійна модель;

RL – Reinforcement Learning – навчання з підкріпленням;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

TCN – Temporal Convolutional Network – часова згорткова мережа.

ВСТУП

Існує кілька типів задач для успішного вирішення яких можна використовувати машинне навчання. Серед них можна виділити задачу прогнозування. Результат прогнозування дозволяє визначити майбутній стан об’єкта на основі попереднього стану. Через особливо великий об’єм даних існує потреба у розробці методів, які дозволяють швидко та чітко виконувати поставлену задачу прогнозування.

Проте з часом властивості та умови задач можуть змінюватись, внаслідок чого моделі, які базуються на довгому вивченні стохастичних даних, можуть не впоратись з новими задачами та деградувати. Для вирішення подібної проблеми використовують методи онлайн навчання та методи для знаходження концептуальних змін, які дозволяють швидко налаштувати модель під нову проблему.

Є декілька основних різновидів концептуального дрейфу: різкі зміни (abrupt) та поступові зміни з часом (gradual), повільна еволюцій даних (incremental), зміни, які повторюються (recurring). Різні випадки приводять до різної швидкості деградації моделі, що навчена на статичних даних і тому у задачі має використовуватись модель, яка здатна вирішувати певний випадок дрейфу, який може трапитись у середовищі або модель, яка здатна працювати зі всіма видами змін, якщо відсутня інформація про тип нестаціонарності середовища.

Концептуальний дрейф становить серйозну загрозу для моделей у задачах прогнозування. Задачі прогнозування зазвичай виконуються з певною частотою, у випадку якщо проміжок між прогнозами низький, то модель деградує і дає велику кількість хибних прогнозів до виявлення людиною. Тому мають бути створені методи для ідентифікації концептуального дрейфу, інформація з яких обробляється моделлю або ж створена модель, яка здатна поєднати адаптацію до концептуального дрейфу із задачею прогнозування без зовнішніх методів.

В той же час статистичні методи можуть не враховувати складні нелінійні властивості що утворюються між даними, а отже для складних задач потрібно використовувати більш складні методи на основі машинного навчання, зокрема нейронних мереж.

Сучасні методи машинного навчання дозволяють визначати складні властивості у даних, давати швидкий результат, постійно підлаштовуватись під нові дані, обробляти викиди та помилки систем без потреби у втручанні людини.

Галузь розробки сучасних методів обробки часових рядів продовжує широко досліджуватись у науковій спільноті.

Досі не існує методу, який би зміг впоратись зі всіма складними властивостями часових рядів.

Поєднання методів машинного навчання дозволяє нівелювати недоліки індивідуальних моделей, підвищити виразність моделей та поєднати декілька різних підходів до розв’язання задач.

Метою роботи є огляд та порівняння сучасних методів для задач прогнозування в умовах різних видів концептуального дрейфу, знаходження сильних та слабких сторін методів, створення можливих покращень методів для конкретних задач, поєднання декількох можливих методів та створення власного комплексного методу для вирішення задачі.

Результатом дослідження можуть бути використані для вдосконалення вже працюючих базових моделей прогнозування, для адаптації до більш складних змін середовища, створення нових, адаптивних до концептуального дрейфу, моделей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Опис предметної галузі

Прогнозування майбутніх станів середовища є необхідною складовою процесу прийняття рішень, тому використовується у різних галузях: економіці, енергетиці, медицині, логістиці та багатьох інших.

Прогнозування дозволяє убезпечити від прийняття неправильних, суб'єктивних рішень, підвищити рівень керування процесами, створити плани підприємства, визначити майбутні продажі товарів, заощадити коштів витрачені на процеси.

Задача прогнозування представляє із себе знаходження значень $x_{t+1}, x_{t+2}, \dots$ на основі даних $x_1, x_2, \dots, x_t$, тобто знаходження майбутніх станів на основі попередніх (задача екстраполяції даних). Проте існують і задачі прогнозування з декількома вхідними ознаками для знаходження цільової ознаки.

Для автоматизації задач прогнозування застосовують алгоритми машинного навчання та статистичні методи.

Навколишнє середовище може зазнати певних змін з плином часу, що суттєво впливає на функціонування розроблених систем машинного навчання. Зміна середовища приводить до деградації моделі та потреби у перенавчанні моделі із новими даними, проте перенавчання може тривати довго, потребувати великих обчислювальних ресурсів для розрахунків, дозволу від компанії. В той же час робота з великим об'ємам даних у реальному часі не має часу для навчання моделі заново, а отже навчання призведе до зупинки роботи сервісу і відповідних втрат підприємства. Для уникнення стрімкої деградації системи при змінах середовища варто використовувати методи інкрементального навчання. Методи

інкрементального навчання дозволяють вчитись новим даним «на льоту» одночасно із виводом результату.

Концептуальні зміни відповідають за зміни у розподілі P(Y|X), тобто за зміни між зв’язком вихідних та вхідних даних, а отже «логіка» середовища змінилась.

Існує декілька різновидів концептуального дрейфу: різкі зміни (abrupt) та поступові зміни з часом (gradual), повільна еволюцій даних (incremental), зміни, які повторюються (recurring).

Види дрейфу можуть виникати відповідно до властивостей середовища, про які може бути відомо спеціалісту або які можуть залишитись невідомі. В разі невизначеності в можливих типах дрейфу виникає потреба у моделі, яка здатна працювати у всіх різновидах змін.

Статистичні методи й досі виступають в якості моделей для задач прогнозування з урахуванням дрейфу, якщо задачі достатньо прості і немає потреби в зменшенні помилки. Статистичні методи дозволяють швидко та без великих обчислювальних ресурсів зробити прогноз про майбутній стан об’єкту. Моделі на основі машинного навчання зазвичай забезпечують меншу помилку у прогнозуванні, здатні виконувати задачі зі складними властивостями, проте можуть потребувати збільшену кількість обчислювальних ресурсів, більший датасет та потребу у часі в залежності від реалізованої архітектури.

В той же час складні моделі можуть працювати гірше за більш прості моделі на простих задачах, бо можуть ускладнювати задачу.

Отже потрібно підбирати потрібний метод прогнозування в умовах концептуального дрейфу, який залежить від складності поставленого завдання, для збереження коштів, обчислювальних ресурсів, а також підвищення точності прогнозування.

Тема дослідження є достатньо актуальною через велику кількість потоків даних для задач прогнозування. Також оскільки потоки можуть еволюціонувати, то існує потреба у розробці методів, які дозволяють

визначати концептуальний дрейф та підлаштовувати модель під нові властивості середовища.

1.2 Огляд сучасних методів прогнозування

Зі зростанням кількості потоків та джерел інформації існує потреба у створенні універсальних та швидких рішень для задач прогнозування.

Сучасні дослідження у галузі прогнозування та концептуальних дрейфах включають: вивчення моделей для специфічних випадків концептуальних дрейфів, створення загальних фундаментальних моделей, які навчаються на великих датасетах, створення систем зі швидкою адаптацією до нових змін, створення малих ансамблевих моделей та моделей з використанням навчання з підкріпленням, використання статистичних методів для детекції та прогнозування.

1.2.1 Фундаментальні моделі

У статті [1] описується порівняння класичних статистичних, глибоких моделей для задачі прогнозування та фундаментальними моделей на декількох датасетах.

Фундаментальні моделі побудовані на основі архітектури великих мовних моделей і потребують великий розмір датасетів та великої кількості графічних процесорів.

Для задачі прогнозування фундаментальні моделі сприймають вхідні дані, протягом певного вікна, у вигляді набору токенів. Проте оскільки вхідні значення можуть бути представлені безперервно, то вхід має бути нормалізований та квантизований для отримання токену.

Відображення у вигляді токенів для кожного елементу може погано працювати при роботі із часовими рядами та має велику обчислювальну складність. Створення patch, який зберігає певну кількість елементів

послідовності у одному токені, дозволяє зменшити складність, краще узагальнити модель та дані. Архітектура на основі patch називається PatchTST [2].

Великі фундаментальні моделі потребують багато часу для розрахунку, проте забезпечують високий показник адаптації навіть до наборів даних на яких не навчались (zero-shot learning) і здатні працювати у нестаціонарних середовищах, зокрема в умовах концептуальних змін, здатні працювати в умовах складних нелінійних властивостей у даних.

Проте великі моделі не підходять для умов швидкої онлайн обробки даних, у статті наводиться цифра у 6.68 секунд для відповіді моделі Chronos Large та 4.04 для Moirai MoE Base [1]. Фундаментальні моделі краще працюють на окремих датасетах, коли моделі на основі глибокого навчання та статистики видають результати кращі за 50% не залежно від датасету, тобто фундаментальна модель може не впоратись із поставленою задачею, якщо тренувальні датасети не мали схожу структуру до даних робочого середовища. Відповідно до статті фундаментальні моделі можуть суттєвих різнитись при fine-tuning в залежності від специфіки датасету.

Фундаментальні моделі підходять для підприємств, які мають великі обчислювальні ресурси та мають справу з великою кількістю різних датасетів. У випадку, якщо використовується мала кількість датасетів або датасети не мають складних нелінійних властивостей, то краще обрати більш спеціалізовану модель, навчену на потрібному датасеті, незважаючи на універсальність фундаментальних моделей.

1.2.2 Нейромережеві методи

Глибокі нейроні мережі не використовувались при онлайн обробці через потребу у використані оберненої матриці Гессе, яка потребує $O(n^2)$ пам’яті.

Проте сучасні моделі глибоких мереж використовують градієнтні методи зі збільшеними значеннями параметрів кроку. Таким чином можна використовувати звичайний стохастичний градієнтний спуск з високим значенням моментуму або інформаційну матрицю Фішера.

Однак донавчання всіх параметрів глибокої мережі займає багато часу, а отже потрібно використовувати сучасні методи для fine-tuning [3].

Використання Low-Rank Adaptation (LoRA) [4] створює додатковий набір параметрів у просторі низького рангу для запам’ятовування нових даних без забування старих параметрів шару. LoRA забезпечує швидке оновлення параметрів та має низьку просторову складність, що критично для онлайн навчання. Проте LoRA працює з просторами малих розмірностей, а отже може не «вловити» складні властивості нових даних або різкий концептуальний дрейф. Оскільки метод лише адаптує певні властивості до робочої моделі, то сильні зміни предметної галузі здатні сильно вплинути на ефективність робочої моделі, в такому випадку адаптація не може допомогти і потрібно перенавчання або робити fine-tuning моделі.

Метод часткової адаптації параметрів не є заміною онлайн навчання. У випадку якщо потрібно онлайн навчання, то краще обрати більш просту модель, проте яка буде оновлюватись відповідно до нових даних без накопичення помилок. В такий спосіб буде досягнуто поставленої цілі без необхідності у постійному перенавчанні моделі та великої витраті на обчислювальні ресурси.

Звичними моделями прогнозування є DLinear, NLinear [5], які здатні працювати на простих задачах не гірше за трансформери [6] і представляють і себе прості нейронні мережі. Різниця методів в тому, що NLinear нормалізує вхід відносно останнього входу, нормалізовані значення тренуються на лінійному шарі, в результаті відбувається денормалізація, а DLinear знаходить тренд та сезонність за допомогою ковзного середнього та різниці між входом та ковзним середнім, результати навчаються на

лінійному шарі, виходи лінійного шару поєднуються. Отже моделі є неглибокими та доволі простими, проте забезпечують достатній рівень для простих задач, проте не здатні визначати складні нелінійні властивостями між об’єктами середовища.

Оскільки моделі неглибокі, то можуть навчатись в онлайн режимі використовуючи градієнтні методи або методи на основі похідних другого порядку, що дозволяє швидко підлаштовуватись під вхідні дані. Проте ці методи не визначають концептуальних дрейф явно, тому потрібно використовувати різні методи обгортки, які дозволять визначити момент дрейфу.

Високі результати неглибоких моделей при низьких результатах глибоких моделей трансформера показують, що прості задачі не потребують складних і важких систем, які можуть ускладнювати задачу. Оскільки більшість задач є простими, то в більшості випадків краще використовувати простіші і спеціалізовані моделі, які показують гарний результат при мінімальних обчислювальних ресурсів та не потребують великих датасетів, оскільки мережі неглибокі.

1.2.3 Моделі мета-навчання

Іншим різновидом досліджень у галузі прогнозування є створення моделей із мета-навчання. Під мета навчанням розуміють різновид навчання, який має на меті створення моделей, які вчаться як навчатись від досвіду попередніх завдань, таким чином модель здатна швидко адаптуватись до змін.

У роботі [7] представлено розділення концептуального дрейфу на мікро-дрейф та макро-дрейф.

Макро-дрейф представляє із себе зміни стабільний дрейф, який трапляється протягом тривалого часу та періодів, де еволюційний тренд

може бути передбачуваним, наприклад зміна автомобільного трафіку при економічних та демографічних змінах.

Мікро-дрейф представляє із себе тимчасові зміни у датасеті, які відбуваються у контексті макро-дрейфу, що характеризуються нестабільними та непередбачуваними змінами від прикладу до прикладу. Прикладом подібної поведінки є різке збільшення трафіку у негоду або свята.

Для вирішення двох поданих різновидів дрейфу представлено алгоритм мета-навчання для моделей глибокого навчання під назвою LEAF (Learning to Extrapolate and Adjust for Forecasting).

Метод складається із двох модулів: модуля екстраполяції, який відповідає за обробку довгострокових змін, модуля коригування, який відповідає за обробку короткострокових змін.

