

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методика динамічного оновлення профілю користувача у рекомендаційній системі з урахуванням змін поведінки»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Олександр ІЛЛЮЧЕНКО
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62
Олександр ІЛЛЮЧЕНКО

Керівник: _____ Людмила СОЛЯНИК
д-р хім. наук, доц.

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Іллюченко Олександр Сергійовичу

1. Тема кваліфікаційної роботи: «Методика динамічного оновлення профілю користувача у рекомендаційній системі з урахуванням змін поведінки»

керівник кваліфікаційної роботи Людмила СОЛЯНИК, д-р хім. наук, доцент.

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, поведінкові дані користувачів, сучасні моделі та методи динамічного оновлення профілю, алгоритми машинного навчання для рекомендаційних систем, вимоги до точності та адаптивності персоналізованих рекомендацій.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасних рекомендаційних систем та моделей користувацького профілю.

2. Дослідження методів та алгоритмів динамічного оновлення профілю користувача з урахуванням змін поведінки.

3. Розробка математичної моделі динамічного оновлення профілю користувача.

5. Проектування та розробка модуля оновлення профілю користувача у рекомендаційній системі.

6. Експериментальні дослідження та оцінка запропонованої моделі.

5. Перелік ілюстративного матеріалу: *презентація*

1. Мета, об'єкт та предмет дослідження.

2. Актуальність роботи.

3. Математична модель динамічного оновлення профілю користувача.

4. Етапи роботи методики.

5. Результати моделювання на основі методики динамічного оновлення профілю користувача у рекомендаційній системі з урахуванням змін поведінки.

6. Висновки.

7. Апробація роботи.

6. Дата видачі завдання «31» жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	31.10 - 01.11.25	
2	Вивчення сучасних методів динамічного оновлення профілю користувача	02.11 - 05.11.25	
3	Дослідження орієнтованих на послідовності підходів та моделей глибокого навчання для моделювання поведінки користувача.	06.11 - 11.11.25	
4	Аналіз впливу короткострокових та довгострокових інтересів на формування профілю користувача та стійкість рекомендацій.	12.11 - 19.11.25	
5	Розробка та оптимізація математичної моделі динамічного оновлення профілю користувача.	20.11 – 31.11.25	
6	Реалізація та інтеграція моделі динамічного оновлення профілю користувача.	01.12 - 13.12.25	
7	Оформлення роботи: вступ, висновки, реферат	14.12 - 16.12.25	
8	Розробка демонстраційних матеріалів	17.12 - 19.12.25	

Здобувач вищої освіти

_____ (підпис)

Олександр ІЛЛЮЧЕНКО

Керівник

кваліфікаційної роботи

_____ (підпис)

Людмила СОЛЯНИК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 79 стор., 8 табл., 6 рис., 27 джерел.

Мета роботи – покращення процесу динамічного оновлення даних профілю користувача у рекомендаційній системі з урахуванням змін поведінки.

Об'єкт дослідження – процес оновлення профілю користувача у рекомендаційній системі.

Предмет дослідження – моделі та методи динамічного оновлення профілю користувача у рекомендаційній системі.

У роботі використано методи аналізу рекомендаційних систем, включаючи порівняльний аналіз алгоритмів, огляд сучасних архітектур динамічного оновлення профілю, математичне моделювання короткострокових і довгострокових інтересів користувача, а також методи формалізації поведінкових патернів.

Проведено аналіз сучасних методів динамічного оновлення профілю, зокрема time-decay моделей, sliding-window підходів, online-learning, sequence-aware моделей (LSTM, GRU, Transformer), графових нейронних мереж та контекстно-орієнтованих методів персоналізації.

Окрему увагу приділено платформам Spotify, YouTube, TikTok, Amazon, які демонструють практичні варіанти адаптації профілю на основі змін поведінки користувача.

Розроблено та оптимізовано математичну модель динамічного оновлення профілю користувача, яка поєднує короткострокові та довгострокові інтереси, використовує механізм часових ваг та адаптивного коефіцієнта балансу. Запропонована модель дозволяє точніше відокремлювати тимчасові зміни поведінки від стабільних переваг, зменшує вплив шумових дій та забезпечує плавність еволюції профілю.

Проведено експерименти щодо оцінювання стійкості та адаптивності запропонованої моделі, зокрема аналіз швидкості реакції на зміну поведінки, стійкість до випадкових шумових взаємодій, збалансованість короткострокових і довгострокових сигналів. Отримані результати демонструють переваги комбінованого підходу над класичними time-decay та sliding-window моделями.

КЛЮЧОВІ СЛОВА: ДИНАМІЧНЕ ОНОВЛЕННЯ ПРОФІЛЮ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, КОРОТКОСТРОКОВІ ІНТЕРЕСИ, ДОВГОСТРОКОВІ ІНТЕРЕСИ, ПОВЕДІНКОВЕ МОДЕЛЮВАННЯ, TIME-DECAY, SLIDING WINDOW, ONLINE LEARNING, SEQUENCE-AWARE MODELS, TRANSFORMERS, GNN.

ABSTRACT

Text part of the master's qualification work: 79 pages, 6 pictures, 8 table, 27 sources.

The purpose of the work - improvement of the dynamic user profile update process in a recommender system considering behavioral changes.

Object of research - the process of updating the user profile in a recommender system.

Subject of research - models and methods of dynamic user profile updating in a recommender system.

Summary of the work. In this work, a variety of methods for analysing recommender systems have been applied, including comparative analysis of algorithms, a review of modern architectures for dynamic profile updating, mathematical modelling of short-term and long-term user interests, as well as methods for formalizing behavioral patterns.

An analysis of contemporary methods of dynamic profile updating has been conducted, including time-decay models, sliding-window approaches, online learning, sequence-aware models (LSTM, GRU, Transformer), graph neural networks, and context-aware personalization techniques. Particular attention is given to platforms such as Spotify, YouTube, TikTok, and Amazon, which demonstrate practical implementations of profile adaptation based on changes in user behavior.

A mathematical model for dynamic user profile updating has been developed and optimized. This model combines short-term and long-term interests, uses temporal weighting mechanisms and an adaptive balancing coefficient.

The proposed model allows for more accurate separation of temporary behavioral changes from stable preferences, reduces noise influence, and ensures smooth evolution of the profile.

Experiments have been conducted to evaluate the stability and adaptability of the proposed model, including the analysis of reaction speed to behavioral changes, robustness to random noisy interactions, and the balance between short-term and long-term signals. The obtained results demonstrate the advantages of the combined approach over classical time-decay and sliding-window models.

KEYWORDS: DYNAMIC PROFILE UPDATING, RECOMMENDER SYSTEMS, SHORT-TERM INTERESTS, LONG-TERM INTERESTS, BEHAVIORAL MODELLING, TIME-DECAY, SLIDING WINDOW, ONLINE LEARNING, SEQUENCE-AWARE MODELS, TRANSFORMERS, GNN.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	12
ВСТУП	13
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	16
1.1 Класифікація рекомендаційних систем, принципи роботи та сучасні тенденції персоналізації.....	16
1.2 Структура, параметри та роль у формуванні рекомендацій профілю користувача.....	19
1.3 Моделювання поведінки користувача	23
1.3.1 Поведінкові закономірності та їх роль у персоналізації	23
1.3.2 Часові зміни поведінки користувача.....	24
1.3.3 Короткострокові й довгострокові інтереси	25
1.3.4 Особливості моделювання поведінки користувача в рекомендаційній системі	26
1.4 Аналіз сучасних систем, що використовують динамічне оновлення профілю.....	27
1.4.1 Медіастрімінгові платформи з адаптивною моделлю профілю	28
1.4.2 Музичні стрімінгові сервіси з миттєвою адаптацією	28
1.4.3 Платформи з поведінковим профілюванням на основі взаємодії	29
1.4.4 E-commerce платформи з короткостроковою адаптацією інтересів	29
1.4.5 Спільні обмеження існуючих систем.....	30
2 АНАЛІЗ МЕТОДІВ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА	31
2.1 Огляд існуючих підходів до оновлення профілю користувача	31
2.2 Приклади реалізацій динамічного оновлення профілю користувача у промислових системах.....	33
2.3 Порівняльний аналіз методів адаптації до змін поведінки користувачів	35
2.3.1. Групи методів адаптації	35
2.3.2. Методи з часовим зважуванням (Time Decay Models)	36
2.3.3. Моделі з ковзним вікном (Sliding Window Models).....	37
2.3.4. Порівняння Time Decay та Sliding Window	38
2.3.5. Методи з оновленням параметрів у потоковому режимі (Online / Incremental Learning).....	38
2.3.6. Контекстно-орієнтовані методи адаптації (Context-Aware Models)	40
2.3.7. Методи відстеження “станів” користувача (User State Tracking).....	41
2.3.8. Графові методи адаптації профілю користувача	42
2.3.9. Методи адаптації на основі навчання з підкріпленням (Reinforcement Learning, RL)	44