Метод потребує великого обчислювального ресурсу для тренування, великої різнорідної датасетів із можливих даних. В той же час прогнозування на етапі роботи не потребує великого ресурсу.

Метод краще підходить для завдань, які мають складні структури між даними або для завдань з різними типами складності концептуальних дрейфів. Для простих задач прогнозування із простими дрейфами краще використовувати більш прості методи онлайн обробки з детекторами концептуальних дрейфів. В такий спосіб можна зберегти кошти та час витрачені на знаходження великої кількості датасетів та обчислювальних ресурсів для тренування моделі мета навчання.

1.2.4 Моделі з навчання з підкріпленням

У статті [8] представлена модель для яка є гібридом Online Convex Programming (OCP) та навчання з підкріпленням. OCP використовує Exponentiated Gradient Descent (EGD) і відповідає за довгострокову

стабільність моделі, в той же час алгоритм навчання з підкріпленням дозволяє швидко реагувати на зміни, тобто на концептуальний дрейф.

Особливістю моделі є швидка адаптація до зміни середовища в порівнянні з іншими моделями. Архітектура моделі складається із двох моделей: temporal та cross-variable. Перша модель вчиться працювати на часових рядах, друга модель вчиться результату на основі одного входу. OneNet визначає і вчить ваги між двома моделями. Ваги залежать від втрат EGD та системи навчання з підкріпленням.

Подібна система працює при онлайн обробці, проте потребує навчання двох мереж одночасно, що потребує більшу кількість обчислювальної потужності.

У роботі [9] описується додавання методу навчання з підкріпленням Proximal Policy Optimization (PPO) для моделі Autoregressive moving-average (ARMA), яке дозволяє описувати зміни середовища у часі.

Модель навчання з підкріпленням виступає у ролі «коригувальника», яка дозволяє виправити результат прогнозу ARMA у невеликому діапазоні. Модель навчається підбирати таку дію, яка б дала найменшу помилку. PPO потребує початкового навчання у офлайн режимі та може продовжувати навчатися у онлайн, що забезпечує адаптацію до змін середовища.

Оскільки подібна структура не змінює архітектуру початкової моделі, то структура може бути застосована до статистичних методів для створення адаптації до змін середовища. Проте такий метод буде працювати на моделях, які вже працюють добре і у випадках якщо дрейф не є сильним, оскільки діапазон змін обмежений.

Цей спосіб використання навчання з підкріпленням дозволяє зберегти робочу модель без потреби у навчанні моделі заново, що може потребувати більшу кількість часу та ресурсів компанії. Оскільки використовується неглибока нейронна мережа та статичний метод, то результати отримуються швидко.

Розглянуті методи навчання з підкріпленням є допоміжними для корекції помилок, не потребують значних обчислювальних ресурсів, оскільки складаються з нейронних мереж, зазвичай неглибоких, отже здатні навчатись не дуже складним властивостям середовища, в той же час дозволяють підвищувати точність моделі, в деяких випадках суттєво. Структури можна використовуючи навколо статистичних моделей без модифікації структури працюючих методів, що дозволяє швидко і якісно підвищувати якість отриманих результатів без зупинки роботи процесу на підприємстві.

1.2.5 Статистичні методи

Статистичні методи можуть використовуватись для визначення моменту дрейфу, що можна використовувати з іншими моделями, та для проведення прогнозування.

Класичні методи і досі дозволяють отримати достатні показники точності для задач прогнозування, а тому досі використовуються моделі на основі ARIMA, ES, MA, лінійної регресії. Подібні моделі не дозволяють будувати складні залежності та не можуть визначити нелінійні властивості у даних, проте більшість датасетів не мають складних властивостей, тому статистичні методи досі залишаються популярними.

Існують методи, які є лише обгортками для моделей, які проводять прогнозування.

Метод Bayesian Online Changepoint Detection (BOCPD) дозволяє визначити точку змін на основі ймовірностей. Детектор може використовуватись іншими методами для реакції на концептуальний дрейф.

Метод Adaptive Conformal Prediction (ACP) [10] є модифікацією методу Conformal Predictions для онлайн обробки, дозволяє визначити інтервал в межах якого знаходиться прогноз виданий моделлю. ACP дозволяє підлаштовувати інтервал у онлайн режимі, збільшуючи ширину

інтервалу при концептуальному дрейфі середовища. Проте метод сильно залежить від швидкості навчання, тому краще використовувати методи на основі ансамблів.

Ансамблевий метод на основі ACP називається AgACI. Оскільки метод ансамблевий, то працює на основі декількох моделей, а отже має метод має нижчий показник дисперсії та дозволяє забезпечити високий показник точності інтервалу результату без потреби в складних налаштуваннях гіперпараметру швидкості навчання.

1.3 Постановка задачі

Метою дослідження є огляд та порівняння сучасних методів для задач прогнозування в умовах різних видів концептуального дрейфу, створення можливих покращень методів для конкретних задач, поєднання декількох можливих методів та створення власного комбінованого методу для вирішення задачі.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

− провести аналіз предметної галузі;

− дослідити існуючі методи вирішення задачі;

− розробити власний комбінований метод прогнозування в умовах концептуального дрейфу;

− оцінити працездатність та ефективність використання розробленого методу прогнозування.

Основну увагу в дослідженні приділено розробці комбінованого методу прогнозування в умовах концептуального дрейфу та оцінка його працездатності і ефективності.

2 МЕТОДИ ТА МОДЕЛІ ПРОГНОЗУВАННЯ В УМОВАХ КОНЦЕПТУАЛЬНОГО ДРЕЙФУ

2.1 Статистичні методи та моделі

Одними із найпоширеніших методів обробки даних в задачі прогнозування є статистичні методи.

Статистичні методи дозволяють здійснювати задачу прогнозування з високою точності, з малими витратами обчислювальних ресурсів та високою швидкістю отримання результатів.

Ці методи базуються на використанні на основі простих статистичних функцій та параметрів, а отже не здатні моделювати складні процеси середовища. Більшість статистичних методів здатні працювати лише із лінійними залежностями, проте існують моделі для роботи і з нелінійними структурами.

Проте більшість задач прогнозування мають просту структуру, а отже не потребують використання складних моделей для вирішення задач, через що показник точності моделей залишається достатньо високим на більшості задач.

Статистичні моделі не потребують навчання на прикладах, проте потребують налаштування гіперпараметрів у системі, що може зайняти певний час під час встановлення робочого середовища.

Простота моделей дозволяє впроваджуватись на підприємствах з обмеженими обчислювальними ресурсами або у автономним системах, з обмеженим доступом до електроживлення.

Проте у випадку радикальних змін середовища, моделі можуть не охоплювати нові зміни, що потребує повторного налаштування параметрів моделі. Це може потребувати необхідних дозволів, що може бути складно в залежності від рівня підприємства.

Авторегресійна інтегрована ковзна середня (ARIMA) – це один із статистичних методів, що базується на використанні авторегресії та ковзного середнього. За допомогою авторегресії моделюються майбутні значення на основі попередніх значень. Інтегрованість дозволяє усунути тренд (тобто нестаціонарність) за допомогою диференціювання. Ковзне середнє дозволяє ураховувати попередні помилки моделі. Подібна модель працює з лінійними нестаціонарностями, на одновимірних даних. У випадку складного середовища модель показує гірші результати.

Для вирішення проблем моделі можна використовувати модифікації методу ARIMA.

SARIMA – модифікація ARIMA, яка враховує сезонність.

ARIMAX – модифікація ARIMA, яка враховує зовнішні фактори.

Vector AutoRegression (VAR) – модель, яка є узагальненням AR, яка використовується при роботі з багатовимірним входом.

Error-Trend-Seasonality (ETS) є іншим типом методів прогнозування, який здатен працювати із сезонністю у даних. Ідеє ETS полягає в експоненційному згладжуванні попередніх спостережень. Тобто ваги для зважування спостережень знижуються експоненційно від найновішого до найстарішого спостережень.

Найпопулярнішими видами нелінійних моделей є порогові.

Threshold Autoregressive Models (TAR) є прикладом порогової моделі. Модель заснована на режимах – регіонах, які розділені параметром порогу, для використання різних авторегресійних виразів.

Self-Exciting Threshold AutoRegressive (SETAR) є модифікацією моделі TAR, яка приймає попередні значення точки для визначення авторегресійного виразу для роботи.

Проте порогові методи критикуються через різку зміну між режимами, для вирішення проблеми було введено клас Smooth Transition AR (STAR) моделей, які забезпечують плавну зміну режимів.

Іншим популярним видом нелінійних моделей є Markov Switching Autoregressive Model (MS-AR). Модель будується із набору авторегресій і перемикається між ними. Перемикання відбувається з певною ймовірністю.

Для роботи із прогнозування також підходять непараметричні моделі, наприклад Kernel Regression, який використовує ядерні функції для визначення виходу моделі.

Більшість з розглянутих вище моделей не пристосовані до концептуального дрейфу, а отже потребують відповідної модифікації.

Для непараметричних моделей можна використовувати ковзне вікно, яке дозволяє проводити оцінку на основі останніх n прикладів, які надійшли в модель. Можна використовувати параметр забування, наприклад експоненційне згладжування, яке використовується забуває старих прикладів з експоненційною швидкістю. У випадку лінійних моделей, як ARIMA, потрібно проводити перенавчання моделі на нових даних.

2.2 Методи класичного машинного навчання

Задача прогнозування в умовах концептуального дрейфу є класичною задачею контрольованого онлайн навчання.

Класичні методи машинного навчання не потребують знань про форму процесу, а вчаться прямо на даних, які надходять та здатні працювати з багатовимірними даними без модифікацій.

Моделі класичного машинного навчання мають більшу гнучкість у функціях, що застосовуються до даних, в тому числі нелінійних, що дозволяє визначати складні патерни в даних та кращу взаємодію між старими та новими даними в порівнянні зі статистичними методами.

Іншою перевагою є легша адаптація до концептуального дрейфу. У моделях онлайн навчання новим прикладам надається більша значимість, ніж старим, що дозволяє продовжувати навчання без потреби ручного

втручання у роботу системи, тоді як більшість статистичних методів потребують створення нової моделі.

Методи класичного машинного навчання є більш стабільними та менш уразливими до викидів у вхідних даних, в порівнянні зі статистичними методами.

Серед моделей класичного машинного навчання для прогнозування в умовах концептуального дрейфу можна виділити декілька типів моделей: моделі на основі дерев, ймовірнісні моделі, моделі на основі екземплярів, лінійні моделі, градієнт бустинг.

Дерева Гефдінга – різновид дерев рішень для інкрементального навчання. Замість зберігання всіх прикладів метод зберігає лише статистичні дані про приклади, які поступили на вхід. Через певну кількість входів відбувається перевірка на розділення дерева рішень. Подібна модель асимптотично збігається до класичного дерева рішень при статичних даних. Для вирішення концептуального дрейфу використовуються модифікація Hoeffding Adaptive Tree, який використовує у своїй роботі алгоритм детекції дрейфу. Методи дерев рішень є методами, що підтримують інтерпретованість вихідних даних, тобто підтримують можливість визначати причину прийняття рішення.

Адаптивний випадковий ліс – ансамблевий метод на основі дерев Гефдінга, де кожне дерево має власний метод детекції дрейфу. У випадку дрейфу дерево замінюється на нове. Ансамблеві методи дозволяють зменшити дисперсію точності, тому подібний метод буде більш стабільним за просте дерево Гефдінга. Дерева рішень мають властивість пояснювати власні рішення, а отже у випадку дрейфу можна побачити властивості, які були змінені між двома станами середовища.

Онлайн метод k-найближчих сусідів (k-NN) – це метод, який використовує вікно певного розміру для збереження вхідних прикладів, тобто старі приклади видаляються з надходженням нових точок. Для адаптації до дрейфу потрібно використовувати окремий метод детекції.

Метод є чутливим до прокляття розмірності, тому у випадку великих розмірностей краще використовувати інший метод або використовувати методи зменшення ознак, адаптації ваг.

Онлайн метод наївного баєсового класифікатора є достатньо простою модифікацією. Для онлайн методу потрібно лише зберігати кількість прикладів класів, кількість ознак та значення середнього, дисперсії для неперервних змінних. Оскільки кожна зі збережених ознак може бути оновлена інкрементально, то подібний метод має високу швидкість навчання. Проте варто пам’ятати, що в наївному баєсовому класифікаторі використовується формула Баєса для незалежних змінних, а отже вхідні дані мають мати припущення про незалежність. Проте в задачі прогнозування вхідні дані будуть мати високі значення кореляції, що означає нижчу ефективність моделі баєсового класифікатора. Для вирішення проблеми концептуального дрейфу можна використовувати ковзне вікно або фактор забування.

До онлайн методів можна віднести методи лінійної та логістичної регресії, що використовують для навчання стохастичний градієнтний спуск або рекурсивний метод найменших квадратів.. Стохастичний градієнтний спуск може бути повільним методом навчання у випадку дрейфу, оскільки використовує похідні першого порядку, рекурсивний метод найменших квадратів адаптується до дрейфу швидше, проте є менш стійким до викидів у даних. Фактор забування можна налаштувати за допомогою зовнішніх методів детекції дрейфу.

Онлайн градієнт бустинг працює на основі градієнту на кожному прикладі, на основі якого оновлюються параметри слабких дерев та ансамблю. Для роботи із концептуальним дрейфом можна використовувати зовнішні методи детекції. Метод на слабких деревах є нестабільним у нестатичному середовиші, має високу чутливість до шумів та має накопичення помилок, оскільки ранні моделі не можуть бути виправлені.