2.3.10. Метрики адаптивності та способи оцінки	45
2.3.11. Порівняльна таблиця методів адаптації.....	45
2.4 Математична модель динамічного оновлення профілю користувача.....	47
2.4.1 Загальна концепція моделювання профілю користувача	47
2.4.3 Оновлення короткострокового та довгострокового профілів	49
2.4.4 Оцінювання впливу нових і старих взаємодій у моделі з часовими вагами	50
2.4.5 Інтеграція короткострокових та довгострокових вподобань у єдиній моделі.....	51
2.4.6 Адаптивний механізм оновлення параметра балансу α	52
2.4.7 Модель інтеграції контекстних факторів у процес оновлення профілю	53
2.4.8 Узагальнена модель багатфакторного оновлення профілю.....	54
3 РОЗРОБКА МЕТОДИКИ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА .57	
3.1 Опис запропонованої методики динамічного оновлення профілю користувача	57
3.1.1 Архітектура запропонованої методики.....	57
3.1.2 Алгоритм динамічного оновлення профілю	59
3.1.3 Робота компонентів та їхня взаємодія	60
3.1.4 Обробка потокових даних	60
3.1.5 Таблиця типів подій та їхніх ваг.....	61
3.2 Використані програмні засоби	61
3.3 Реалізація системи	68
3.3.1 Архітектура модулів	68
3.3.2 Логіка взаємодії компонентів	69
3.3.3 Інтерфейс користувача	70
3.3.4 Приклади роботи системи	71
3.3.5. Модуль логування та моніторингу.....	72
3.3.6. Моніторинг у реальному часі	73
3.3.7. Результати тестування системи	74
3.3.8. Сумарна оцінка продуктивності всієї системи	76
3.3.9. Візуалізація потокової роботи	77
ВИСНОВКИ.....	79
ПЕРЕЛІК ПОСИЛАНЬ	81
ДОДАТОК А. ДЕМООНСТРАЦІЙНІ МАТЕРІАЛИ	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- RS – Recommender System (рекомендаційна система)
- CF – Collaborative Filtering (колаборативна фільтрація)
- CBF – Content-Based Filtering (контентно-орієнтована фільтрація)
- MF – Matrix Factorization (матрична факторизація)
- ALS – Alternating Least Squares (алгоритм почергових найменших квадратів)
- BPR – Bayesian Personalized Ranking (байєсівське персоналізоване ранжування)
- GNN – Graph Neural Network (графова нейронна мережа)
- RNN – Recurrent Neural Network (рекурентна нейронна мережа)
- LSTM – Long Short-Term Memory (рекурентна мережа з довгою короткостроковою пам'яттю)
- GRU – Gated Recurrent Unit (рекурентна мережа з керованими воротами)
- TFRS – TensorFlow Recommenders (фреймворк рекомендаційних моделей на базі TensorFlow)
- STI – Short-Term Interests (короткострокові інтереси)
- LTI – Long-Term Interests (довгострокові інтереси)
- Time-decay – модель часової деградації ваг
- Sliding window – модель ковзного вікна
- Online learning – інкрементальне / потокове навчання
- RL – Reinforcement Learning (навчання з підкріпленням)
- HMM – Hidden Markov Model (прихована марковська модель)
- MDP – Markov Decision Process (марковський процес прийняття рішень)
- API – Application Programming Interface (прикладний програмний інтерфейс)
- IDE – Integrated Development Environment (інтегроване середовище розробки)
- JSON – JavaScript Object Notation (формат текстового обміну даними)
- SQL – Structured Query Language (мова структурованих запитів)

ВСТУП

Актуальність теми дослідження зумовлена стрімким розвитком рекомендаційних систем, які є ключовими компонентами сучасних цифрових платформ - від стрімінгових сервісів до інтернет-магазинів та соціальних мереж. Ефективність рекомендаційної системи безпосередньо залежить від точності побудови та оновлення профілю користувача, який відображає його інтереси, поведінку та контекст взаємодії.[3] Сучасні користувачі демонструють динамічні, непостійні та багатовимірні патерни поведінки, що робить традиційні статичні моделі профілю малоефективними. У багатьох промислових системах (YouTube, Spotify, TikTok, Amazon) першочерговим викликом є здатність моделі швидко реагувати на зміни інтересів, відфільтровувати «шумові» взаємодії та підтримувати баланс між короткостроковими й довгостроковими вподобаннями.[5] Тому проблема побудови адаптивного механізму динамічного оновлення профілю користувача є надзвичайно актуальною як з наукової, так і з практичної точки зору.

Мета роботи – покращення процесу динамічного оновлення даних профілю користувача у рекомендаційній системі з урахуванням змін поведінки.

Для досягнення поставленої мети визначено такі основні завдання дослідження:

1. Провести аналіз сучасних моделей і методів оновлення профілю користувача в рекомендаційних системах.
2. Дослідити алгоритми time-decay, sliding-window, online learning, sequence-aware та графові методи щодо їх здатності адаптуватися до змін поведінки.
3. Розробити математичну модель динамічного оновлення профілю, яка поєднує короткострокові та довгострокові інтереси користувача.
4. Спроекувати архітектуру модуля Profile Update Engine для інтеграції в рекомендаційну систему.

5. Реалізувати експериментальну систему оновлення профілю з використанням сучасних ML-фреймворків (TensorFlow Recommenders, PyTorch, River).
6. Провести експериментальну оцінку точності, адаптивності та стійкості розробленої моделі до шумових даних і різких змін поведінки користувача.
7. Порівняти отримані результати з класичними методами та оцінити їх ефективність у реальних сценаріях застосування.

Об'єктом дослідження є процес оновлення профілю користувача у рекомендаційній системі.

Предметом дослідження є моделі та методи динамічного оновлення профілю користувача у рекомендаційній системі.

У дослідженні використано такі методи: аналіз науково-технічної літератури та існуючих архітектур рекомендаційних систем; математичне моделювання динаміки інтересів користувачів; побудова алгоритмів *time-decay*, *sliding-window* та онлайн-оновлення; моделювання послідовної та графової поведінки; програмна реалізація за допомогою TensorFlow Recommenders, PyTorch, River та інших ML-фреймворків; експериментальна перевірка адаптивності моделі.

Наукова новизна одержаних результатів полягає у наступному:

1. Удосконалено математичну модель динамічного оновлення профілю, яка поєднує короткострокові та довгострокові інтереси та забезпечує стабільну реакцію на зміну поведінки.
2. Удосконалено метод адаптації профілю шляхом поєднання *time-decay* механізму, онлайн-оновлення та *sequence-aware* моделей у єдиному гібридному підході.
3. Отримали подальший розвиток моделі обробки поведінкових сигналів шляхом інтеграції різних рівнів контексту та розроблення механізму адаптивного балансування ваг інтересів.

Практичне значення отриманих результатів полягає в можливості впровадження розробленої моделі динамічного оновлення профілю у промислові

рекомендаційні системи, що дозволить підвищити точність персоналізованих рекомендацій, адаптивність до змін поведінки користувача та ефективність роботи платформи в умовах великої кількості подій у реальному часі.

Апробація результатів магістерської роботи. Основні положення та отримані результати були представлені 24 квітня 2025р. на Всеукраїнській науково-технічній конференції «Застосування програмного забезпечення в ІКТ», та 15 травня 2025р. на V Всеукраїнській науково-практичній конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті».

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Класифікація рекомендаційних систем, принципи роботи та сучасні тенденції персоналізації

У цифровому середовищі, де користувачеві доступні тисячі одиниць контенту чи товарів, рекомендаційні системи виступають ключовим механізмом навігації та індивідуалізації взаємодії. Їх головне призначення полягає в тому, щоб визначити, які елементи каталогу мають найбільшу цінність для конкретної людини у конкретний момент часу. Еволюція таких систем відбувалася паралельно з розвитком інтернет-платформ і методів аналізу даних: від простих фільтрів до складних моделей зі здатністю враховувати контекст і динаміку поведінки.

У науковій літературі не існує єдиної класифікації, проте найбільш поширеним є поділ на три великі групи: контентно-орієнтовані, колаборативні та гібридні системи.[1]

Контентно-орієнтовані моделі формують рекомендації на основі характеристик об'єктів і уподобань користувача. Вони порівнюють нові елементи з тими, яким користувач раніше віддавав перевагу. Такий підхід добре працює у сферах, де об'єкти мають виражені властивості, наприклад у музичних сервісах або книжкових каталогах. Водночас він схильний до проблеми надмірної однотипності рекомендацій через «замкнутість» у межах уже відомих інтересів [2].

Колаборативна фільтрація базується не на властивостях контенту, а на поведінці спільноти. Система аналізує схожість між користувачами чи між об'єктами, будує матрицю взаємодій та прогнозує, які елементи можуть бути релевантними. Її перевагою є здатність відкривати користувачеві нові категорії контенту. Однак класичні алгоритми цього типу залежать від достатньої

кількості даних, тому стикаються з проблемами “холодного старту” та розрідженості матриці.[3]

Гібридні моделі поєднують обидва підходи, компенсуючи їхні недоліки. Сучасні системи найчастіше реалізують саме гібридні схеми, інтегруючи також елементи машинного навчання, факторизацію матриць, графові методи, контекстне моделювання та глибинні нейронні мережі.[1; 4]

Функціонування рекомендаційної системи зазвичай включає кілька послідовних етапів.

Перший етап - збір даних, до яких можуть належати як явні оцінки, так і неявні сигнали: час перегляду, частота переходів, історія пошуку, кліки або взаємодія з окремими розділами платформи. Значну роль відіграє також контекст - момент часу, пристрій, геолокація, сезонні тренди.

Другий етап - обробка інформації, що включає очищення даних, виявлення шуму, стандартизацію та перетворення у формат, придатний для моделювання. У великих системах цей процес супроводжується побудовою ознак, агрегуванням подій та виділенням латентних характеристик.[3]

Третій етап - побудова моделі прогнозування, що залежить від застосованої математичної чи алгоритмічної бази. Традиційні методи використовують подібність евклідовими метриками або косинусною відстанню, тоді як сучасні підходи - факторизацію матриць, градієнтні методи оптимізації чи глибинні нейромереві архітектури.[2; 4]

Четвертий етап - генерація рекомендацій, де відбувається ранжування кандидатів та обчислення індивідуальної корисності для кожного об'єкта. У великих платформах цей етап часто розділяється на два підпроцеси: вилучення широкого набору можливих елементів і подальше детальне ранжування.[5]

П'ятий етап - оцінка результатів. У лабораторних умовах використовуються метрики точності (MAP, NDCG, Precision@K), тоді як у реальних сервісах основним індикатором успішності є поведінка користувачів: глибина сесії, показники взаємодії та конверсії.[3]

Упродовж останнього десятиліття рекомендаційні системи перестали бути статичними механізмами, що реагують лише на сукупність попередніх дій. У центрі розвитку знаходяться моделі, здатні адаптуватися до динаміки поведінки користувача, враховувати довгострокові й короткострокові інтереси, а також реагувати на зміни контексту.

Одним із ключових напрямів є застосування нейронних мереж для моделювання складних взаємозв'язків між користувачами та об'єктами. Рекурентні та трансформерні архітектури дозволяють враховувати часові залежності, а графові нейромережі ефективні у випадках, коли профіль користувача можна подати як багатовимірний граф взаємодій.[4; 6]

Іншим важливим трендом є контекстуальна персоналізація, коли рекомендації залежать від поточних умов використання сервісу. Наприклад, алгоритм враховує, що користувач переглядає контент зі смартфона увечері, коли має обмежену увагу та віддає перевагу короткому формату.[7]

Окремо варто відзначити динамічне формування профілю користувача, що є критично важливим у середовищах із високими змінами інтересів. Системи переходять від статичних моделей до адаптивних, у яких інформація про користувача постійно уточнюється, оновлюється та зважується залежно від давності та значущості сигналів.[6]

Сучасні тенденції також включають використання змішаних джерел даних, інтеграцію соціального контексту, емоційних характеристик, аналізу текстових описів за допомогою моделей обробки природньої мови, а також вставлення логіки пояснюваності рекомендацій для підвищення довіри користувачів.[1; 4]

У сукупності ці напрями демонструють, що рекомендаційні системи перестають бути суто інструментом пошуку релевантного контенту. Вони перетворюються на комплексні адаптивні моделі поведінкового аналізу, здатні прогнозувати зміни інтересів та активно підлаштовуватися до користувацької динаміки - що й визначає актуальність подальших досліджень у сфері оновлення профілю користувача.

1.2 Структура, параметри та роль у формуванні рекомендацій профілю користувача

Профіль користувача є центральним елементом будь-якої рекомендаційної системи, оскільки саме він визначає характер персоналізації та ступінь відповідності рекомендацій індивідуальним інтересам. Незалежно від складності алгоритмів або архітектури системи, ефективність прогнозування значною мірою залежить від того, наскільки повно й коректно відображено модель користувача. У сучасних інформаційних сервісах профіль набуває форми багатовимірної структури, що поєднує як історичні дані, так і контекстуальні сигнали, глибокі латентні ознаки та поведінкові патерни.[1; 3]

У найзагальнішому розумінні профіль користувача - це формалізоване подання даних, що характеризують уподобання, інтереси, стиль поведінки та індивідуальні закономірності взаємодії користувача з платформою. Профіль виступає посередником між реальними діями людини та алгоритмічною логікою системи: він забезпечує перехід від сирих подій (кліків, переглядів, оцінок) до структурованих представлень, придатних для аналізу та прогнозування.[1]

До основних функцій профілю належать:

- збереження поточного стану інтересів;
- узагальнення поведінки у вигляді ознак;
- забезпечення вхідних даних для алгоритмів рекомендацій;
- адаптація системи до змін у поведінці користувача;
- узгодження довгострокових і короткострокових інтересів.

Фактично профіль є інформаційною моделлю користувача, яка постійно уточнюється та накопичує новий досвід взаємодії.[3]

За своєю структурою профіль може бути простим або складним, залежно від типу платформи та методів, що використовуються для персоналізації. В узагальненому вигляді його структура охоплює декілька рівнів (рис.1.1).

1. Явні дані (explicit data).

Це інформація, яку користувач надає добровільно:

- оцінки чи рейтинги;
- вподобання в анкетах;
- списки бажань;
- підписки на категорії чи теми.

Явні сигнали є високоточними, але їх зазвичай мало: більшість користувачів не залишає оцінок або робить це нерегулярно.[2]



Рис.1.1 Компоненти профілю користувача

2. Неявні дані (implicit data).

Це найбільш цінний і масивний тип інформації, який включає:

- історію переглядів, прослуховувань, покупок;
- тривалість взаємодії з об'єктами;
- частоту та час використання сервісу;
- навігаційні дії (переходи, пошуки, повернення).

На основі цих даних будуються латентні фактори - приховані ознаки, що відображають глибинні інтереси, які користувач не висловлює прямо.[3]

3. Контекстуальні параметри.

У міру розвитку мобільних сервісів та інтерактивних платформ контекст почав відігравати суттєву роль. До таких параметрів належать:

- час доби, день тижня, сезон;

- тип пристрою;
- місцезнаходження;
- поточна активність (робота, відпочинок, навчання).

Контекст дозволяє системі уникати статичності та формувати рекомендації більш доречні для певного моменту.[3; 6]

4. Демографічні та соціальні характеристики.

До них відносять вік, мову, регіон проживання, тип споживчої групи тощо. Такі дані допомагають у сегментації, однак у багатьох системах використовуються мінімально, щоб уникнути надмірного нав'язування патернів.[1]

5. Латентні представлення (embeddings).

Сучасні системи дедалі частіше замінюють класичні ознаки на латентні вектори. Це високорозмірні представлення користувача, сформовані нейронними мережами або методами факторизації. Вони містять приховані взаємозв'язки між діями й дозволяють точніше прогнозувати поведінку.[4; 6]

Профіль користувача виступає ключовим елементом у функціонуванні рекомендаційної системи на всіх етапах її роботи (рис.1.2).



Рис.1.2 Етапи роботи рекомендаційної системи з даними користувача

1. Фільтрація та відбір кандидатів.

На першому етапі система визначає, які об'єкти взагалі можуть бути цікавими для користувача. Для цього аналізуються довгострокові інтереси, накопичені в профілі, а також їх співвідношення з поточними запитами.[3]

2. Ранжування.

Другий етап передбачає визначення важливості кожного кандидата. Алгоритм оцінює ступінь відповідності об'єкта профілю користувача, ураховуючи відносини між ознаками, латентними векторами та контекстом [4; 5].

3. Оновлення профілю після взаємодії.

Рекомендаційні системи з навчанням у режимі реального часу постійно уточнюють модель користувача: кожен перегляд, відмова або взаємодія змінює ваги параметрів профілю. Саме здатність до такого оновлення дозволяє системі адаптуватися до швидких змін у вподобаннях.[6]

4. Балансування довгострокових і короткострокових інтересів.

Профіль допомагає зберігати стійкі переваги, сформовані протягом тривалого періоду, але також враховує швидкоплинні наміри. Наприклад, користувач може тимчасово цікавитися певною темою, і система має реагувати на це, не змінюючи назавжди глобальний профіль.[4]

Ефективність формування профілю напряму залежить від якості даних, їх повноти та релевантності. Частими проблемами є:

- розрідженість взаємодій у нових користувачів;
- швидка зміна інтересів, що робить статичні моделі неточними;
- шум у даних, спричинений випадковими або вимушеними діями;
- переобтяження профілю старими подіями, які вже не відповідають актуальним інтересам;
- складність урахування контексту та залежностей між подіями.

Ці труднощі стали передумовою розвитку адаптивних методик, здатних не лише акумулювати інформацію, але й враховувати динаміку її змін, що є ключовим у системах сучасного покоління.[3; 6]

У більшості сучасних сервісів профіль користувача перетворюється з простого набору даних на складний динамічний об'єкт, який характеризує поведінку в широкому часовому та контекстуальному діапазоні. Саме він визначає якість персоналізації, здатність системи адаптуватися, зменшувати надмірність інформації та підтримувати високий рівень взаємодії користувачів із платформою. Таким чином, профіль стає не просто структурованим набором характеристик, а ключовою ланкою між людиною та алгоритмом - центральною частиною механізму формування рекомендацій. [1; 4]

1.3 Моделювання поведінки користувача

Моделювання поведінки користувача є одним з ключових напрямів розвитку сучасних рекомендаційних систем. На відміну від класичних підходів, що спиралися на статичні профілі та усереднені оцінки, сучасні моделі враховують динамічний характер взаємодії людини з платформою. Поведінка користувача змінюється як у коротких часових періодах (протягом дня), так і в довгих (місяці, роки), що створює необхідність у методах, здатних відтворювати та прогнозувати ці зміни. У цьому підрозділі розглянуто основні закономірності поведінки, типи інтересів та способи їх моделювання в рекомендаційних системах.

1.3.1 Поведінкові закономірності та їх роль у персоналізації

Поведінка користувача - це послідовність дій, які утворюють певні патерни взаємодії з платформою. Ці патерни можуть бути стабільними, випадковими або циклічними, і саме їх відтворення дозволяє будувати точні прогнози щодо майбутніх дій.[8]

До найбільш поширених поведінкових закономірностей належать:

1. Послідовність дій та залежності між подіями. Дії користувача рідко бувають випадковими. Часто вибір наступного елемента залежить від попереднього:

- після перегляду матеріалу певної тематики користувач частіше переходить до пов'язаних тем;
- завершення взаємодії з одним об'єктом підвищує ймовірність переходу до схожих або тематично споріднених;
- існують стійкі траєкторії поведінки, характерні для окремих груп користувачів.

Саме такі залежності використовують моделі типу LSTM, GRU та Transformer, які добре працюють із часовими послідовностями.[7–9]

2. Повторюваність патернів. Багато користувачів мають стабільні звички, що проявляються регулярно: перегляди в певний час доби, пріоритет певних жанрів/категорій, використання сервісу у вихідні тощо. Виявлення таких патернів дозволяє системі формувати рекомендації, узгоджені зі звичками користувача.[3; 8]

3. Залежність від зовнішнього контексту. На поведінку впливають зовнішні фактори: сезон, місце перебування, поточні події, освітній чи робочий графік. Контекстуальна поведінка має тенденцію до зміни, тому її важливо враховувати при оновленні профілю.[3]

4. Ситуативність вибору. У багатьох випадках користувач діє не відповідно до довгострокових інтересів, а під впливом моменту. Наприклад, пошук інформації для роботи чи навчання створює тимчасові зміни в поведінці.[8]

1.3.2 Часові зміни поведінки користувача

Поведінка має чітко виражену часову структуру. Сучасні рекомендаційні системи виділяють кілька рівнів часової динаміки.

1. Короткострокова динаміка – це зміни, які відбуваються протягом годин або днів. Вони включають:

- пошукові сесії;

- тематичні “сплески” інтересу;
- реакції на актуальні потреби;
- швидкі зміни контентної стратегії користувача.