2.3 Глибинні нейронні мережі

Класичні моделі машинного навчання можуть не дати потрібний результат у випадку складних зв'язків та властивостей у середовищі, для вирішення складних задач можна використовувати методи на основі нейронних мереж.

Для задач прогнозування використовуються часові моделі з певними модифікаціями.

Популярними рішеннями є моделі на основі рекурентних нейронних мереж (RNN). Проста модель RNN не має довгої пам'яті, тому є менш використовуваним рішенням.

Модель Long-Short Term Memory (LSTM) є кращим варіантом для послідовних обробок ніж RNN, проте потребує більше пам'яті, бо має систему із 3-х вентилів (вентиль забування, вхідний вентиль, вихідний вентиль). Модель може справлятись з довгостроковими залежностями, тобто пам'ятає інформацію довгих послідовностей. Комірки пам'яті дозволяють запам'ятовувати та зберігати важливу інформацію. Вхідний вентиль – контролює інформацію, яка має бути додана до комірки пам'яті. Вентиль забування – визначає інформацію, яка має бути прибрана із комірки пам'яті. Вихідний вентиль – відповідає за інформацію, яка має бути виведена коміркою пам'яті.

Gated recurrent unit (GRU) полегшена модель LSTM, яка має 2 вентилі (вентиль оновлення та вентиль скиду), що дозволяє навчатись на меншій кількості параметрів, проте з можливою втратою точності. Модель здатна ефективно справлятись з послідовностями та часовими рядами. Вентиль оновлення – відповідає за інформацію, яка має бути збережена із прихованого стану для наступного часового кроку. Вентиль скиду – визначає, яка частина минулих прихованих статів має бути забута.

Серед згорткових нейронних мереж (CNN) популярним рішенням є Temporal Convolutional Network (TCN), яка відповідно до дослідження

дозволяє отримувати кращі результати в порівнянні з рекурентними нейронними мережами на послідовних задачах [11].

Модель працює на основі Causal Convolution та Dilated Convolution та Residual блоків.

Causal Convolution використовує лише інформацію з попередніх та теперішніх даних, оскільки в часовому ряді відсутня інформація про майбутні стани середовища.

Dilated Convolution використовує вхідні дані часових послідовностей з певним пропуском.

Residual блок з'єднує вході різних блоків через shortcut, що дозволяє ефективніше передавати дані між різними блоками без втрат. Кожен residual block мережі складається із декількох шарів Dilated Convolution.

Моделі на основі трансформерів є найпопулярнішими для вирішення більшості типів задач машинного навчання. Моделі на основі трансформерів часто використовуються для задач прогнозування, проте потребують великих датасетів для навчання.

Основна проблема self-attention у потребі $O(n^2)$ пам'яті та часу для роботи.

Модель Informer була створена для вирішення проблеми [12]. Модель використовує інструмент ProbSparse Attention, який працює на ідею того, що більшість ваг у attention будуть близькі до нуля, а отже вибір лише найкращих запитів дозволить пришвидшити роботу моделі. Self-attention Distilling використовує Conv1D та stride після кожного encoder блоку для зменшення надлишкових ознак. Архітектура корисна при наявності великих часових послідовностей для обробки та наявності складних властивостей у даних для аналізу.

PatchTST розбиває вхідний часовий ряд на послідовність патчів, як у моделі Visual Transformer. Патчі представляються у вигляді linear embedding, операції проводяться над патчами, а не окремими точками. В такий спосіб складність моделі $O(N^2)$, проте N, в цьому разі, є кількістю

патчів, а не кількістю точок, що дозволяє суттєво збільшити швидкість прогнозування.

DLinear та NLinear є лінійними неглибокими моделями для прогнозування з особливою попередньою обробкою. Відповідно до результатів дослідження, [6] моделі здатні дати результати не сильно гірші за багатошарові трансформери на простих задачах, що має вказувати на те, що спеціалізовані архітектури все ще можуть працювати краще за створення комплексних архітектур з великими датасетами.

Модель DLinear працює на основі ідеї, що часовий ряд може бути декомпозований на тренд та сезонність. Тренд часового ряду можна визначити через ковзне середнє, тоді сезонність має бути різницею вхідного значення та тренду. Тренд та сезонність мають власні лінійні шари для обробки. Сума виходів лінійних шарів формує кінцевих результат прогнозування.

Модель NLinear працює на основі нормалізації, шляхом знаходження різниці останнього значення спостереження до значень входу, перед процесом навчання лінійного шару. До виходу лінійного шару додається значення останнього значення спостереження, для проведення денормалізації. Архітектура дозволяє зменшити вплив дрейфу на процес навчання та прогнозування моделі, тим самим стабілізуючи модель.

Модель Fast-Slow Network (FSNet) працює на розділення часового ряду на дві компоненти: швидку та повільну [13]. Швидка компонента реагує на короткострокові зміни, повільна компонента відповідає за вивчення структури часового ряду.

Отже за реакцію на дрейф відповідає швидка компонента. Швидка компонента має бути неглибокою, для онлайн навчання.

Повільні та швидкі компоненти мережі можуть мати різні швидкості навчання, оптимізатори та архітектури.

Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) [14] використовує декілька блоків шарів прямого

поширення, кожен блок видає минуле значення x та значення y, в кожний наступний шар передається значення різниці вхід та виходу x, тобто кожен наступний блок тренується на залишку попереднього блоку. В кінці розраховується сума для всіх значень y для кожного блоку.

Архітектура N-BEATS є глибокою та може мати велику кількість параметрів, у результаті швидке онлайн навчання подібної моделі може бути проблемним.

Проте онлайн навчання не дозволяє навчатись у декілька ітерацій для запам’ятовування даних. В такому випадку потрібно використовувати методи, які здатні адаптувати модель за один прохід даних.

Чисті методи похідних другого порядку дозволяють моделі швидко зійтись, проте можуть не підходити до онлайн навчання, де дані можуть змінюватись з часом. Моделі з похідними другого порядку є чутливими до шуму, навіть один викид може сильно змінити значення параметрів мережі. Найбільшою ж проблемою методів похідних другого порядку є використання матриці Гессе, яка потребує квадратичної кількості параметрів, що потребує великих витрат пам’яті у великих моделей. Додатково, обертання матриці Гессе потребує багато часу, що може бути критично в онлайн моделях. Моделі другого порядку працюють на основі припущення про стаціонарність наданих даних, що не відповідає дійсності при онлайн обробці нових даних.

Для глибоких мереж використовують методи оптимізації похідних першого порядку проте з адаптивними значеннями параметрів. До адаптивних значень параметрів належать: високі значення швидкості навчання, weight decay, різні значення параметрів оптимізаторів. Під високими значення швидкості навчання приблизно маються на увазі значення 10^{-1} або 10^{-2}. Агресивний weight decay дозволяє тримати значення параметрів низькими, в такому випадку модель буде більш адаптивною та зможе змінитись у випадку зміни у потоку даних. Модифікації параметрів оптимізатора Adam може включати зміну значення параметру моменту.

Проте використання агресивних значень параметрів може призвести до нестабільності моделі.

Оптимізатори Adagrad, Adam мають обмеження щодо строку процесу навчання. Adam надає перевагу оновлення на початку, з повільним зменшенням ефективності тренування параметрів, що вже були використані. Adagrad накопичує помилки без забування, а отже в якийсь момент модель перестане навчатись. Отже подібні оптимізатори варто оновлювати при онлайн навчанні, бо процес онлайн навчання не сходитись.

Збільшення глибини негативно впиває на оновлені параметрів моделі. Проходження по кожному новому шару моделі збільшує час обробки, що може бути критичним, якщо потрібна моментальна обробка. Велика кількість параметрів на глибоких шарах має повільне оновлення, а отже моделі не здатні швидко реагувати на дрейфи. Глибинні нейронні мережі потребують великих датасетів та довгого навчання. У онлайн обробці приклади подаються один раз, а отже модель не зможе добре запам’ятати інформацію та буде катастрофічно забувати отриманні дані.

Оскільки навчання одному прикладу формує нестабільний градієнт, то використовувати буферу із декількох прикладів дозволить збільшити точність моделі, за рахунок швидкості навчання моделі.

2.4 Нейро-нечіткі мережі

Для задач прогнозування можна використовувати мережі на основі поєднання лінійних шарів із нечіткою логікою. Додавання нечіткої логіки вирішує проблему інтерпретованості вхідних даних. Подібних нейро-нечіткі мережі використовують алгоритм нечікого виведення Такагі-Сугено-Канга (ТСК) в процесі своєї роботи.

Однією із таких моделей є система evolving Takagi Sugeno (eTS), яка дозволяє навчатись даних у онлайн режимі без потреби у великих датасетах офлайн навчання [15].

Еволюційна складова моделі дозволяю змінювати власну архітектуру в залежності від актуальності даних та складності вхідної задачі. Еволюційна складова дозволяє збільшити тривалість життєвого циклу використовуваної моделі.

Еволюція з постійними додаваннями та видаленням кластерів може викликати нестабільність мережі, що може призвести до підвищення рівня похибки.

Мережа працює з використанням eClustering для створення кластерів, які дозволяють визначити функції належності вхідної точки. Комбінації функцій належності з лінійним шаром формують виходи мережі [16].

Модель здатна працювати з будь-якими значеннями кореляції між вхідними даними.

Модель не потребує ніяких додаткових модифікацій для роботи із задачею прогнозування.

Більш простою моделлю є мережа на основі нео-нечіткого нейрона [17].

Нео-нечітка нейронна мережа – представляє із себе нелінійну модель на основі системи ТСК нульового порядку з множинним входом та єдиним виходом. Оскільки модель реалізовує ТСК нульового порядку, то вихід формується на основі суми нелінійних трансформацій вхідних даних, а отже модель може ефективно працювати із прогнозуванням, якщо зробити припущення про слабку та просту кореляцію вхідних даних. Через низьку глибину модель може з вищою ефективністю використовувати методи оптимізації та швидше зійтись. Модель на основі нечітких модель мають властивості інтерпретованості, що дозволяє визначити вплив кожної ознаки на результат, на відміну від звичайних нейронних мереж. Оскільки нео-нечітка мережа не є глибокою та не має великої кількості нейронів, то модель не потребує великої кількості пам’яті та графічного процесору для розрахунків.

Для адаптації до довгострокових змін зовнішнього середовища і дрейфів можна використовувати еволюційну нео-нечітку нейронну мережу [18].

Еволюційна модель починає з малої кількості функцій належності для входу з поступовим збільшенням або зменшенням кількості функцій належності для кожного параметру входу в залежності від властивостей вхідних даних. Взаємонезалежність входів один між одним дозволяє налаштовувати лінійну зміну кількості параметрів при еволюції замість експоненційної.

Додавання функції належності відбувається за ознакою високого значення помилки або за максимальне значення належності нижче певного порогу. Додана функція належності має значення вхідної точки в якості центру, значення дисперсії та ширини, у випадку використання значення дорівнює зваженій відстані до найближчого центру.

В якості функцій належності можна використовувати трикутні функції, функцію Гауса, B-сплайни.

Видалення функцій належності відбувається з використанням часу життя функції належності та функції корисності/активності, яка підраховуються ступінь активації функцій належності.

Значення центрів та дисперсії можуть бути змінними відповідно до вхідних даних, центри у подібній схемі намагаються наблизитись до нової вхідної точки, разом зі змінною ширини кластеру (дисперсії).

Якщо існують більш складні залежності між вхідними даними, то можна використовувати модель Cascade Neo-Fuzzy Neural Network (CNFNN), яка дозволяє проводити обробку у декілька шарів нео-нечітких нейронів [19].

Кожен шар має всі властивості звичайної нео-нечіткої мережі, проте архітектура має особливість.

Особливістю каскадної моделі є використання значення виходу попереднього шару в якості додаткового одного входу. Оскільки

додатковий вхід залежить від попередніх входів, то подібна структура дозволяє встановити непряму нелінійну залежність між вхідними ознаками, таким чином прибирається один із недоліків одношарової нео-нечіткої моделі, а саме не можливість встановлення складної взаємодії між вхідними даними.

Модель не потребує використання методу зворотного поширення помилки для навчання незважаючи на багатошарову структуру. Навчання виконується послідовно для кожного із шарів.

При роботі з моделлю варто пам’ятати, що на кожному шарі модель має проходити через нелійнійний перетворювач, що може бути дорогою за часом операцією, що може приводити до затримок у моментальних задачах.

Каскадна модель може бути еволюційною [20].

Еволюція більшої кількості шарів може призвести до збільшення нестабільності, через використання виходів попередніх шарів, які вже можуть бути зашумленими і породжувати ще більше помилок в наступних шарах.

Всі розглянуті нечіткі моделі здатні працювати в онлайн задачах прогнозування в умовах концептуального дрейфу. А отже можуть бути використані у експериментах. Проте звичайна одношарова нео-нечітка мережа є достатньо слабкою для використання з іншими моделями без особливого збільшення часу виконання.

2.5 Методика проведення дослідження

В сучасних дослідженнях прогнозування використовують навчання моделей у офлайн режимі. В такий спосіб модель буде добре налаштована під властивості часового потоку, який обробляється.

Концептуальні дрейфи становлять особливу небезпеку при обробці потоків даних, тому дослідження прогнозування в умовах концептуального

дрейфу має включати дослідження поведінки системи під час онлайн обробки потоку даних.