Короткострокові сигнали мають велику вагу при формуванні рекомендацій у режимі реального часу. Такі системи потребують адаптивних моделей, здатних оновлювати профіль після кожної взаємодії.[7; 8]

2. Середньострокова динаміка охоплює тижні або місяці. Цей рівень відображає:

- тимчасові вподобання;
- зміни стилю поведінки;
- перехід користувача між сегментами (наприклад, новачок - досвідчений користувач).

Середньострокові зміни найчастіше моделюються через вагові коефіцієнти у профілі або ковзаючі вікна, де старі дії поступово втрачають вагу.[3; 8]

3. Довгострокова динаміка – це усталені уподобання, сформовані протягом тривалого періоду. Довгострокові інтереси складають основу профілю, забезпечуючи стабільність рекомендацій та запобігаючи хаотичним змінам у пропозиціях системи. Для їх виявлення використовують факторизаційні моделі, латентні вектори та узагальнені ознаки, що змінюються повільно.[3; 4]

1.3.3 Короткострокові й довгострокові інтереси

Важливим завданням є розмежування тимчасових і стабільних інтересів. Обидва типи відіграють різну роль у персоналізації.

Довгострокові інтереси – це фундаментні характеристики користувача: стилі, категорії, групи об’єктів, до яких він повертається знову і знову. Вони формуються повільно, стійкі до випадкових подій та визначають ядро рекомендацій. Моделювання довгострокових інтересів зазвичай здійснюється за допомогою матричної факторизації або статичних embeddings, де ваги змінюються лише під час періодичного перенавчання.[3; 4]

Короткострокові інтереси – це ситуативні наміри, які мають обмежений час дії. Приклади:

- пошук певної інформації на короткий період;
- перегляд серії пов'язаних матеріалів;
- реакція на одну подію чи потребу.

Короткострокові наміри суттєво впливають на поведінку протягом сесії, але не повинні суттєво змінювати довгострокову модель користувача.[7; 8]

Моделювання короткострокових інтересів часто ґрунтується на:

- рекурентних нейронних мережах (LSTM/GRU);
- Transformer-моделях;
- сесійних моделях типу SASRec, GRU4Rec;
- ковзаючих вікнах із підвищеною вагою останніх подій[7; 9].

Найпоширеніші стратегії комбінування обох типів інтересів включають:

- зважування подій у часі - нові сигнали мають більшу вагу;
- поділ профілю на "коротке" і "довге" вікно;
- гібридні моделі, де короткострокові embeddings поєднуються з довгостроковими
- моделі з механізмами уваги, які визначають, які події є релевантними саме зараз [4; 8; 9].

1.3.4 Особливості моделювання поведінки користувача в рекомендаційній системі

Моделювання поведінки користувача є технічно складним завданням через низку факторів:

1. Непередбачуваність користувача. Дії можуть змінюватися залежно від настрою, зовнішніх факторів чи випадкових причин.
2. Шум у даних. Не всі дії відображають реальні інтереси. Частина з них є випадковими або технічними (помилкові кліки, випадкові відкриття).[3]

3. Швидкоплинність намірів. Ситуативні інтереси можуть з'являтися й зникати протягом години, що ускладнює баланс між адаптивністю й стабільністю. [8]

4. Масштабованість. Платформи з мільйонами користувачів потребують моделей, здатних оновлювати профілі швидко й без повторного навчання на всіх даних. [3]

5. Взаємодія з контекстом. Одні й ті ж дії у різних контекстах можуть означати різне, що збільшує складність моделі. [8]

Сучасні рекомендаційні системи більше не спираються лише на загальні вподобання. Вони працюють із динамікою поведінки, що дозволяє:

- прогнозувати наміри користувача у межах поточної сесії;
- адаптувати рекомендації до актуального контексту;
- уникати застарілих або нерелевантних пропозицій;
- оновлювати профіль без повного перенавчання моделей;
- забезпечувати точні рекомендації навіть при мінімальній кількості історичних даних.

1.4 Аналіз сучасних систем, що використовують динамічне оновлення профілю

Сучасні рекомендаційні платформи дедалі частіше переходять від статичних моделей користувача до адаптивних механізмів, що враховують зміни поведінки у коротко- та довгостроковій перспективі. Динамічне оновлення профілю стає ключовим елементом підвищення точності персоналізації, оскільки воно дозволяє системі оперативно реагувати на зміни інтересів, наслідки контекстних чинників та взаємодію користувача з платформою. Нижче розглянуто кілька провідних платформ, які впроваджують різні варіанти динамічної адаптації профілю, а також визначено обмеження та невирішені проблеми, що формують підґрунтя для необхідності розробки нової методики.

1.4.1 Медіастрімінгові платформи з адаптивною моделлю профілю

Netflix давно використовує багаторівневу модель профілю, у якій поєднується історія переглядів, оцінки, часові патерни активності та контекст взаємодії. Оновлення профілю відбувається постійно: кожен перегляд або пропуск контенту коригує ваги інтересів користувача.

Платформа застосовує два ключові підходи:

- оновлення на основі останніх дій (short-term preferences) - якщо користувач протягом кількох днів дивиться переважно один жанр, система підвищує його значущість;
- довгострокове згладжування (long-term profile) - пріоритети вирівнюються з урахуванням загальної історії переглядів.[10]

Однак система не завжди розрізняє тимчасові та стабільні зрушення вподобань. Наприклад, перегляд незвичного жанру через зовнішній тригер (рекомендацію друга, настроєвий фактор) інколи надмірно зміщує модель. Це демонструє обмеженість механізму контекстного врахування.[4; 10]

1.4.2 Музичні стрімінгові сервіси з миттєвою адаптацією

Spotify активно застосовує динамічні моделі профілю, у яких музичні уподобання оновлюються практично після кожного прослуховування. Система відстежує:

- жанрові схильності,
- темп і ритмічні характеристики треків,
- ситуаційний контекст (час доби, активність користувача, тип пристрою),
- циклічність інтересів.

Особливістю Spotify є виявлення повернення до циклічних шаблонів, наприклад сезонних музичних вподобань. Проте навіть тут існують проблеми: складно відділити постійні інтереси від тимчасових імпульсів, а модель чутлива до шуму - коротка серія випадкових прослуховувань інколи впливає більше, ніж варто.[4; 7; 8]

1.4.3 Платформи з поведінковим профілюванням на основі взаємодії

YouTube застосовує гібридну адаптацію профілю, де поведінкові сигнали (утримання уваги, швидкість взаємодії, пропуски) інколи мають більшу вагу, ніж сам жанр контенту[5].

Профіль оновлюється:

- у режимі реального часу - після кожного перегляду;
- на довгих інтервалах - шляхом періодичної реконструкції моделі на основі всіх накопичених взаємодій.

YouTube стикається зі специфічною проблемою: окремі одиничні дії можуть суттєво зрізати або зміщувати тематичні кластери в профілі. Це особливо помітно, коли користувач переглядає контент, який мало відповідає його загальним інтересам (наприклад, ролик, пов'язаний з навчанням або роботою). У таких випадках система інколи помилково трактує дію як стійку зміну поведінки.[3; 5; 8]

1.4.4 E-commerce платформи з короткостроковою адаптацією інтересів

Amazon формує динамічний профіль, який активно реагує на зміни купівельної поведінки. На відміну від медіаплатформ, тут особливу роль відіграє:

- намір (intention-based modelling),
- короткострокові цілі користувача (пошук конкретного товару),
- індивідуальна сезонність.

Проблема Amazon - перевантаження профілю короткостроковими намірами. Наприклад, одноразовий пошук техніки для подарунку може призвести до довготривалої появи нерелевантних товарів у рекомендаціях.[3; 11]

1.4.5 Спільні обмеження існуючих систем

Незважаючи на активний розвиток, більшість систем мають низку загальних недоліків, пов'язаних саме з динамічним оновленням профілю:

1. Недостатнє розділення короткострокових і довгострокових інтересів. Навіть складні моделі часто не відрізняють тимчасову дію від фундаментальної зміни уподобань.[4; 7]

2. Висока чутливість до шумових дій. Окремі випадкові взаємодії можуть непропорційно впливати на профіль. [3; 5]

3. Недостатній урахунок контексту. Більшість систем трактують усі дії як однаково значущі, хоча вони можуть мати різну природу.[3; 7; 8]

4. Слабка інтерпретованість оновлень профілю. Користувач не завжди розуміє, як формується рекомендація, а розробники - що саме змінило модель [1; 3].

5. Проблеми з адаптацією до поведінки, яка змінюється плавно, а не стрибкоподібно. Поступові зміни інтересів, розтягнуті в часі, моделі часто фіксують занадто пізно.[4; 7; 8]

2 АНАЛІЗ МЕТОДІВ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА

2.1 Огляд існуючих підходів до оновлення профілю користувача

Оновлення профілю користувача є одним із ключових процесів у сучасних рекомендаційних системах, оскільки саме цей механізм забезпечує актуальність оцінок, релевантність прогнозів та здатність моделі реагувати на зміну інтересів. Протягом останніх років сформувалося кілька груп підходів, що відрізняються способом фіксації нової інформації, характером реакції на поведінкові патерни та рівнем адаптивності [1; 4]. У цьому підрозділі узагальнено найбільш поширені методи, які застосовуються в промислових і дослідницьких системах (рис.2.1).



Рис. 2.1 Підходи до оновлення профілю користувача

Статичні моделі профілю. Початковим етапом розвитку рекомендаційних технологій були статичні моделі, де профіль формувався під час реєстрації або

на основі обмеженого набору ознак, що не змінювалися протягом використання системи. До них належали анкети інтересів, самооцінки жанрів або списки улюблених об'єктів. [2; 3] Перевагою таких моделей була простота впровадження, проте їхній недолік очевидний: відсутність динаміки призводила до швидкої втрати актуальності профілю. Статичні підходи практично не використовуються у сучасних системах, але вони становлять історичну основу для розуміння еволюції методів.

Інкрементне оновлення стало одним із перших способів зробити профіль адаптивним: кожна взаємодія користувача збільшує вагу певної характеристики профілю. Подібні принципи застосовувалися в ранніх варіантах колаборативної фільтрації, а також у сучасних індустріальних системах із частковим online-learning[1; 10]. Недоліком є зсув у бік історичної інформації, через що модель повільно реагує на зміни інтересів.