Складні і навчені у офлайн моделі можуть не впоратись із онлайн обробкою та навчанням під час концептуального дрейфу. Проте навчені у офлайн режимі моделі мають властивість забування у випадку переобладнання під онлайн навчання.

Дослідження включають перевірку працездатності та якості розробленого комбінованого методу і порівняння з базовими моделями машинного навчання. Для визначення якості було зроблено перевірку на адаптивність.

Перевірка на адаптивність спрямована на перевірку властивостей систем пристосовуватись до нових даних, експеримент був проведений з виконанням онлайн обробки для комбінованого методу та базових методів.

Проте часто є датасет певного розміру і не використання датасету для навчання зумовлює втрату точності, оскільки офлайн навчання дає змогу моделям навчитись властивостям середовища, в той же час онлайн моделі мають вчитись всій інформації «на ходу» за одну ітерацію. Тому інше дослідження має включати перевірку якості офлайн моделі.

Перевірка офлайн моделі може бути здійснена через розбиття датасетів на тренувальні, тестову та валідаційну. Офлайн моделі вивчають загальні представлення середовища за допомогою тренувальних даних і тестуються за допомогою тестових та валідаційних даних. Гібридну модель було протестовано і здійснено порівняння з базовими офлайн моделями. Дослідження ефективності гібридної моделі проводилась з офлайн навчанням базовою моделлю та онлайн навчанням для більш слабкої моделі. В такому випадку базова модель не забуває інформацію про часовий потік, а слабкіша модель відповідає за адаптивність часового потоку.

За допомогою додавання штучних дрейфів можна визначити місця змін без застосування методів детекції дрейфів. Була здійснена візуальна перевірка реакцій розроблених систем на дрейфи.

3 РОЗРОБКА КОМБІНОВАНОГО МЕТОДУ ПРОГНОЗУВАННЯ

3.1 Опис методу

Послідовності у задачах прогнозування можуть бути великих розмірів і звичайні моделі можуть довго сходитись. Модель eTS є повною і самостійною мережею, яка витрачає достатньо часу на прогноз, а тому краще не використовувати в якості моделі адаптера. Каскадна нео-нечітка мережа складається із декількох шарів, що також може вплинути на швидкість роботи адаптеру. Тому для онлайн навчання було обрано одношарову модель нео-нечіткого нейрону.

Головним недоліком нео-нечіткої моделі є неспроможність визначати складні залежності між вхідними даними, проте має високу швидкість адаптації до даних, що робить нео-нечітку мережу варіантом для поєднання зі складною моделлю в якості адаптеру. Обробку складних взаємодій між вхідними даними можна перекласти на базову модель, яка має можливість до аналізу взаємодій, бо побудована на перцептроні. Адаптер відповідає до корекції відповіді моделі до нових вхідних даних, якщо потрібно.

Для використання в якості адаптера можна використовувати навчання моделі на залишках. Адаптер на залишках намагається «виправити» похибку базової моделі, додаючи певне розраховане значення до виходу базової моделі. Для навчання на залишках потрібно: отримати вихід основної моделі, знайти різницю між реальним виходом середовища та виходом основної моделі, передати різницю в якості виходу до нео-нечіткої мережі, скласти виходи базової моделі та нео-нечіткої моделі.

Розроблений комбінований метод складається із базової моделі, яка має вчитись складним зв’язкам і адаптивної моделі, яка має підлаштовується під різкі зміни середовища. Під час виконання прогнозу виходи двох мереж складаються для отримання прогнозу наступного стану часового ряду.

Розроблений комбінований метод виконується у декілька етапів:

− передача вхідних оброблених даних на вхід базовій та адаптивній моделям. В результаті передачі даних отримуються значення цілі для гібридної моделі. Виходом адаптивної моделі будуть прогнозовані залишки, виходом базової моделі будуть прогнози майбутніх станів середовища;

− знаходження реальних залишків для цілі адаптивної моделі шляхом розрахунку різниці реальної цілі та прогнозу базової моделі;

− навчання адаптивної моделі на втратах між результатами залишків базової моделі та прогнозованими залишками. Окреме навчання на залишках працює при онлайн навчанні моделі адаптеру із замороженою базовою моделлю;

− якщо базова модель навчається одночасно із адаптивною моделлю, то базова та адаптивні моделі навчаються через втрату комбінованого методу. Базова модель навчається на різниці цілі прогнозування із залишками, тому варто послаблювати модель залишків, щоб адаптивна модель не забирала на себе роботу базової моделі.

Наявність окремих ознак кожного із параметрів входу дозволяє уникати забування на всіх параметрах моделі одночасно.

Для послаблення моделі на основі залишків можна використовувати показник адаптивного коефіцієнту α, який відповідає за зважування результату нео-нечіткої моделі.

Адаптер тренується на реальних та передбачуваних залишках базової моделі, отже використовуючи формулу (3.1) можна визначити реальні залишки для адаптивної моделі. В такому випадку адаптер виступає послабленим комбінованим методом прогнозування.

$$y_{pred} = y_{base} + \alpha \cdot y_{residual}, \tag{3.1}$$

де y_{pred} – загальний прогноз гібридної моделі;

y_{base} – прогноз базової моделі;

α – адаптивний коефіцієнт;

$y_{residual}$ – прогноз моделі адаптера.

Для проведення навчання потрібно визначити втрати гібридної моделі. Загальні втрати моделі визначаються за функцією оцінки втрат з реальними даними. В якості функції оцінки втрат використовувалась середньоквадратична похибка, оцінка втрат знаходиться за формулою (3.2):

$$loss_{pred} = MSE(y_{true}, y_{pred}), \tag{3.2}$$

де $loss_{pred}$ – загальні втрати гібридної моделі;

$MSE(y_{true}, y_{pred})$ – середньоквадратична похибка;

y_{true} – реальний наступний стан середовища.

Архітектура може застосовуватись у декількох режимах роботи гібридної моделі:

– базова та адаптер моделі навчається на датасеті. Базова модель перестає навчатись у онлайн режимі;

– базова та адаптер моделі обидві продовжують навчатись у онлайн режимі.

У першому випадку базова модель буде стабільна до часового ряду, який розглядається. Адаптеру легше адаптуватись до стабільної системи, оновлюючись лише у випадку незвичної поведінки середовища, тобто концептуального дрейфу. Такий варіант дозволяє убезпечити модель від забування старих даних, які можуть мати актуальність і після довгого часу життя системи. Використання адаптеру у онлайн дозволить продовжити цикл життя моделі та дати більше часу на оновлення застарілої моделі під час деградації. В цьому варіанті модель нео-нечіткої мережі працює в якості адаптера залишків базової моделі.

У другому випадку адаптер підхоплює зміни середовища швидше за базову модель, якщо базова модель не розроблена для онлайн обробки. У випадку, якщо базова модель є онлайн моделлю, то задача перетворюється

у ансамбль з двох методів, результати яких складаються. Нестабільність базової моделі може привести до нестабільності у моделі адаптеру, тому краще спочатку «розігрівати» базову модель, що дозволить адаптеру працювати на моделі, яка вже має знання про середовище. Базова модель буде мати високих показник забування та низький показник запам’ятовування через онлайн навчання.

У випадку заморозки базової моделі варто використовувати іншу функцію втрат, яка навчається не робити прогноз загального виходу середовища, а зосереджується на похибках отриманих базовою моделлю. Для цього варто розрахувати прогнозовані помилки моделі адаптера за формулою (3.3). В такому разі адаптивна модель стає чистим адаптером залишків для офлайн моделі:

$$y_{residual}^{true} = y_{true} - y_{base}, \tag{3.3}$$

де $y_{residual}^{true}$ – реальне значення залишків.

Відповідно до формули (3.4) можна знайти окрему функцію втрат моделі адаптера.

$$loss_{residual} = MSE(y_{residual}^{true}, \alpha \cdot y_{residual}), \tag{3.4}$$

де $loss_{residual}$ – втрати адаптивної моделі.

Адаптер дає змогу зберегти значення ефективності високим, при змінах у середовищі, які невідомі базовій моделі, проте з проходженням часу ефективність базової моделі спадає. Отже, за час доки модель нео-нечіткої мережі використовується в якості адаптера, можна перевчити базову модель на нових даних. Перевчати модель можна у випадку складного нелінійного дрейфу, якого ніколи не бачила базова модель та який не може бути визначений нео-нечіткою мережею через низьку виразність мережі або нові складні нелінійні зв’язки між входами.

Під час навчання на залишкам може виникнути проблема навчання шуму, особливо при високій якості базової моделі, в такому разі процес навчання моделі на основі залишків має бути пропущений. Якісною прогнозом можна вважати прогноз зі значення 10^{-3}, 10^{-4} або краще, подібні похибки вказують на гарні здібності до генералізації базової моделі і не потребують виправлення за допомогою гібридних моделей. Окрім встановлення порогів навчання можна використовувати методи регуляризації для моделі залишків.

Незважаючи на те, що модель адаптера не вчиться на залишках якісних прогнозів все одно можуть виникати проблеми на етапі прогнозування результатів. Не можна очікувати якісного результату від моделі, яка не бачила прикладу, у випадку складних властивостей середовища. У випадку простого середовища адаптивна модель зможе апроксимувати бажаний відсутній вихід.

То ж у випадку якісної базової моделі краще вимкнути адаптер, доки середовище стане нестабільним і адаптер зможе коректно виправляти помилки базової моделі.

Показник еволюції мережі може викликати нестабільність у адаптері, тому краще рідше оновлювати архітектуру мережі або взагалі оновлювати лише у момент концептуального дрейфу за допомогою зовнішнього методу зміни середовища.

Показник еволюції можна відключити, у такому випадку стабільність моделі не буде падати, проте модель не зможе довго працювати з часовими рядами при деградації базової моделі.

Мала кількість параметрів моделі адаптеру забезпечує низький показник втрати часу на обробку результату додаткової моделі.

Інтерпретованість нечіткої моделі може постраждати, оскільки модель буде знати лише ознаки в яких базова модель найбільш невпевнена, замість повної інформації про вихід моделі. Проте інформація про

невпевненість базової моделі може бути корисна при повторному навчанні базової моделі з використанням активного навчання.

Модель адаптера може приймати на вхід багато ознак, проте повертає єдине значення. Отже, для використання у багатокроковій обробці прогнозів потрібно використовувати вихід моделі в якості нового входу моделі. Проте подібний метод є ітеративним, а отже може постраждати час виконання адаптації. Використання штучних даних моделі в якості входу призводить до накопичення похибки. В такому разі було проведено експерименти лише у вигляді одиничних виходів моделі.

Нео-нечітка модель працює із вхідними даними у певному діапазоні, тому варто використовувати scaler, який здатен привести всі вхідні дані до одного діапазону у онлайн режимі.

У випадку онлайн scaler має навчатись прямо зі входу, у випадку офлайн scaler має навчатись зі всього тренувального набору даних.

При початку онлайн навчання scaler буде без даних для аналізу, тому значення параметрів входу можуть бути нестабільними.

Модель адаптера залишків можна використовувати у працюючій моделі без потреби у зміні параметрів або перенавчання оригінальної моделі. Підприємства можуть мати складні бюрократичні процедури для оновлення базових моделей, в такому випадку можна досягнути потенційного покращення мережі за невелику кількість обчислювальних потужностей та часу обробки. У випадку простих числових рядів модель адаптера зможе швидко пристосуватись до даних без потреби в оновленні базової системи.

Розглянувши сучасні методи рішення онлайн задач прогнозування з концептуальним дрейфом можна прийти висновку про ефективність архітектури NLinear, DLinear, які нормалізують за останнім значенням та роблять декомпозицію входу на сезонність та тренд. LSTM та GRU є стандартними моделями із довготривалою пам’яттю, що може стати корисним у випадку запам’ятовування сезонності у часових рядах задач

прогнозування. Модель TCN дозволяє розглядати локальні патерни для отримання значення прогнозування.

Описані вище методи використовувались під час проведення дослідження.

Модель FSNet має схожу структуру до архітектури, що розглядається у досліджені. Модель тренує швидшу модель на основі залишків, проте швидшою моделлю виступає неглибока мережа на основі перцептрона Розенблатта. FSNet на основі неглибокої мережі не дасть швидшої адаптації за нео-нечіткої мережі.

Модель N-BEATS не використовується, бо модель є глибокою, а отже оновлення параметрів буде доходити довго, внаслідок чого модель не зможе ефективно адаптовується до концептуальних дрейфів, а тому модель не зможе взяти участь у ефективності адаптації до потоку даних у онлайн режимі.

Статистичні моделі мають великий показник похибки на складних датасетах, тому прийнято рішення відмовитись від статистичних методів, бо досліджується лише порівняння якості моделі без урахування часу виконання роботи, а отже статистичні моделі, без сумніву, будуть гіршими за інші методи.

Трансформери можуть мати перевагу у запам’ятовуванні складних патернів, високому показнику паралелізації розрахунків, проте потребують великої кількості пам’яті, часу на навчання та розміру датасету. Незважаючи на усунення недоліку у часі виконання self-attention у моделі Informer, структура все ще має багатошарову архітектуру, що не дозволяє моделям трансформеру навчатись у онлайн режимі, а отже модель не зможуть взяти участь у онлайн дослідження для оцінки ефективності адаптації, тому трансформери не використовувались.

Фундаментальні моделі не були розглянуті через відсутність достатньої кількості обчислювальних ресурсів потрібних для навчання та підтримки подібних великих моделей. Більш того кількість прикладів у

датасеті є несуттєвою для тренування глибоких моделей трансформера. А pretrain моделі вже навчені на обраних датасетах, оскільки обрані датасети є популярними рішеннями для навчання у галузі прогнозування, тому похибки прогнозування базової моделі будуть низькими і дослідження якості гібридної моделі не може бути проведено.

Використання моделей класичного машинного навчання може мати проблеми з налаштуванням обраної модель нео-нечіткої мережі для роботи з моделями, оскільки було оглянуто методи для роботи з класифікацією, проте всі обрані датасети працюють із задачею регресійного аналізу. Єдина оглянута регресійна модель є лінійної регресія, але різні форми лінійної регресії вже використовуються у обраних моделях.

У результаті розробки комбінованого методу вдалось визначити слабкі та сильні сторони адаптера та відповідальність кожної із моделей. Описано інструменти, які можуть використані для навчання моделі в онлайн та офлайн режимах. Обрано архітектури, які були використані під час проведення досліджень. Виведено формули для розрахунку втрат гібридної моделі та окремо моделі на основі залишків виходу.

3.2 Інструменти та параметри реалізації

Реалізація запропонованого комбінованого методу потребує додаткової програмної розробки всіх необхідних методів попередньої обробки, створення моделей, розрахунку статистик, виведення графіків, проведення навчання та тестування.

Для реалізації експериментів було використано мову програмування Python. Мова програмування Python є стандартом у роботі з даними та моделями машинного навчання.

Python має велику підтримку спільноти, що сприяло розбудові великої кількості бібліотек.

Для виконання експериментів було використано наступні бібліотеки:

NumPy – бібліотека для роботи з багатовимірними даними та математичними розрахунками у Python. Бібліотека написана на C, а отже має високу швидкість навчання, проте є не підтримує багатопоточні розрахунки.

Pandas – бібліотека для роботи із малою кількістю багатовимірними даними у одному локальному середовищі у вигляді таблиць. Бібліотека працює повільно при датасетах великих розмірностей. Бібліотека написана на numpy, а отже має всі недоліки та переваги numpy. Бібліотека не підтримує багатопоточної обробки. Бібліотека дозволяє виконувати різні математичні операції на різними колонками та рядками таблиць.

Matplotlib та seaborn – бібліотеки для візуалізацій даних у середовищі Python. Matplotlib працює із багатовимірними даними у numpy, seaborn включає підтримку бібліотеки pandas.

PyTorch – фреймворк машинного навчання для роботи із нейронними мережами. Модель є обгорткою для оптимізованих алгоритмів написаних мовою C. Бібліотека підтримую багатопроцесорну обробки на графічному процесорі на основі CUDA ядер. Бібліотека має автоматичне диференціювання, що дозволяє уникати ручного розрахунку градієнтів. Фреймворк дозволяє впроваджувати весь цикл роботи життя нейронної мережі від навчання до роботи.

Датасети представлені у форматі .csv, для обробки датасетів було використано бібліотеку pandas.

Обробку даних було виконано за допомогою pandas та numpy.

Створення дрейфів було додано використовуючи numpy.

Розбиття на тестову та тренувальну складову набору даних та створення потоку даних було зроблено за допомогою numpy.

Для відображення дрейфів було використано бібліотеку seaborn.

Відображення оригінальних датасетів було зроблено за допомогою matplotlib.

Для створення та навчання моделей машинного навчання було використано Pytorch.

У лістингу 3.1 представлено завантаження основних бібліотек, які були використані під час проведення експерименту.