Для врахування фактору часу використовуються моделі затухання (decay), де стара поведінка зменшує свою вагу. Такий підхід активно застосовується у sequence-aware моделях і time-aware факторизації[3; 8]. Поширені механізми:

- експоненційне затухання;
- лінійне згасання;
- групування взаємодій у часові вікна.

Правильно налаштоване затухання дозволяє реагувати на зміни, але може «забувати» довгострокові вподобання[4].

У міру зростання обсягів даних з'явилися контекстуальні методи, які враховують умови, за яких користувач взаємодіє з системою. До контексту можуть належати:

- час доби,
- тип пристрою,
- місце взаємодії,
- короткострокова мета,
- інтенсивність активності.

Оновлення профілю відбувається вибірково - система визначає, чи свідчить дія про зміну інтересів, чи є наслідком короткочасного контексту.

Це робить моделі більш стійкими, але потребує складного аналізу та значних обчислювальних ресурсів.[6; 12]

Методи факторизації та нейронні моделі представляють користувача як латентний вектор, що оновлюється під час навчання. У промислових системах поширені два варіанти: корекція векторів у режимі online-learning; періодичне донавчання моделі (batch learning). Такі методи забезпечують глибоке представлення інтересів, однак локальне оновлення інколи неможливе через необхідність глобального перенавчання[4; 7; 9].

Більшість платформ поєднує короткостроковий профіль (актуальні інтереси) і довгостроковий (стабільні вподобання). Подібні підходи застосовуються у sequence-aware, session-based і attention-моделях[7; 9]. Проблемою є баланс між двома частинами: якщо короткострокові сигнали мають надто велику вагу - рекомендації стають хаотичними, якщо надто малу - система ігнорує зміни інтересів.

Частина сучасних платформ використовує механізм активного навчання, де система самостійно уточнює профіль, запитуючи користувача про:

- релевантність запропонованих рекомендацій;
- причину перегляду чи вибору об'єкта;
- інтерес до нових категорій.

Цей підхід підвищує точність моделі, але має обмеження: надмірні запити знижують задоволення користувача та порушують природність взаємодії[1; 13].

2.2 Приклади реалізацій динамічного оновлення профілю користувача у промислових системах

1. Spotify: адаптація рекомендацій на основі короткострокових сигналів.

Spotify є однією з платформ, де проблема швидкої зміни музичних уподобань має найбільше значення. Система формує профіль, який складається

з довгострокових характеристик (улюблені жанри, повторювані артисти) та короткострокових сигналів, що відображають поточний настрій слухача. Наприклад, якщо користувач протягом дня прослухав кілька композицій з жанру lo-fi чи ambient, система тимчасово зміщує ваги профілю в бік спокійної музики. Такий підхід дає змогу уникати «інерційних» рекомендацій і орієнтуватися на актуальні інтереси.

Для обробки динамічних змін Spotify застосовує методи колаборативних моделей, які можуть інкрементально оновлювати вектори користувача та треків. Крім того, у рекомендаційній системі Spotify використовується моделювання послідовностей прослуховування, що дозволяє прогнозувати, які жанри чи треки можуть стати актуальними у найближчий період[14; 15].

2. YouTube: багаторівневе оновлення профілю та реакція на «свіжі дії» користувача

YouTube працює з мультимедійним контентом різних типів, а профіль користувача формується на основі численних поведінкових сигналів: переглядів, глибини взаємодії, лайків, коментарів, часу перегляду, пропусків та ін. Особливістю YouTube є поєднання довгострокових інтересів (улюблені тематичні канали, часто переглядані категорії) із короткостроковими «тригерами», такими як випадковий пошук, перегляд ролика іншого жанру чи вірогідність прив'язки до трендів.

YouTube активно використовує динамічне оновлення профілю для відображення зміни інтересів у реальному часі. Окремі моделі відповідають за підбір контенту в стрічці «Головна», інші - за рекомендації під відео, а ще інші - за персоналізацію пошуку. Всі вони отримують оновлені сигнали про дії користувача практично одразу, що дозволяє швидко коригувати його профіль.

Платформа також застосовує послідовні моделі (зокрема на базі Transformer-архітектур), які аналізують черговість переглядів та виявляють сценарії зміни інтересів[5; 10].

3. TikTok: надшвидке формування профілю та обробка мікросигналів

ТікТок відомий тим, що для побудови профілю користувача достатньо кількох хвилин активності. Це стало можливим завдяки унікальній архітектурі, яка враховує навіть найменші поведінкові стимули: час перегляду ролика до кінця, момент припинення перегляду, повтори, взаємодію зі звуками, переходи на профілі авторів, перегляд хештегів тощо. Такі сигнали мають різну вагу, і система постійно перебудовує профіль.

На відміну від класичних рекомендаційних систем, ТікТок майже не використовує статичні інтереси. Основний акцент робиться на короткострокових змінах поведінки. Наприклад, якщо користувач кілька разів переглянув відео з подібною тематикою (наприклад, подорожі чи гумор), стрічка швидко наповнюється контентом цієї категорії. Через деякий час, якщо інтерес зникає, ваги відповідних характеристик знижуються[16].

Така система демонструє приклад найбільш динамічної форми оновлення профілю: зміни відбуваються не щодня і не щогодини, а впродовж *секунд*, залежно від реакції користувача.

2.3 Порівняльний аналіз методів адаптації до змін поведінки користувачів

2.3.1. Групи методів адаптації

Динаміка інтересів користувача є ключовим фактором, який визначає ефективність системи рекомендацій. Глибина й точність профілю користувача залежить від того, наскільки швидко система здатна виявити зміни у перевагах, відфільтрувати випадкові відхилення та інтегрувати стабільні патерни поведінки у модель. У цьому підрозділі здійснено порівняльний аналіз основних методів, які на сьогодні застосовують для адаптації до змін поведінки, із визначенням їх сильних сторін, обмежень і можливостей комбінування.

Поведінка користувачів у цифрових платформах має стохастичний, нелінійний характер. На вибір впливають десятки чинників: ситуативний

контекст (час доби, тип пристрою), короткочасні інтереси, довгострокові вподобання, вплив середовища, рекомендації від друзів, рекламні події тощо.

Унаслідок цього формуються два різні часові рівні поведінки:

- довгостроковий профіль - стабільні й інертні захоплення;
- короткостроковий контекст - тимчасові зсуви інтересів, активні на горизонті від кількох хвилин до кількох днів[8; 12; 16].

Тому центральним викликом є побудова такої моделі, яка:

- 1) враховує довготривалі вподобання, що формують основу персоналізації;
- 2) миттєво реагує на зміни, уникаючи "залипання" на старі патерни;
- 3) відкидає випадковий шум, що не є ознакою зміни смаку.

У сучасній практиці виділяють такі основні підходи:

1. Методи з часовим зважуванням (Time Decay Models).
2. Моделі з ковзним вікном (Sliding Window).
3. Методи з оновленням параметрів у потоковому режимі (Online/Incremental Learning).
4. Контекстно-орієнтовані моделі (Context-Aware Adaptation).
5. Методи виявлення "режимів" поведінки (User State Tracking).
6. RL-орієнтовані методи адаптації (Reinforcement Learning for Preference Evolution).
7. Гібридні багаторівневі моделі (Hybrid Adaptive Frameworks).

Кожен підхід має свої переваги, а вибір залежить від характеру платформи, швидкості оновлення даних, критичності точності та ресурсних обмежень[3–4; 8; 10; 12; 16].

2.3.2. Методи з часовим зважуванням (Time Decay Models)

У традиційних моделях усі взаємодії користувача часто розглядаються як рівнозначні, незалежно від часу. Однак з поведінкової точки зору давні взаємодії

мають меншу релевантність, тому моделі з часовим згасанням застосовують експоненційні або лінійні функції ваги.

Одна з найпоширеніших формул:

$$w(t) = e^{-\lambda(T-t_i)} \quad (2.1)$$

де

- T - поточний час,
- t_i - час взаємодії,
- λ - коефіцієнт швидкості згасання.

Таке вагове коригування дозволяє підсилювати значення нових взаємодій. Застосування часових вагових коефіцієнтів має як практичні переваги, так і методологічні обмеження. Зокрема, такі моделі забезпечують оперативну адаптацію до змін у поведінці користувачів, оскільки нові взаємодії отримують більшу вагу порівняно з давніми.

Крім того, відповідний підхід відзначається відносною простотою реалізації та добре масштабується, що робить його доцільним для використання в потокових системах обробки даних. Водночас часові моделі згасання мають низку недоліків. Вони ігнорують внутрішню структуру поведінкових послідовностей і типи взаємодій, розглядаючи всі спостереження як рівнозначні, за винятком часової характеристики.

Це може призводити до надмірної чутливості до короткочасних, випадкових сплесків інтересу, які не відображають стійких уподобань користувача. Окрім того, ефективність методу значною мірою залежить від вибору коефіцієнта згасання λ , значення якого зазвичай визначається емпірично та потребує ручного налаштування. [8; 12]

2.3.3. Моделі з ковзним вікном (Sliding Window Models)

У цьому методі система зберігає й аналізує лише взаємодії, що відбулися у межах певного інтервалу:

$$\Delta t = T - t_i \leq W \quad (2.2)$$

де W - ширина вікна (наприклад 1 день, 1 тиждень, 30 останніх дій).