Лістинг 3.1 – Підключення основних бібліотек

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import torch
import torch.nn as nn
import torch.nn.utils.weight_norm as weight_norm
import seaborn as sns
from tqdm import tqdm
```

Додатково було завантажено бібліотеку tqdm, яка дозволяє відслідковувати процес навчання моделі у вигляді індикатора прогресу у реальному часу.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РОБОТИ РОЗРОБЛЕНОГО КОМБІНОВАНОГО МЕТОДУ

4.1 Опис умов проведення експерименту

Для перевірки роботи комбінованого методу було використано стандартні реальні датасети для прогнозування. Датасети спеціально підібрані для покриття різних типів проблем в часових рядах. Для прогнозування використовується одна (остання) ознака з кожного датасету.

Датасет electricity вимірює споживання електроенергії споживачами, датасет має 321-у ознаку, вимірювання проводились кожну годину. Датасет має сильну сезонність у споживанні електроенергії протягом дня, тижня, сезонів.

Датасети etth1 та etth2 є вимірюваннями двох Electricity Transformer Temperature (ETT) протягом кожної години. Вимірювання включають температури трансформаторів високої та середньої напруги та робочу температуру, яка є основної ціллю для прогнозування.

Датасети ettm1 та ettm2 є вимірюваннями двох Electricity Transformer Temperature (ETT) протягом кожних 15 хвилин.

ETT датасети мають сезонність, тренди, дрейфи.

Датасет exchange вимірює валютні курси для 8-ми валют з вимірюваннями кожну годину. Датасет має концептуальні дрейфи, оскільки вартість валют залежить від політичної, економічної ситуацій у державах власниках валют, що може сильно і різко змінюватись.

Датасет traffic вимірює дорожній трафік за 867-ом сенсорам, вимірювання проводились кожну годину. Тимчасові дрейфи можуть бути утворені пробками та аваріями на дорогах. Навантаженість дороги може залежати від дня тижня, наявності свят.

Датасет weather вимірює температуру, складається із 15-ти ознак, вимірювання відбуваються кожні десять хвилин.

Для перевірки роботи комбінованого методу було використано порівняння базових (чистих) та гібридних моделей.

Було проведено експеримент із онлайн роботи.

Експеримент робиться для визначення адаптивності розробленої гібридної моделі у онлайн обробці даних в порівнянні із базовими моделями.

У експерименті моделі не мали попереднього навчання, а навчались у онлайн режимі. Задачею було визначення наступного елементу послідовності. В якості входу передавалось 64 попередні значення послідовності. В якості метрики втрат використовувалась середньоквадратична похибка (MSE).

Для обробки даних використовувався онлайн scaler, який навчався із додаванням нових даних. Онлайн scaler працює за логікою standart scaler, тобто на основі знаходження різниці з середнім значенням та діленню отриманого значення на дисперсію. Проте оскільки датасет нестаціонарний та онлайн, то використовується алгоритм Уелфорда для інкрементального розрахунку середнього та дисперсії.

При проведені онлайн навчання scaler використовувався до входу даних, що дає нестабільність на початку навчання, тому результати перших десяти ітерацій не враховувались при розрахунку MSE.

При проведені офлайн навчання scaler використовувався на всій тренувальній вибірці, отже вхідні дані, які поступали до моделі, були стабільні.

При поєднанні офлайн та онлайн навчання scaler працював відповідно до режиму навчання, проте нестабільність на початку онлайн навчання відсутня через високу кількість офлайн прикладів.

Для експерименту використовувались всі обрані датасети. В якості базових моделей в процесі порівняння були використані GRU, LSTM, TCN, DLinear, NLinear. В якості адаптера було використано одношарову нео-

нечітку мережу з еволюцією кількості нейронів кожні 1000 кроків і функцією Гауса в якості функції належності.

Значення адаптивного коефіцієнту розроблених систем дорівнює 0.1 для всіх експериментів.

Система пропускає навчання гібридної моделі при значені помилки базової моделі кращої за 10^{-3} для уникнення перенавчання адаптивної моделі на вхідному шумі.

У таблиці 4.1 можна побачити результати для базових (чистих) моделей. Відповідно до даних таблиці найкращі результати показувала модель TCN на всіх датасетах. У датасеті ettm1 найкращий результат ділиться з моделлю NLinear. Відповідно до таблиці LSTM та DLinear мали найгірші показники у онлайн навчанні. LSTM виявився занадто важкою моделлю для онлайн адаптації, через що видно суттєву середньоквадратичну похибку в порівнянні із GRU.

Таблиця 4.1 – Результат експерименту для базових (чистих) моделей

Датасети	Значення MSE для базових моделей				
	GRU	LSTM	TCN	DLinear	NLinear
electricity	0.109	0.145	0.106	0.134	0.111
etth1	0.192	0.272	0.182	0.251	0.187
etth2	0.222	0.318	0.2	0.318	0.203
ettm1	0.066	0.127	0.057	0.113	0.057
ettm2	0.091	0.171	0.062	0.211	0.076
exchange	0.673	1.111	0.423	1.661	0.545
traffic	0.357	0.574	0.32	0.81	0.387
weather	0.203	0.317	0.146	0.487	0.193

У таблиці 4.2 можна побачити результати для моделей із додаванням нео-фаззі адаптера, для прогнозування помилки на залишках базової моделі. Додавання адаптивної моделі до базової у онлайн навчанні дозволило зменшити середньоквадратичну похибку моделей. У таблиці видно, що найкращі результати на датасетах показували моделі NLinear та TCN. Аналізуючи результати експерименту, що представлені в таблиці 4.1 можна

дійти висновку, що гібридна модель мала найбільший ефект на модель NLinear, яка стала мати найкращі результати на 5-ти датасетах. В той же час значення датасету electricity мали гірший результат у гібридній моделі порівняно з базовими моделями.

Таблиця 4.2 – Результат експерименту для гібридних моделей

Датасети	Значення MSE для гібридних моделей				
	GRU	LSTM	TCN	DLinear	NLinear
electricity	0.263	0.398	0.249	0.287	0.239
etth1	0.049	0.075	0.041	0.078	0.039
etth2	0.119	0.178	0.083	0.207	0.078
ettm1	0.064	0.144	0.052	0.107	0.049
ettm2	0.109	0.166	0.046	0.224	0.059
exchange	0.056	0.076	0.033	0.069	0.02
traffic	0.186	0.231	0.181	0.26	0.204
weather	0.175	0.2	0.141	0.325	0.193

Оскільки дві таблиці важко сприймати одночасно через кількість даних, було створено нову таблицю, яка узагальнює найкращі моделі по кожній із таблиць.

Для порівняння отриманих результатів було побудовано таблицю 4.3. Таблиця містить інформацію про найкращий результат серед базових моделей та серед гібридних моделей.

Таблиця 4.3 – Порівняння найкращих результатів гібридних та базових (чистих) моделей

Датасети	Значення MSE для базових моделей	Значення MSE для гібридних моделей
electricity	0.106	0.239
etth1	0.182	0.039
etth2	0.2	0.078
ettm1	0.057	0.049
ettm2	0.062	0.046
exchange	0.423	0.02
traffic	0.32	0.181
weather	0.146	0.141

Відповідно до результатів гібридні моделі показали себе краще на всіх датасетах, окрім electricity. Гібридна модель не принесла суттєвого результату на датасеті weather, ettm1 та ettm2. Гібридна модель дала значне покращення в усіх інших датасетах, в середньому на 10^{-1}, в порівнянні від використання однієї базової моделі. У датасеті exchange з високим показником концептуального дрейфу вдалось значно зменшити похибку розрахунків.

Отже можна зробити висновок про ефективність використання нео-нечіткої мережі в якості адаптера залишків у онлайн навчанні, що дозволяє зберегти вищий показник якості системи, ніж у базових моделей.

Обрані датасети в більшій мірі мають концептуальні дрейфи, проте датасет exchange відрізняється наявною кількістю дрейфів та волатильністю даних.

Було досліджено поведінку моделі при різних типах ручних концептуальних дрейфів: лінійному, нелійнійному, різкому, повільному. Дослідження включають створення обраних типів концептуальних дрейфів під час різного моменту навчання моделі.

Дасасет складається приблизно із 8000 прикладів і дрейфи було створено приблизно на 1000-ому та 6000-ому кроках.

Амплітуди розроблених дрейфів становлять 0.1 для різких лінійних, повільного пізнього для інших дрейфів показник амплітуди становить 0.2.

Було обрано моделі TCN та NLinear, оскільки моделі мають найнижчий показник середньоквадратичної похибки серед онлайн базових моделей.

На рисунку 4.1 зображено результат адаптації моделей базових та гібридних TCN, NLinear до лінійного концептуального дрейфу на в датасеті exchange на початку навчання.

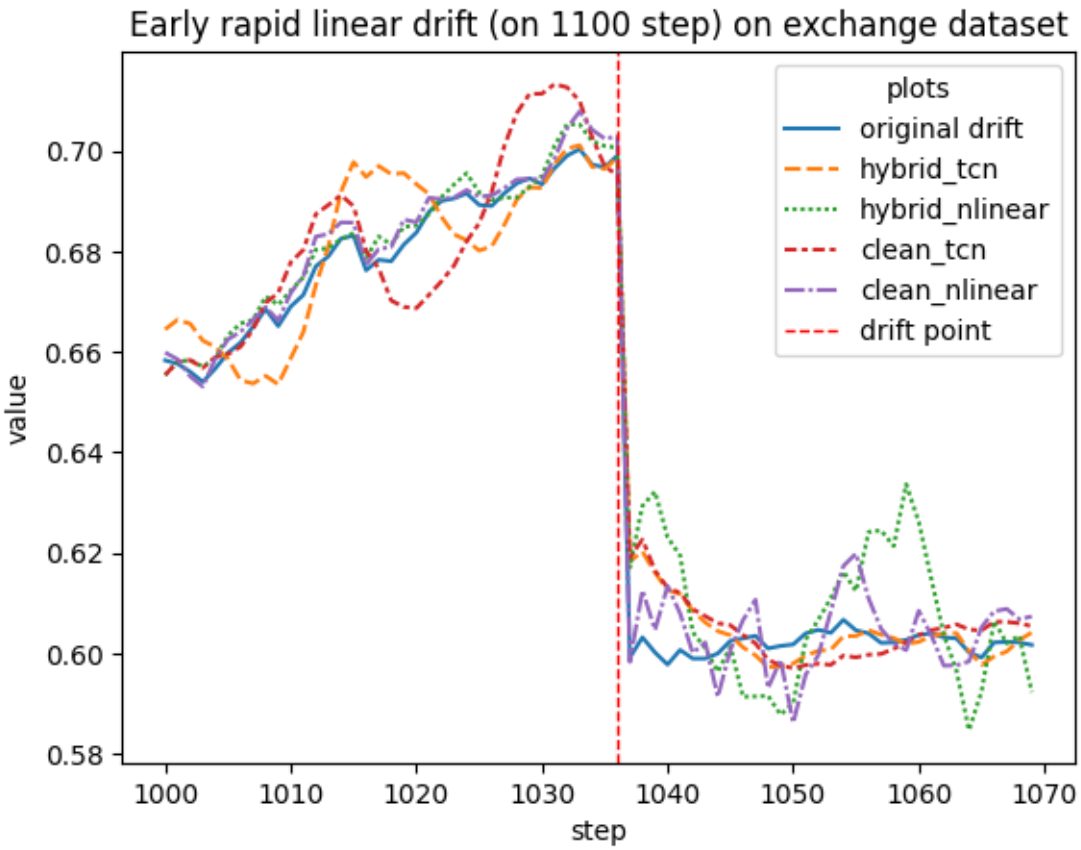


Рисунок 4.1 – Результат прогнозування моделей при дії різкого дрейфу на ранньому етапі онлайн навчання

На рисунку відображена достатньо якісна адаптація до різкого дрейфу, який був доданий до датасету.

Збільшений показник еволюції зумовлює великий показник волатильності гібридних моделей. Одночасне навчання базової та адаптивної моделі сприяє нестабільності отриманих результатів, що можна побачити за межами дрейфа, де гібридні моделі описують структуру проте мають високу амплітуду.

З рисунку видно, що за 100 перших кроків навчання адаптивної моделі NLinear була стабільною, проте дрейф мав сильний вплив на нестабільність моделі. В той же час гібридна модель TCN спочатку була нестабільно, проте стабілізувалась за 100 кроків. У гібридній моделі TCN дрейф не суттєво

підвищив похибку і модель швидко стабілізувалась, порівнюючи із базовою моделлю TCN.

На рисунку 4.2 зображено різкий лінійний дрейф під час пізнього навчання моделі на датасеті. Високі похибки базових моделей спричиняють високі похибки адаптерів, що видно у вигляді сильного падіння точності після моменту дрейфу. В той же час моделі вже навчені до адаптації до простіших форм часового ряду, тобто там де базові моделі мають низький показник похибки.

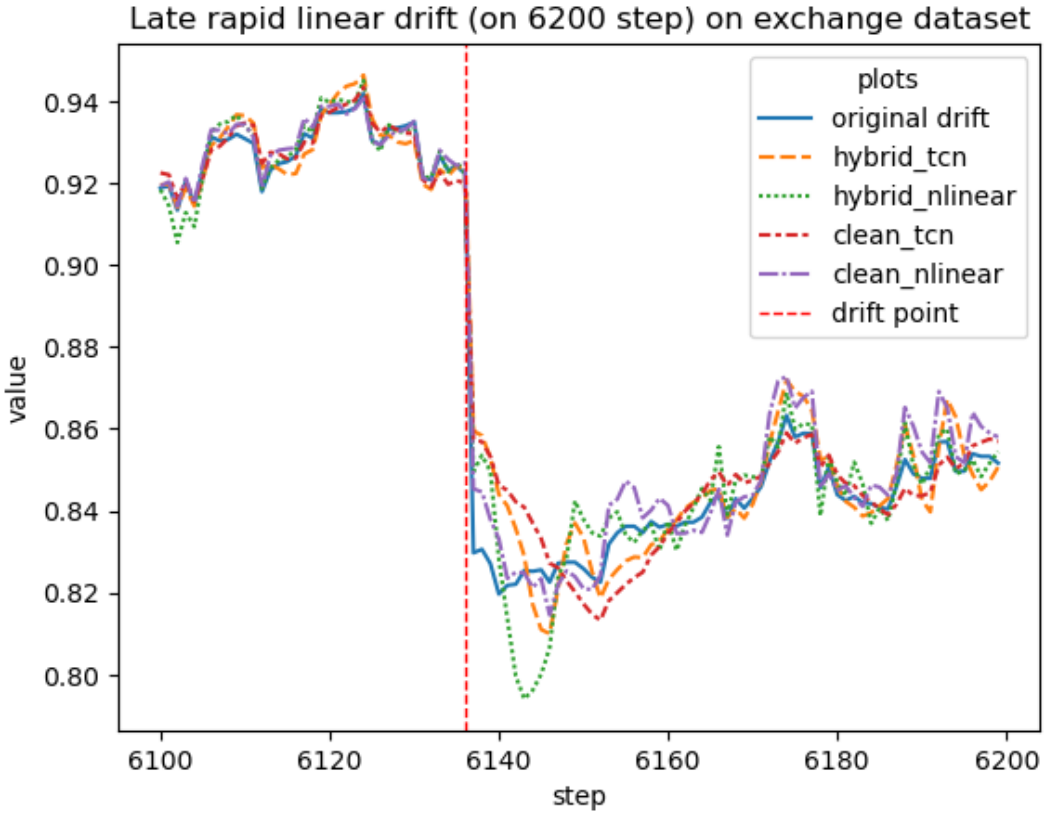


Рисунок 4.2 – Результат прогнозування моделей при дії різкого дрейфу на пізньому етапі онлайн навчання

На графіку до концептуального дрейфу можна побачити, що гібридні та базові моделі добре справляються із прогнозуванням форми часового ряду, отже моделі достатньо адаптувались до середовища. На графіку після концептуального дрейфу можна побачити нестабільність у базових моделях

та гібридних моделях. Через малу глибину модель NLinear швидко адаптувалась до концептуального дрейфу, в той же час гібридні моделі мали сильні стрибки у прогнозах після дрейфу.

На рисунку 4.3 зображений лінійний продовжний дрейф у кінці навчання. Можна побачити якісну адаптацію моделей до форми часового рядку, отже гібридні моделі здатні працювати в онлайн при менш агресивних умовах дрейфів.

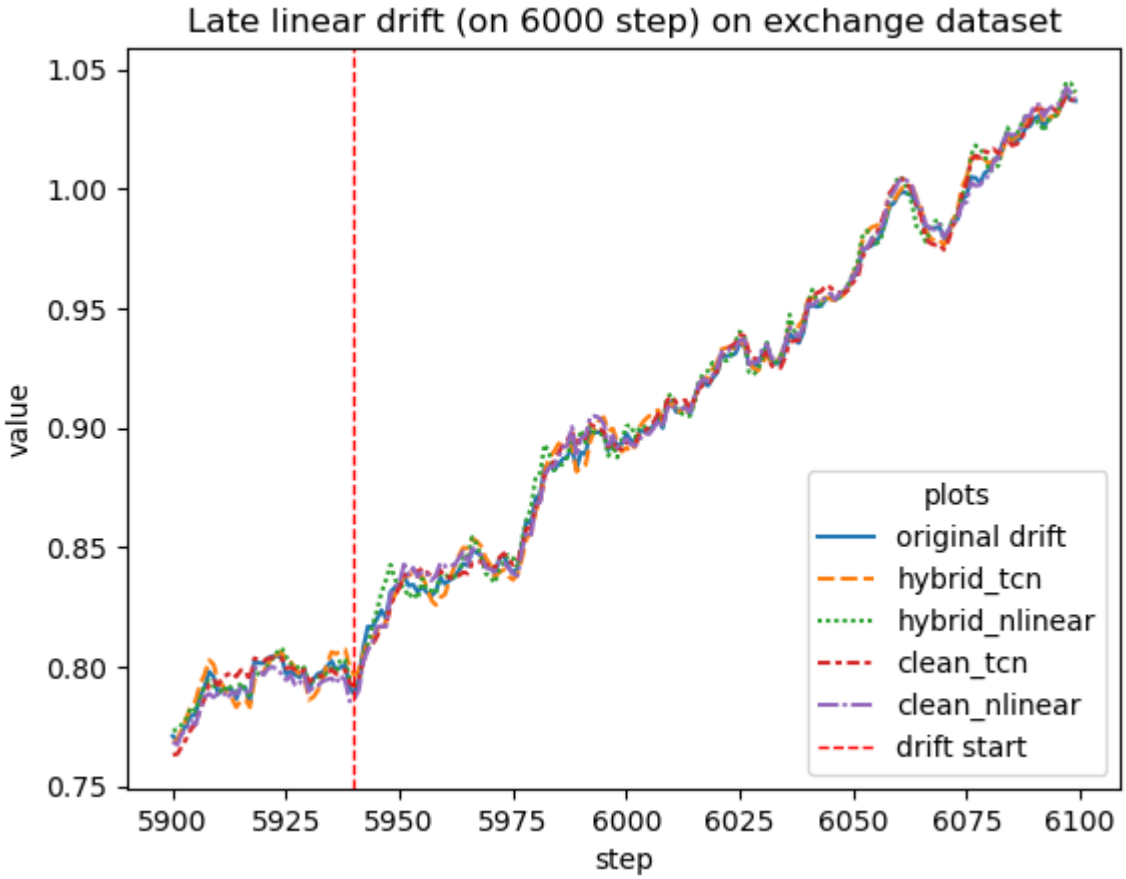


Рисунок 4.3 – Результат прогнозування моделей при дії повільного дрейфу на пізньому етапі онлайн навчання

Гібридні та базові моделі вдало адаптувались до повільного концептуального дрейфу без різких стрибків у прогнозах. Отже онлайн гібридна модель здатна впоратись із повільними лінійними дрейфами, як і не глибокі мережі. Візуально важко сказати, яка модель підлаштовується

краще часового ряду, проте, відповідно до таблиць, гібридні моделі працюють краще за чисті моделі.

На рисунку 4.4 зображено нелінійний поліноміальний дрейф під кінець тренування. Попри показники середньоквадратичної похибки гібридна модель NLinear має регіони нестабільності, які можна побачити на рисунку. Гібридні моделі достатньо добре описують форму і нестабільність часового ряду.

Гібридна модель TCN показала себе добре під час обробки нелійнійного концептуального дрейфу та швидко адаптовувалась до змін. Гібридна модель NLinear має вищий показник волатильності, що можна побачити на пікових значеннях змін. Базова модель NLinear добре адаптувалась до змін.

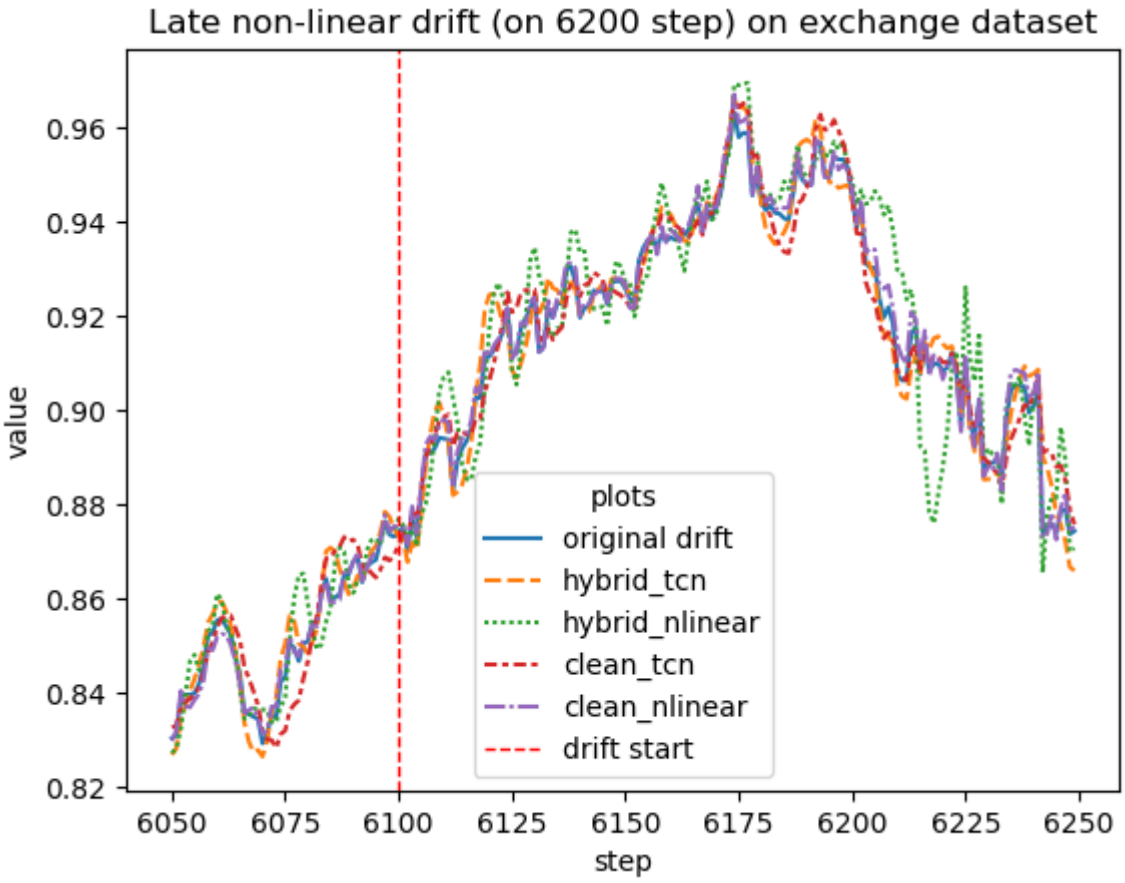


Рисунок 4.4 – Результат прогнозування моделей при дії нелінійного повільного дрейфу на пізньому етапі онлайн навчання

На рисунку 4.5 можна побачити нелійнійний дрейф під час початку навчання. Відображений регіон достатньо рівний, що зумовлює часткову стабільність всіх моделей. У кінці дрейфу можна побачити різкий перехід у нових стан середовища. Гібридна модель NLinear єдина змогла швидко і правильно підлаштуватись під новий стан середовища. Інші моделі видавали нестабільні результати, а отже вивчились на викид, замість запам’ятовування загальної форми числової послідовності. Модель TCN продовжувала бути нестабільною до моменту закінчення оглянутого графіку.

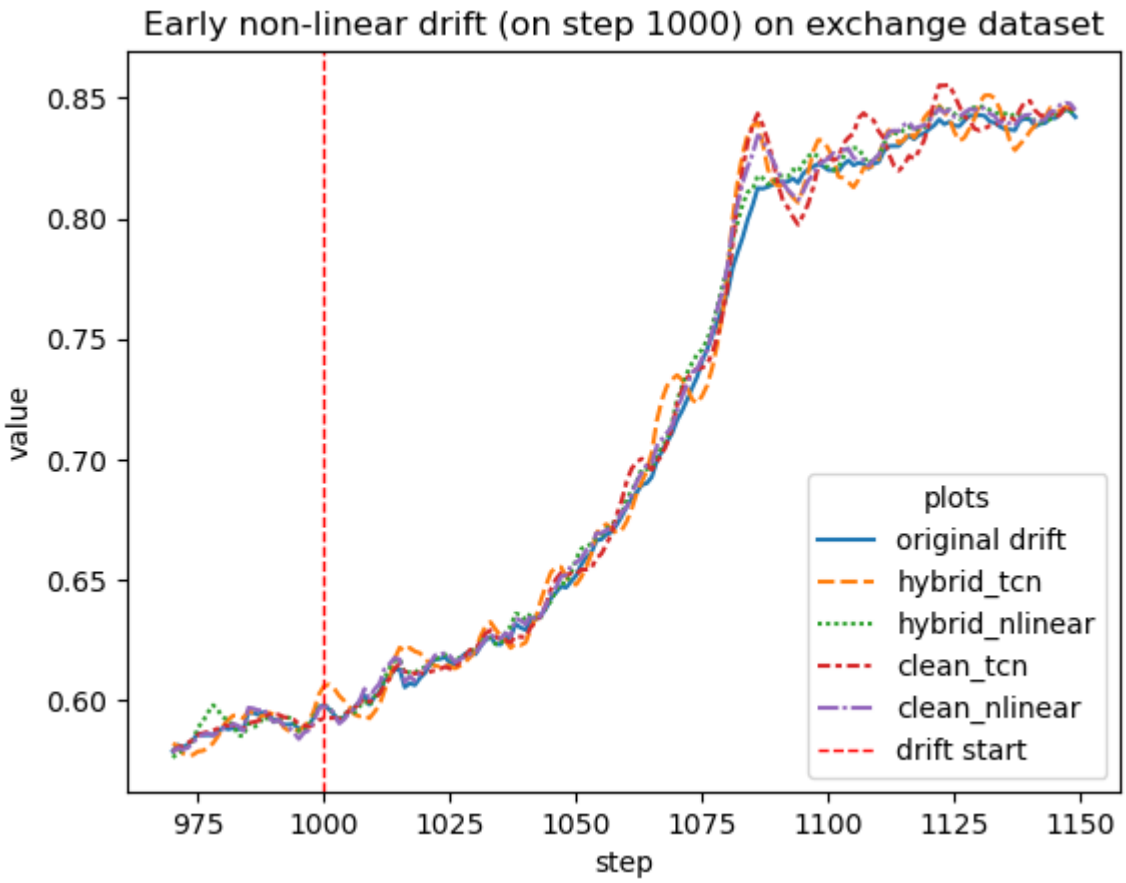


Рисунок 4.5 – Результат прогнозування моделей при дії нелінійного повільного дрейфу на ранньому етапі онлайн навчання

Аналіз графіків дозволив зробити висновки про адаптивність онлайн гібридної моделі. Було визначено, що нестабільність базової моделі сприяю

нестабільності адаптивної моделі, що погіршує показник похибки моделі. Розроблені гібридні моделі показали швидкі властивості адаптації до різних форм датасету exchange.

Моделі показали нестабільність при різких лінійних концептуальний дрейфах. Модель гібридної NLinear здатна адаптовуватись до нелійнійних поліноміальних дрейфів навіть на початковому етапі. Всі обрані моделі були здатні адаптуватись до повільних лінійних дрейфів. Всі моделі були здатні адаптовуватись до складності часових рядів.

Проте гібридні моделі мали нестабільність на початковому етапі навчання, що може бути неприйнятно під час використання моделей у реальному середовищі.

Отже, було прийнято рішення про офлайн навчання базової моделі, що дозволить позбутись нестабільності і пришвидшить адаптацію моделі на основі залишків.

Наступний експеримент був проведений на датасеті traffic, яка містить різни види дрейфів та більше даних для аналізу та навчання. В якості моделей було обрано GRU, TCN, NLinear. Модель LSTM та DLinear не використовувались, бо показували гірші результати ніж архітектурно схожі моделі. В якості адаптивної моделі використовується нео-нечітка модель без еволюції.

Оскільки у реальному середовищі на виробництві може бути різна кількість прикладів, які можна використовувати в якості тренувальних вибірок, то було проведено досліди на тренувальних вибірках різних розмірностей.

Було використано декілька пропорцій розбиття тестової та тренувальної вибірок: 10/90, 30/70, 50/50, 70/30, 90/10.

Кожна тестова вибірка була рівномірно розділена на тестову та валідаційну вибірку, для оцінки якості тренування моделі на кожній із епох.

Для уникнення перенавчання було зроблено раннє завершення навчання моделі за ознакою втрат валідаційної вибірки.

Під час експерименту базові моделі навчались офлайн на тренувальній вибірці. Після навчання базова модель заморожується. Створюється гібридна модель до якої додається базова модель та адаптивна модель. Адаптивна модель, у складі гібридної моделі, навчається на тренувальній вибірці за один прохід. Після цього починається процес онлайн навчання. Під час онлайн навчання відбується адаптація нео-нечіткої мережі до тестової вибірки та розрахунок показників середньоквадратичної похибки. Середньоквадратична похибка розраховується для базової моделі без онлайн навчання, для гібридної моделі з онлайн навчанням та для базової моделі з онлайн навчанням.

Під час навчання онлайн частини валідаційну та тестову вибірку було складено у одну вибірку.

Результати проведеного експерименту можна побачити у таблиці 4.4. Відповідно до таблиці гібридна система принесла суттєву користь моделі TCN, яка навчалась на тренувальній вибірці менше 90. Гібридна модель не дала суттєвих результатів для NLinear, GRU у онлайн обробці, покращення в яких становлять не більше 10^{-2} в порівнянні з базовою моделлю без онлайн навчання. Онлайн навчання базових моделей привело до погіршення узагальнюючих властивостей та великого показника забування часової послідовності. Онлайн навчання базових моделей мало сенс у моделі TCN при пропорціях розділення тренувальних вибірок 70/30, 50/50, 30/70, що дало підвищення якості моделі в порівнянні із офлайн базовою моделлю. Великі тренувальні вибірки не дали суттєвого збільшення точності, оскільки обрані базові моделі мають невелику архітектуру. Велика вибірка приводила до перенавчання моделі вже у першій епосі, незважаючи на використання clipping та L2 регуляризацію в обох моделях. Показники похибки у базових моделей, які навчались офлайн з онлайн донавчанням менше ніж значення похибки у моделей, які навчались повністю онлайн.

Можна побачити, що навчання на найбільшій вибірці не обов’язково дає найкращий результат, особливо у таких малих моделях як NLinear.

Найкращі результати у NLinear та TCN досягаються при 30% тренувальної вибірки. У випадку GRU найкращий результат досягається при 70% тренувальної вибірки.

Таблиця 4.4 – Порівняння результатів навчених у офлайн та онлайн моделей на датасеті traffic

Тип навчання	Значення MSE для різних Train/Test split				
	90/10	70/30	50/50	30/70	10/90
NLinear офлайн	0.095	0.064	0.077	0.062	0.065
NLinear гібрид	0.094	0.062	0.073	0.061	0.064
NLinear онлайн	0.155	0.273	0.254	0.227	0.197
TCN офлайн	0.079	0.072	0.076	0.076	0.074
TCN гібрид	0.072	0.055	0.055	0.053	0.07
TCN онлайн	0.092	0.067	0.072	0.054	0.169
GRU офлайн	0.083	0.076	0.082	0.1	0.093
GRU гібрид	0.081	0.071	0.076	0.092	0.087
GRU онлайн	0.247	0.218	0.202	0.188	0.167

Отже було зроблено перевірку якості моделей з різними видами навчання.

4.2 Оцінка отриманих результатів

У результаті проведених експериментів можна зробити висновок про якість розробленого комбінованого методу.

Оцінка якості була визначена за допомогою метрики MSE.

При тесті онлайн адаптивності, гібридна модель, яка складається із базової моделі та адаптивної моделі, яка є нео-нечіткою мережею, підвищила або зменшила похибки для всіх моделей у різних задачах. Використання гібридної моделі підвищило показник середньоквадратичної похибки на датасеті electricity, для всіх інших датасетах показник зменшився. Датасети ettm та weather не принесли суттєвих зменшень похибки навчання. На датасеті exchange похибка була зменшена на 0.4, в інших датасетах зменшення знаходились в діапазоні 0.1. Вплив адаптера на модель NLinear був більший TCN, що спричинило різний розподіл найкращих моделей у чистих та гібридних варіантах.

Отже гібридна модель може бути ефективно використана в якості адаптера до вхідних даних при онлайн навчанні. Проте варто перевіряти роботу подібної архітектури під кожну окрему задачу.

Досліджувані датасети були створені для опису різних можливих станів та умов часових рядів для задач прогнозування. Датасети мають різні види та форми концептуальних дрейфів. Датасет exchange є одним із датасетів з найбільшими показниками кількості та складності концептуальних дрейфів і високе зменшення похибки гібридною моделлю може сигналізувати про ефективність моделі з адаптером у задачах під дією концептуального дрейфу. Було прийнято рішення про створення штучних концептуальних дрейфів.

Було оглянуто взаємодію онлайн гібридної моделі із різними типами концептуальних дрейфів під час різного етапу навчання моделей.

Було створено різкі та повільні лінійні концептуальні дрейфи, нелінійні поліноміальні концептуальні дрейфи, дрейфи було додано під час початку навчання та у кінці навчання. Моделі на початку навчання були нестабільні через «розігрівання» базової моделі, перед початком навчання гібридної моделі, що означало майже відсутність навчання адаптеру. Гібридні моделі показали властивості адаптації до концептуальних дрейфів, проте нестабільні похибки базових моделей під час та після дрейфів сприяли

нестабільності адаптивної моделі. Особливо добре впоралась гібридна модель NLinear у задачі ранньої адаптації до поліноміального дрейфу, що дозволили уберегти модель від нестабільності протягом певної кількості кроків. В інших задачах з різкими дрейфами, NLinear показала власну нестабільність. Всі моделі впорались з адаптацією до повільного лінійного дрейфу. У випадку пізнього нелінійного дрейфу гібридна модель TCN показала гарні властивості до адаптації, в той же час гібридна модель NLinear занадто сильно збільшувала власну амплітуду часового ряду.

Окрім «розігріву» моделі, для вирішення проблеми нестабільності базової моделі було прийнято рішення про офлайн навчання базової моделі. Таким чином базова модель отримує знання про часовий ряд без забування, що дозволить краще узагальнювати вхідні дані. Офлайн навчання було проведено на моделях TCN, GRU, NLinear, оскільки моделі показали найкращі результати у фазі онлайн навчання. В якості датасету було обрано traffic, який містив більшу кількість даних для обробки. У офлайн режимі адаптер навчався лише у одну епоху із навченим на тренувальних даних scaler.

Гібридні моделі з попереднім офлайн навчанням базової моделі і подальшим онлайн навчанням адаптера доказали свою ефективність на прикладі датасету traffic.

У чистій моделі TCN вдалося зменшити похибку з 0.32 до 0.181 використовуючи гібридну модель при онлайн підході. Використовуючи офлайн навчання базової моделі TCN вдалось досягти найкращого значення похибки у 0.072, додаючи онлайн адаптер вдалось досягти значення похибки у 0.053. Отже на моделі TCN гібридним підходом вдалось знизити похибку на 0.14, офлайн базової моделі вдалось знизити похибку ще на 0.109, додавання онлайн адаптера допомогло знизити похибку на 0.019. Сумарно вдалось знизити значення похибки на 0.3.

Гібридна частина у онлайн навчанні моделі GRU зменшила похибку з 0.357 до 0.186. Офлайн навчання зменшило похибку до 0.076 додавання

онлайн адаптер зменшило похибку до 0.071. Гібридним підходом вдалось зменшити похибку на 0.171, офлайн навчання зменшило похибку ще на 0.11, онлайн адаптер допоміг додатково знизити похибку на 0.005.

Чиста модель NLinear мало похибку 0.387 при онлайн обробці, гібридна модель дозволила зменшити похибку до 0.204. У офлайн навчанні найкраща похибка чистої моделі була 0.062, гібридна модель з онлайн адаптером дозволила знизити похибку до 0.061. Гібридний підход зменшив похибку у онлайн навчання на 0.183. Різниця між онлайн та офлайн навчання із базовою моделлю становить 0.142. Додавання адаптеру до офлайн моделі зменшило похибку на 0.001.

Отже додавання гібридної моделі суттєво покращує роботу моделей при чистому онлайн підході. Навіть найменше офлайн навчання суттєво підвищує ефективність базової моделі. Додавання онлайн адаптеру у офлайн базову модель принесло результат у моделі TCN, для моделі GRU та NLinear покращення знаходяться на межі похибки, проте у онлайн обробці потоків даних невідомо складність подальших потоків даних, а отже гібридну модель можна заморозити до моменту початку дрейфу, який може бути визначений зовнішніми детекторами, або модель адаптеру можна лишити навчатись онлайн, бо модель не погіршує результат, в такому випадку модель буде вчитись даним, у яких базові моделі виконують якісну обробку та адаптер зможу адаптовуватись до дрейфів без потреби у детекторі дрейфу. Онлайн донавчання базової моделі принесли результати лише для моделі TCN, що дало покращення в порівнянні з офлайн моделлю. Для моделей GRU та NLinear онлайн понавчання після офлайн навчання погіршило ефективність базових моделей.