У результаті застосування цього підходу формується динамічний короткостроковий профіль користувача, який автоматично оновлюється зі зміною поточного часу. Такий механізм забезпечує явний контроль над актуальністю профілю, легко реалізується на практиці та ефективно зменшує вплив застарілих уподобань на результати рекомендацій. Разом із тим метод має суттєві обмеження. Дані, що виходять за межі заданого часового вікна, повністю виключаються з аналізу, навіть якщо вони відображають важливі та стабільні інтереси користувача. Крім цього, якість рекомендацій істотно залежить від вибору ширини вікна, параметр якого зазвичай визначається емпірично. Також модель не враховує різницю в інтенсивності взаємодій усередині вікна, що може призводити до втрати частини релевантної поведінкової інформації. [8; 12]

2.3.4. Порівняння Time Decay та Sliding Window

Таблиця 2.1

Критерій	Time Decay	Sliding Window
Врахування старих даних	Так, зі зменшеною вагою	Ні, видаляються
Реакція на зміни	Помірна	Швидка
Контроль “шуму”	Середній	Низький
Гнучкість налаштувань	Висока	Середня
Точність при різких змінах	Нижча	Вища

Обидва методи вважаються базовими, проте в системах із динамічною поведінкою (YouTube, TikTok) використовують складніші варіанти, де ці моделі комбінуються.[4; 5; 8; 10; 16]

2.3.5. Методи з оновленням параметрів у потоковому режимі (Online / Incremental Learning)

У багатьох сучасних рекомендаційних системах нові дані надходять постійно: перегляди, кліки, пошукові запити, взаємодії з контентом. Тому

застосування моделей, які потребують повного перенавчання, є неефективним, оскільки створює затримки та значні витрати ресурсів. Альтернативою є потокове (online) навчання, де модель оновлюється частковими кроками відповідно до кожної нової взаємодії.

Online Matrix Factorization

Традиційна матрична факторизація ґрунтується на мінімізації функції похибки:

$$\min_{U,V} \sum_{(i,j)} (r_{ij} - U_i^T V_j)^2 + \lambda(\|U\|^2 + \|V\|^2) \quad (2.3)$$

де U_i - вектор користувача, V_j - вектор елемента.

В online-версії параметри U_i і V_j оновлюються лише для пари (i, j) , для якої отримано нову взаємодію:

$$U_i \leftarrow U_i + \eta(e_{ij}V_j - \lambda U_i) \quad (2.4)$$

$$V_j \leftarrow V_j + \eta(e_{ij}U_i - \lambda V_j) \quad (2.5)$$

де

- η - швидкість навчання,
- e_{ij} - помилка прогнозу.

Запропонований підхід добре масштабується в умовах інтенсивного надходження даних і демонструє високу результативність у системах з великим обсягом подій. Він забезпечує практично миттєву адаптацію до змін у поведінці користувачів і дає змогу працювати з актуальними даними в режимі реального часу без необхідності повного переобчислення профілю.

Водночас використання такого підходу пов'язане з певними ризиками. Модель може надмірно підлаштовуватися під короткочасні поведінкові зрушення, що негативно позначається на узагальнювальній здатності. Для збереження стабільності навчання зазвичай доводиться вводити додаткові обмеження, зокрема на норму векторних представлень. Крім того, коректна робота моделі значною мірою залежить від точного вибору швидкості навчання, який потребує уважного налаштування.

Online Learning у контексті нейронних мереж. У deep-learning рекомендаційних системах (Wide&Deep, YouTube DNN, LightGCN) застосовують механізм mini-batch online updates, коли на кожній мікропартії нового трафіку виконується часткове донавчання моделі. Щоб уникнути “катастрофічного забування” старих знань, застосовують регуляризацію, replay-буфери та повільні “тіньові” копії моделі[4; 7–9].

2.3.6. Контекстно-орієнтовані методи адаптації (Context-Aware Models)

Зміна поведінки користувача часто пов’язана зі зміною контексту: часу доби, типу пристрою, місцезнаходження, активності, соціального оточення. Тому сучасні рекомендаційні платформи включають контекст у профіль користувача та використовують моделі, які динамічно реагують на контекстні переходи[4; 8; 12].

В Factorization Machines з контекстними ознаками контекст додається до моделі як додаткові вектори факторів:

$$\hat{y} = w_0 + \sum_i w_i x_i + \sum_{i < j} v_i v_j x_i x_j \quad (2.6)$$

де контекстні змінні можуть включати:

- time-of-day,
- day-of-week,
- device type,
- session duration,
- останні дії у сесії.

У ряді платформ (зокрема Google Discover) користувач моделюється не як єдиний вектор, а як набір станів, кожен з яких активується у власному контексті:

- “вечірній користувач” - більш схильний до розважального контенту,
- “користувач у дорозі” - короткі відео або аудіоконтент,
- “робочий режим” - інформаційні запити[4; 12].

Система автоматично змінює режим роботи на основі аналізу характерних патернів користувацьких взаємодій. Контекстно-орієнтований підхід забезпечує підвищену точність у короткострокових сценаріях використання та дозволяє коректно враховувати різні типи активності користувача без явного задання правил перемикавання. Така властивість робить його особливо придатним для застосування в мобільних і мультимодальних рекомендаційних системах.

Разом із тим ефективність цього підходу значною мірою залежить від якості доступних контекстних даних. За наявності шуму або неповних спостережень можливі помилкові переходи між режимами, що знижує стабільність рекомендацій. Крім того, у порівнянні з простішими моделями, контекстні методи зазвичай вимагають складніших рішень для забезпечення масштабованості. [4; 12].

2.3.7. Методи відстеження “станів” користувача (User State Tracking)

У сучасних системах поведінка користувача моделюється як процес із прихованими станами. Такий підхід дозволяє врахувати настроєві зміни, зміну стилю споживання контенту, циклічні патерни, фази “активності/пасивності” [4; 8]. Модель Hidden Markov Models (HMM) припускає існування множини прихованих станів:

$$S = \{s_1, s_2, \dots, s_k\} \quad (2.7)$$

і матриці переходів:

$$A = [a_{ij}] \quad (2.8)$$

де a_{ij} - ймовірність переходу зі стану s_i до стану s_j .

Стан визначає розподіл користувацьких дій. При зміні стану рекомендації автоматично адаптуються [3; 12].

Neural State-space Models

У глибоких системах замість HMM застосовують рекурентні та трансформерні моделі:

– GRU4Rec,

- SASRec,
- TiSASRec (Time Interval Self-Attention),
- Mamba4Rec (selective state-space models).

Ці моделі здатні:

- виявляти періодичні та спорадичні зміни,
- фіксувати довгі залежності у поведінці,
- узгоджувати короткостроковий і довгостроковий контексти.

Підхід, заснований на моделюванні станів, дає змогу описувати кілька різних режимів поведінки користувача в межах єдиної моделі та гнучко переходити між ними. Він демонструє високу результативність у рекомендаційних системах зі складною динамікою споживання контенту, зокрема на платформах на кшталт YouTube чи TikTok, а також добре поєднується з методами підкріплювального навчання та графовими моделями. При цьому застосування такого підходу супроводжується суттєвими ускладненнями. Побудова й навчання моделей станів потребують значних обчислювальних ресурсів і великих обсягів історичних даних. Крім того, отримані результати зазвичай складніше піддаються інтерпретації порівняно з простішими поведінковими моделями. [4; 7–9; 16]

2.3.8. Графові методи адаптації профілю користувача

У сучасних рекомендаційних системах дедалі більшого поширення набувають графові моделі. Вони дозволяють описувати користувачів, об'єкти, категорії, сесії, події та інші компоненти у вигляді вузлів графа, а взаємодії - у вигляді ребер. Зміна поведінки користувача в такій системі відображається у локальних змінах структури графа, що відкриває можливості для дуже гнучкого оновлення профілю[6].

Графові нейронні мережі (GraphSAGE, GAT, LightGCN) дозволяють “поширювати” інформацію між вузлами таким чином, що зміна в одному вузлі впливає на його сусідів. Основна формула агрегації ознак сусідів

$$h_v^{(k)} = \sigma(W^{(k)} \cdot \text{AGG}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\})) \quad (2.9)$$

де

- $h_v^{(k)}$ - представлення вузла на k-му шарі,
- $\mathcal{N}(v)$ - множина сусідів вузла,
- AGG - агрегатор (mean, sum, attention),
- $W^{(k)}$ - матриця параметрів шару.

Коли користувач змінює поведінку (переглядає нові жанри, переходить у нові тематичні області), у графі з'являються нові ребра, і модель автоматично оновлює представлення вузла користувача[6].

У багатьох платформах застосовуються динамічні графові моделі, де ребра мають часові мітки:

$$G_t = (V, E_t) \quad (2.10)$$

а рекомендаційна модель обчислює:

$$h_v^{(t)} = f(h_v^{(t-1)}, \Delta E_t) \quad (2.11)$$

тобто інкрементально оновлює лише те, що змінилося.

Графові методи вирізняються здатністю зберігати стабільність у разі нерегулярних або нелінійних змін користувацької поведінки. Вони дозволяють явно моделювати складні залежності, зокрема переходи між категоріями контенту, соціальні впливи між користувачами в рекомендаційних системах із соціальною складовою, а також тематичні траєкторії поведінки. Окремою перевагою є можливість роботи з великими структурованими даними, що охоплюють графи з мільйонами вузлів і зв'язків між ними. Разом із тим практичне впровадження графових підходів пов'язане з високими вимогами до обчислювальних ресурсів і складної інфраструктури.

Такі методи зазвичай потребують використання спеціалізованих графових бібліотек або баз даних, зокрема DGL, PyG, Neo4j чи DeepGraph. Крім того, результати роботи графових моделей, як правило, є менш прозорими для інтерпретації порівняно з класичними фільтраційними методами.

2.3.9. Методи адаптації на основі навчання з підкріпленням (Reinforcement Learning, RL)

У задачах рекомендацій користувач часто представлений як “оточення”, а система - як “агент”, який намагається максимізувати довгострокову корисність. Зміна поведінки користувача розглядається як зміна середовища, що вимагає від агента гнучкої реакції.

Опис задачі відбувається через MDP:

- S - множина станів (поточні інтереси користувача),
- A - множина дій (які рекомендації показати),
- R - функція винагороди (кліки, перегляд, утримання уваги),
- T - ймовірності переходів між станами (зміна поведінки).

Ціль:

$$\max_{\pi} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t] \quad (2.12)$$

де π - політика рекомендацій[17–18].

Адаптивність RL: Q-learning / Deep Q-Networks (DQN). Навчання відбувається за формулою:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (2.13)$$

що дозволяє системі швидко пристосовуватися до нових переходів між станами.