Як можна побачити, то ефективність гібридного підходу залежить від властивостей досліджуваного часового ряду та архітектури обраної базової мережі. Комбінований метод краще застосовувати разом з моделлю, яка навчалась офлайн. При довгій обробці часових рядів властивості системи можуть сильно змінитись, з чим і має допомогти адаптер.

Тому ефективне поєднання декількох гібридних мереж, з різними базовими моделями, у вигляді ансамблю має допомогти у вирішенні складних часових потоків для задачі прогнозування. Проте наявність адаптера у кожній із моделей може спричиняти нестабільність всієї системи, тому ансамбль має мати ваги для навчання, які б навчились на найбільш стабільні типи систем при конкретному виді концептуального дрейфу.

Подальші дослідження можуть бути зосереджені на використані адаптера у фундаментальних моделях.

Оскільки фундаментальні моделі навчаються на великих та різних датасетах і мають багато параметрів і можуть працювати над декількома часовими рядами, то онлайн навчання моделей сприяє катастрофічному забуванню даних. В той же час часові ряди можуть еволюціонувати з часом, що може привести до підвищення рівня похибки у фундаментальних моделей, у випадку складного дрейфу. Перенавчання фундаментальної моделі потребує великих ресурсів. Нео-нечітка модель в якості адаптера забезпечує навчання в умовах нестаціонарності середовища, коли показник похибки фундаментальної моделі починає зростати. Низька кількість параметрів дозволяє створювати створювати нео-нечітки моделі адаптери під кожну конкретну задачу, як у випадку із LoRA у стаціонарних середовищах. Еволюція адаптивної мережі в такому випадку має відбуватись з нижчою частотою, ніж раз на 1000 кроків, оскільки фундаментальні моделі деградують повільно, а еволюція породжує нестабільність у адаптивній моделі. Отже використання адаптерів у фундаментальні моделі має дозволити збільшити цикл життя фундаментальної моделі до моменту навчання нової моделі.

Нео-нечітка мережа адаптер не має використовуватись при відсутності значної еволюції часового ряду для прогнозування, бо буде вчитись на шумі, тим самим погіршувати якість базової моделі.

Проте фундаментальні моделі створені для того щоб адаптуватись до всіх даних, а отже невідомо, коли саме часовий ряд вийде за межі того, що

може бути якісно спрогнозовано фундаментальною моделлю, бо на відмінну від LoRA, яка навчається на даних, про які модель не знає, у задачах прогнозування важко додати у потік даних інформацію, про яку модель не знає.

Проте фундаментальні моделі, зазвичай, мають великий горизонт прогнозування, на відміну від нео-нечіткої мережі, яка має горизонт розмірністю один. В такому випадку нео-нечітка мережа має визначити наступні значення горизонту рекурсивно додаючи прогнози до входу моделі. Використання штучних прогнозів призводить до накопичення похибки прогнозування, а отже зменшує ефективність моделі адаптера. Тому краще застосовувати гібридну модель адаптера з фундаментальною моделлю у випадку невеликого горизонту передбачення.

То ж дослідження з фундаментальною моделлю має проходити з великою тестовою вибіркою, в якій можуть бути декілька різних потоків даних, які мають різні типи складних дрейфів, для того щоб переконатись, що фундаментальна модель не зможе впоратись із потоком даних і перевірити якість адаптації моделей нео-нечітких мереж або ж каскадних нео-нечітких мереж, у випадку додавання складних взаємодій між вхідними даними, на яких фундаментальна модель була не навчена.

ВИСНОВКИ

У межах виконаного дослідження було розглянуто сучасні підходи до вирішення завдання прогнозування в умовах концептуального дрейфу, здійснено їх систематизацію, порівняльний аналіз та розроблено власний комбінований метод, що базується на використанні гібридної моделі для досягнення підвищеної точності прогнозування.

Проведено аналіз предметної галузі. Визначено суть задачі прогнозування та складність її вирішення. Проведено аналіз літературних джерел.

Було проведено дослідження на перевірку адаптивності гібридної моделі до онлайн навчання та з окремим навчанням адаптера у онлайн режимі. Досліджено поведінку кожної із розроблених моделей при дії різних типів концептуальних дрейфів під час різного етапу навчання.

Для проведення досліджень було обрано п’ять базових моделей: GRU, LSTM, TCN, NLinear та DLinear. Було використано базові датасети для задач прогнозування: electricity, etth, ettm, exchange, traffic, weather, які описують різні типу складних середовищ.

В якості моделі адаптера було використано одношарову модель нео-нечіткого нейрона.

Нео-нечітка мережа не може визначати складних взаємодій між входами, а отже базова модель має справлятись з обробкою взаємодій між входами.

Нео-нечітку мережу було протестовано у еволюційному та статичному режимах.

Проте датасети були надто короткими задля визначення моменту закінчення циклу життя статичної нео-нечіткої мережі, внаслідок чого було навіть статичні моделі показували високі показники.

Основною проблемою нео-нечіткої мережі є один вихід моделі, через що дослідження проводились у середовищі з єдиним виходом.

Було проведено дослідження адаптивних властивостей гібридної мережі, використовуючи онлайн навчання базової та адаптивної моделі. Було порівняно результати якості базових та гібридних мереж у онлайн режимах. Гібридна модель показала кращі результати на датасетах окрім electricity. На датасетах weather та ettm результати покращень були незначними. На інших моделях вдалось суттєво зменшити похибку розрахунків.

Було проведено візуальне спостереження за реакцією гібридних та базових моделей на різні типи концептуальних дрейфів, під час різного етапу навчання моделей. Було визначено, що нестабільність базової моделі спричиняє нестабільність у адаптивної моделі, тому було прийнято рішення про стабілізацію базової моделі офлайн навчанням.

Було проведено дослідження додавання адаптивної мережі у офлайн базову модель для роботи із онлайн змінами. Було розбито датасет на тренувальні та тестові вибірки у різних пропорціях. Базові моделі були натреновані на тренувальних вибірках. Адаптер спочатку був натренований у гібридній моделі на тренувальній вибірці, з подальшим тренуванням на тестовій вибірці. Суттєві покращення гібридної моделі відбулись у моделі TCN, інші моделі мали несуттєві покращення. Отже ефективність гібридної моделі з офлайн навченою базовою моделлю залежить від архітектури моделі та складності часового потоку.

Основним завданням дослідження є розробка комбінованого методу прогнозування в умовах концептуального дрейфу та оцінка його працездатності і ефективності.

Запропонований в роботі комбінований метод прогнозування в умовах концептуального дрейфу є працездатним і ефективним для забезпечення високоточного прогнозування. Його універсальність і здатність до адаптації роблять цей метод ефективним інструментом як для наукових досліджень, так і для прикладних задач у сфері прогнозування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Montet F., Pasquier B., Wolf B. Benchmarking foundation models for time-series forecasting: zero-shot, few-shot, and full-shot evaluations. *ITISE 2025*. Basel Switzerland, 2025. P. 32. URL: https://doi.org/10.3390/cmsf2025011032 (date of access: 20.04.2026).

2. A time series is worth 64 words: long-term forecasting with transformers / Y. Nie et al. *International Conference on Learning Representations (ICLR 2023)*. 2023. URL: https://arxiv.org/abs/2211.14730 (date of access: 21.04.2026).

3. Parameter-efficient fine-tuning methods for pretrained language models: a critical review and assessment / L. Xu et al. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2026. P. 1–20. URL: https://doi.org/10.1109/tpami.2026.3657354 (date of access: 21.04.2026).

4. LoRA: low-rank adaptation of large language models / Hu E. J. et al. *Arxiv*. 2021. URL: https://doi.org/10.48550/arXiv.2106.09685 (date of access: 03.04.2026).

5. Are transformers effective for time series forecasting? / A. Zeng et al. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023. Vol. 37, no. 9. P. 11121–11128. URL: https://doi.org/10.1609/aaai.v37i9.26317 (date of access: 24.04.2026).

6. Attention is all you need / A. Vaswani et al. *Advances in Neural Information Processing Systems (NeurIPS 2017)*. 2017. URL: https://arxiv.org/abs/1706.03762 (date of access: 24.04.2026).

7. Learning to extrapolate and adjust: two-stage meta-learning for concept drift in online time series forecasting / W. Chen et al. *Thirty-Fourth International Joint Conference on Artificial Intelligence {IJCAI-25}*, Montreal, Canada, 16 – 22 August 2025. California, 2025. P. 4869–4877. URL: https://doi.org/10.24963/ijcai.2025/542 (date of access: 27.04.2026).

8. OneNet: enhancing time series forecasting models under concept drift by online ensembling / W. Chen et al. *Advances in Neural Information Processing Systems 36*, New Orleans, Louisiana, USA, 10–16 December 2023. San Diego, California, USA, 2023. P. 69949–69980. URL: https://doi.org/10.52202/075280- 3066 (date of access: 27.04.2026).

9. Li Sh. Plug-and-Play reinforcement learning for real-time forecast recalibration. *Medium*. URL: https://pub.towardsai.net/plug-and-play-reinforcement-learning-for-real-time-forecast-recalibration-02c6203c429c (date of access: 28.04.2026).

10. Zaffran M. Adaptive conformal prediction for time series forecasting / M. Zaffran et al. *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*. 2022. URL: https://arxiv.org/abs/2202.07282 (date of access: 28.04.2026).

11. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling / Bai S. et al. *Arxiv*. 2018. URL: https://doi.org/10.48550/arXiv.1803.01271 (date of access: 29.04.2026).

12. Informer: beyond efficient transformer for long sequence time-series forecasting / Zhou H. et al. *Arxiv*. 2020. URL: https://doi.org/10.48550/arXiv.2012.07436 (date of access: 29.04.2026).

13. Learning fast and slow for online time series forecasting / Pham Q. et al. *Arxiv*. 2022. URL: https://doi.org/10.48550/arXiv.2202.11672 (date of access: 29.04.2026).

14. N-BEATS: neural basis expansion analysis for interpretable time series forecasting / Oreshkin B. N. et al. *Arxiv*. 2019. URL: https://doi.org/10.48550/arXiv.1905.10437 (date of access: 29.04.2026).

15. Працибуда В.В., Петров К.Е. Модифікація методу DDM для вирішення задач потокової обробки даних з використанням сімейства моделей eTS. *Радіоелектроніка та молодь у XXI столітті*: матеріали XXX Міжнар. молодіж. форуму, 22-24 квіт. 2026 р. Харків, 2026, Т. 6. С. 102–104.

16. Kacprzyk J., Lin T. Y., Liau C.-J. Granular, fuzzy, and soft computing. Springer, 2023.

17. Industrial time series modelling by means of the neo-fuzzy neuron / D. Zurita et al. *IEEE Access.* 2016. Vol. 4. P. 6151–6160. URL: https://doi.org/10.1109/access.2016.2611649 (date of access: 30.04.2026).

18. Evolving neo-fuzzy neural network with adaptive feature selection / A. M. Silva et al. *2013 BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC)*, Ipojuca, Brazil, 8–11 September 2013. 2013. URL: https://doi.org/10.1109/brics-cci-cbic.2013.64 (date of access: 05.05.2026).

19. Zaychenko Y. P., Hamidov G. Cascade neo-fuzzy neural network in the forecasting problem at stock exchange. *System research and information technologies*. 2017. no. 2. P. 92–102. URL: https://doi.org/10.20535/srit.2308-8893.2017.2.09 (date of access: 05.05.2026).

20. Cascade evolutionary neural network with neo-fuzzy-neurons as nodes / Bodyanskiy Y. V. et al. *Eastern-European Journal of Enterprise Technologies*. 2011. URL: https://doi.org/10.15587/1729-4061.2011.1377 (date of access: 02.05.2026).