Policy Gradient (PG, PPO). Політика оновлюється так:

$$\nabla J(\theta) = \mathbb{E}[\nabla_{\theta} \log \pi_{\theta}(a | s) A(s, a)] \quad (2.14)$$

де $A(s, a)$ - перевага дії[17–18].

Застосування методів навчання з підкріпленням у задачах моделювання динамічної поведінки користувачів дає змогу враховувати довгострокові інтереси, зокрема в сценаріях із тривалими сесіями та фокусом на утримання уваги. Такий підхід підтримує оптимізацію складних цільових функцій, серед яких можуть бути час перегляду, показники утримання або баланс і «здоров'я» контентного каталогу.

Важливою властивістю є здатність моделі адаптуватися до змін у поведінці користувача без необхідності повного перенавчання. Водночас практичне використання навчання з підкріпленням супроводжується низкою труднощів. Налаштування моделей і вибір оптимальних гіперпараметрів є складним і ресурсоємним процесом. Навчання зазвичай потребує значного часу та обчислювальних витрат. Крім того, за умов швидких або різких змін поведінки можливі проблеми зі стабільністю навчального процесу.

2.3.10. Метрики адаптивності та способи оцінки

Аналіз методів адаптації неможливий без чітких метрик, які дозволяють оцінити, наскільки швидко й ефективно система реагує на зміни користувача.

1. Метрики швидкості адаптації. Time-to-Adapt (ТТА) показує, скільки взаємодій потрібно системі, щоб стабілізувати прогноз після зміни. Recovery Error (RE) визначається формулою:

$$RE = \frac{1}{N} \sum_{i=1}^N | r_i^{(new)} - \hat{r}_i^{(t+i)} | \quad (2.15)$$

де

- $r_i^{(new)}$ - справжні оцінки після зміни,
- $\hat{r}_i^{(t+i)}$ - прогноз системи.

2. Метрики стійкості (robustness). Volatility Penalty (VP) визначає, наскільки модель “перекочується” між станами без вагомих причин:

$$VP = \sum | \Delta h_u | \quad (2.16)$$

де Δh_u - зміна вектора користувача між кроками.

3. Метрики довгострокової ефективності:

- Retention@K;
- Session Length Improvement;
- Sustained Engagement Score[17–18].

2.3.11. Порівняльна таблиця методів адаптації

Таблиця 2.2

Порівняння основних методів адаптації до змін поведінки

Метод	Швидкість реакції	Стійкість	Обчислювальна складність	Пояснюваність
Time-decay	Висока	Низька	Низька	Висока
Sliding window	Висока	Середня	Середня	Висока
Онлайн-факторизація	Висока	Середня	Середня	Середня

Продовження таблиці 2.2

Порівняння основних методів адаптації до змін поведінки

Метод	Швидкість реакції	Стійкість	Обчислювальна складність	Пояснюваність
Глибинні послідовні моделі	Середня	Висока	Висока	Низька
Графові методи	Середня	Висока	Дуже висока	Низька
RL	Залежить від налаштування	Дуже висока	Дуже висока	Низька

2.4 Математична модель динамічного оновлення профілю користувача

2.4.1 Загальна концепція моделювання профілю користувача

У сучасних рекомендаційних системах профіль користувача розглядається як формальна структура, що відображає його інтереси, поведінкові закономірності, поточні та довгострокові вподобання. На відміну від статичних моделей, динамічні профілі змінюються у часі відповідно до того, як оновлюється поведінка користувача, змінюється контекст його взаємодії та з'являються нові сигнали.

У найзагальнішому вигляді профіль користувача можна подати як вектор у багатовимірному просторі характеристик:

$$P_t = (p_{t,1}, p_{t,2}, \dots, p_{t,n}) \quad (2.17)$$

де

– P_t - профіль користувача у момент часу t ;

– $p_{t,i}$ - оцінка інтересу до характеристики i у момент часу t ;

– n - загальна кількість характеристик (тематики, жанри, функціональні категорії, типи активності тощо).

Кожна зміна у поведінці користувача генерує нову подію I_t , що також представлена у вигляді вектора:

$$I_t = (i_{t,1}, i_{t,2}, \dots, i_{t,n}) \quad (2.18)$$

Задачею математичної моделі є визначення способу переходу від старого профілю P_{t-1} до нового профілю P_t з урахуванням як короткочасних, так і довготривалих змін. Тому динамічний профіль трактують як систему, що еволюціонує за правилами:

$$P_t = F(P_{t-1}, I_t, \Theta_t) \quad (2.19)$$

де

– $F(\cdot)$ - функція оновлення,

– Θ_t - параметри адаптації (ваги, коефіцієнти, швидкість навчання).

2.4.2 Комбінована модель короткострокових та довгострокових інтересів

Короткострокові та довгострокові інтереси користувача мають різну природу та різний вплив:

– короткочасні інтереси (short-term interests) швидко змінюються і відображають актуальний контекст (теми пошуку, тренди, тимчасові потреби);

– довгострокові інтереси (long-term interests) еволюціонують повільніше та відповідають стійким вподобанням.

Тому доцільно представити профіль як комбінацію двох підпрофілів:

$$P_t = w_t \cdot P_t^{short} + (1 - w_t) \cdot P_t^{long} \quad (2.20)$$

де

– P_t^{short} - короткостроковий профіль,

– P_t^{long} - довгостроковий профіль,

– w_t - коефіцієнт ваги короткострокового профілю, що змінюється динамічно.

Вага w_t залежить від того, наскільки стабільною є поведінка користувача у певний проміжок часу. Наприклад:

$$w_t = \sigma(\beta \cdot S_t) \quad (2.21)$$

де

– S_t - показник зміни поведінки (наприклад, швидкість зміни категорій контенту або диверсифікація дій),

– β - параметр чутливості,

– $\sigma(\cdot)$ - сигмоїдна функція.

Інтуїтивне припущення: якщо користувач різко змінює інтереси $\rightarrow S_t$ великий $\rightarrow w_t \rightarrow 1 \rightarrow$ система спирається на короткостроковий профіль; якщо поведінка стабільна $\rightarrow w_t \rightarrow 0 \rightarrow$ домінує довгостроковий профіль.

2.4.3 Оновлення короткострокового та довгострокового профілів

Короткостроковий профіль моделює миттєві дії користувача і оновлюється за схемою експоненційного згасання:

$$P_t^{short} = (1 - \alpha)P_{t-1}^{short} + \alpha I_t \quad (2.22)$$

Параметр $\alpha(0-1)$ регулює швидкість "забування":

- якщо $\alpha \rightarrow 1 \rightarrow$ модель швидко реагує на нові дії;
- якщо $\alpha \rightarrow 0 \rightarrow$ короткостроковий профіль згладжується.

Цей механізм добре описує ситуації на кшталт:

- короткочасний інтерес до певної теми;
- інтенсивні зміни поведінки;
- поведінкові флуктуації.

Довгостроковий профіль змінюється повільніше. Для нього застосовується інша схема:

$$P_t^{long} = (1 - \gamma)P_{t-1}^{long} + \gamma I_t \quad (2.23)$$

де $\gamma \ll \alpha$.

Типові значення:

- $\alpha = 0.2 - 0.5$,
- $\gamma = 0.01 - 0.05$.

Це дозволяє системі:

- не реагувати надто швидко на випадкові дії,
- фіксувати стабільні інтереси,
- розрізнити короткі сплески активності та справжні зміни.

Підставивши формули короткострокового та довгострокового оновлення у загальну формулу отримуємо:

$$P_t = w_t[(1 - \alpha)P_{t-1}^{short} + \alpha I_t] + (1 - w_t)[(1 - \gamma)P_{t-1}^{long} + \gamma I_t] \quad (2.24)$$

2.4.4 Оцінювання впливу нових і старих взаємодій у моделі з часовими вагами

Однією з ключових вимог до моделі динамічного оновлення профілю є здатність не лише додавати нові дані, а й коректно перерозподіляти їхній вплив відносно старішої інформації. Це означає, що система повинна математично визначати, яка частина історичних взаємодій залишається релевантною, а які сигнали варто вважати застарілими. Для цього використовуються часові вагові коефіцієнти (temporal decay weights).

Нехай існує послідовність взаємодій користувача з системою:

$$I = \{(v_1, t_1), (v_2, t_2), \dots, (v_n, t_n)\} \quad (2.25)$$

де

- v_i - вектор ознак взаємодії (наприклад, жанр переглянутого фільму, категорія відео, тривалість перегляду),
- t_i - час взаємодії.

Кожній події ставиться у відповідність ваговий коефіцієнт:

$$w_i = e^{-\lambda(T-t_i)} \quad (2.26)$$

де

- T - поточний момент часу,

- λ - параметр згасання, що визначає швидкість втрати релевантності даних.

Таким чином, модель зводиться до задачі обчислення зваженої сукупності взаємодій:

$$P(t) = \frac{\sum_{i=1}^n w_i v_i}{\sum_{i=1}^n w_i} \quad (2.27)$$

Ця формула забезпечує дві важливі властивості:

1. Актуальна поведінка домінує над історичною. Взаємодії, що відбулися нещодавно, автоматично отримують більшу вагу.
2. Профіль користувача оновлюється плавно. Поступове зменшення впливу старих подій запобігає різким змінам у рекомендаціях при появі поодиноких нових сигналів.

Для рекомендаційних систем із частими взаємодіями (YouTube, TikTok, Spotify) значення параметра λ може бути відносно великим, оскільки поведінка користувачів швидко змінюється. Для систем із повільною динамікою (освітні платформи, професійні сервіси) - меншим, щоб зберігати стабільність профілю.

2.4.5 Інтеграція короткострокових та довгострокових вподобань у єдиній моделі

Окремим аспектом розробки математичної моделі є поєднання короткострокових (short-term) та довгострокових (long-term) уподобань. На практиці користувач може тимчасово проявляти інтерес до певних категорій вмісту, що не відображають його сталі смаки. Тому проста модель із часовими вагами може бути недостатньо стабільною: надмірне підсилення короткострокових дій призведе до втрати цілісності довгострокового профілю. У загальному вигляді профіль складається з двох компонентів:

$$P_{\text{total}}(t) = \alpha P_{\text{long}}(t) + (1 - \alpha) P_{\text{short}}(t) \quad (2.28)$$

де

- $P_{\text{long}}(t)$ - довгостроковий профіль (стійкі вподобання),
- $P_{\text{short}}(t)$ - короткостроковий профіль (тимчасові інтереси),

– $\alpha \in [0,1]$ - коефіцієнт балансу.

Формування довгострокового профілю

Для $P_{\text{long}}(t)$ застосовуються повільні часові ваги:

$$w_i^{(L)} = e^{-\lambda_L(T-t_i)}, \lambda_L \ll \lambda_S \quad (2.29)$$

тобто повільне згасання.

Формування короткострокового профілю: для $P_{\text{short}}(t)$ - швидкі ваги:

$$w_i^{(S)} = e^{-\lambda_S(T-t_i)}, \lambda_S \gg \lambda_L \quad (2.30)$$

Таким чином модель може одночасно враховувати:

- сталі смаки (наприклад, користувач завжди дивиться наукову фантастику),
- тимчасові захоплення (динаміка переглядів дитячих мультфільмів протягом тижня при присутності дитини).

Це дозволяє рекомендаційній системі адаптуватися до контексту, зберігаючи загальну стабільність.

2.4.6 Адаптивний механізм оновлення параметра балансу α

Фіксоване значення α не завжди доречно, оскільки інтенсивність змін поведінки користувача може відрізнятись в різні періоди. У години активного використання короткостроковий профіль варто посилювати, тоді як при стабільній поведінці - послаблювати.

Для цього вводиться адаптивний механізм:

$$\alpha(t) = 1 - \frac{\sigma(t)}{\sigma_{\text{max}}} \quad (2.31)$$

де

- $\sigma(t)$ - ступінь мінливості поведінки користувача за останній період,
- σ_{max} - максимальне значення варіабельності для нормування.

Оцінювання мінливості може проводитись через:

- дисперсію тематичних векторів,
- кількість різнорідних категорій контенту,

– швидкість зміни інтересів за останні k взаємодій.

У разі високої мінливості ($\sigma(t) \rightarrow \sigma_{\max}$) модель автоматично:

$$\alpha(t) \rightarrow 0 \Rightarrow P_{\text{short}}(t) \text{ домінує.}$$

У періоди стабільної поведінки:

$$\alpha(t) \rightarrow 1 \Rightarrow P_{\text{long}}(t) \text{ домінує.}$$

Таким чином система самостійно регулює баланс між довгостроковим і короткостроковим компонентами, забезпечуючи більш точну та контекстно залежну адаптацію.

2.4.7 Модель інтеграції контекстних факторів у процес оновлення профілю

Поведінка користувача в рекомендаційній системі не є статичною - вона залежить від зовнішніх умов, які визначають контекст взаємодії. Дослідження у сфері персоналізованого контенту демонструють, що такі параметри, як час доби, день тижня, пристрій доступу, місце перебування або поточний емоційний стан, суттєво змінюють характер взаємодії з контентом. Тому математична модель оновлення профілю має враховувати контекст не як доповнення, а як окремий структурний компонент, що впливає на результат обчислення.

У загальному вигляді контекст описується як вектор:

$$C(t) = [c_1(t), c_2(t), \dots, c_m(t)] \quad (2.32)$$

де

- $c_j(t)$ - значення конкретного контекстного параметра у момент часу t ;
- m - кількість факторів, що враховуються системою.

Типові контекстні параметри:

- час доби - $c_1(t)$;
- день тижня - $c_2(t)$;
- тип пристрою (мобільний/ПК) - $c_3(t)$;
- активність користувача (споживання відео, пошук, перегляд рекомендацій) - $c_4(t)$;

- тривалість сесії - $c_5(t)$;
- місце перебування - $c_6(t)$;
- інтенсивність взаємодій - $c_7(t)$.

Щоб включити контекст у модель оновлення профілю, вводиться контекстна функція впливу:

$$\phi(v_i, C(t_i)) = v_i \odot g(C(t_i)) \quad (2.33)$$

де

- $g(C(t_i))$ - функція модифікації ознак взаємодії залежно від контексту,
- \odot - поелементне множення.

Це дозволяє підсилювати або послаблювати вплив конкретної взаємодії залежно від умов її здійснення. Наприклад, перегляд розважального контенту увечері може мати більшу вагу, ніж у робочий час; пошук навчальних матеріалів у будні має іншу семантику, ніж у вихідні.

Фінальна формула з урахуванням контексту набуває вигляду:

$$P(t) = \frac{\sum_{i=1}^n w_i \phi(v_i, C(t_i))}{\sum_{i=1}^n w_i} \quad (2.34)$$

Таким чином профіль користувача стає не просто відображенням його взаємодій, а контекстно чутливою структурою, здатною моделювати реальні зміни поведінки.

2.4.8 Узагальнена модель багатofакторного оновлення профілю

Зважаючи на розглянуті аспекти - часові ваги, коротко- і довгострокові вподобання, контекстні фактори - можна сформулювати узагальнену модель, що відображає весь процес адаптації рекомендаційної системи до змін поведінки користувача.

Модель складається з трьох взаємопов'язаних рівнів:

1. Рівень обробки взаємодій (interaction-level).

На цьому рівні відбувається первинне формування опису взаємодії:

$$u_i = \phi(v_i, C(t_i)) \quad (2.35)$$

2. Рівень часової динаміки (temporal-level).

Для кожної взаємодії обчислюється ваговий коефіцієнт:

$$w_i = e^{-\lambda(T-t_i)} \quad (2.36)$$

Для двох компонент моделі:

$$w_i^{(L)} = e^{-\lambda_L(T-t_i)}, w_i^{(S)} = e^{-\lambda_S(T-t_i)} \quad (2.37)$$

3. Рівень обчислення динамічного профілю (profile-level).

Короткостроковий профіль:

$$P_{\text{short}}(t) = \frac{\sum_i w_i^{(S)} u_i}{\sum_i w_i^{(S)}} \quad (2.38)$$

Довгостроковий профіль:

$$P_{\text{long}}(t) = \frac{\sum_i w_i^{(L)} u_i}{\sum_i w_i^{(L)}} \quad (2.39)$$

Адаптивний баланс:

$$\alpha(t) = 1 - \frac{\sigma(t)}{\sigma_{\max}} \quad (2.40)$$

Підсумковий профіль:

$$P_{\text{total}}(t) = \alpha(t)P_{\text{long}}(t) + (1 - \alpha(t))P_{\text{short}}(t) \quad (2.41)$$

Ця формула становить основу динамічної моделі оновлення, яка здатна враховувати нелінійність поведінки користувача, різну швидкість зміни інтересів, вплив середовища та інтенсивність використання системи.

Розроблена модель вирізняється низкою важливих властивостей:

1. Адаптивність. Профіль автоматично змінює структуру залежно від характеру сигналів та контексту.
2. Стійкість до шуму. Поодинокі взаємодії не спричиняють критичних змін рекомендацій.
3. Чутливість до різких зсувів інтересів. При високій мінливості поведінки $\alpha(t)$ зменшується, надаючи пріоритет короткостроковим уподобанням.

4. Балансування історії та актуального контенту. Довгострокові вподобання не зникають навіть за значних змін у поведінці.
5. Гнучкість. Модель дозволяє додавати нові фактори - поведінкові, контекстні або класифікаційні - без зміни її структури.

3 РОЗРОБКА МЕТОДИКИ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА

3.1 Опис запропонованої методики динамічного оновлення профілю користувача

Запропонована методика динамічного оновлення профілю користувача ґрунтується на концепції безперервної адаптації рекомендаційної моделі до зміни поведінки користувача. Підхід поєднує можливості потокової обробки подій, онлайн-оновлення параметрів профілю та розподіленої обчислювальної архітектури, що забезпечує масштабованість і низький час відгуку системи. Методика орієнтована на роботу у реальних умовах, де реакція на зміну інтересів має відбуватися протягом мілісекунд. Одночасно система зберігає інформацію про довгострокові вподобання, що дозволяє формувати збалансовані рекомендації: не лише «миттєвий» контент, а й ті елементи, які відповідають загальному профілю користувача.

3.1.1 Архітектура запропонованої методики

Архітектура методики складається з кількох компонентів, що працюють у режимі конвеєрної обробки даних. Вона побудована за принципом модульної гнучкості й може масштабуватися до мільйонів користувачів.

Основні компоненти архітектури

1. Модуль збору подій (Event Collector)

Приймає події від клієнтських застосунків - веб-інтерфейсу, мобільних додатків або серверних служб. Збір подій може бути реалізований через:

- HTTP-API,
- WebSocket-підключення,
- черги повідомлень (Kafka, RabbitMQ),
- серверні логери.

Подія має уніфіковану форму:

{user_id, item_id, event_type, timestamp, session_id, context}

2. Поточковий процесор (Stream Processor) виконує первинну обробку вхідних даних:

- перетворення подій у стандартизовані структури;
- видалення дублікованих сигналів;
- агрегація у мікробатчі;
- визначення релевантності події;
- фільтрація технічних і випадкових подій.

3. Сховище профілів користувачів (User Profile Store) містить сукупність параметрів, що описують вподобання користувача:

- вектор довгострокових інтересів (LTI);
- вектор короткострокових інтересів (STI);
- контекстні параметри;
- часові коефіцієнти;
- історичні статистики.

Зберігання може бути організоване у вигляді:

- Redis / MongoDB - для швидких оновлень;
- Cassandra / Bigtable - для розподілених даних.

4. Модуль оновлення профілю (Profile Update Engine) застосовує математичну модель, описану у розділі 2.3. Його завдання:

- розрахунок ваги нової події;
- визначення, як подія впливає на STI та LTI;
- адаптація профілю згідно з коефіцієнтами згасання;
- визначення нових значень параметрів профілю.

5. Модуль рекомендацій (Recommendation Engine) після кожного оновлення профілю може виконувати:

- фільтрацію доступного контенту;
- ранжування елементів;

– формування рекомендованих списків.

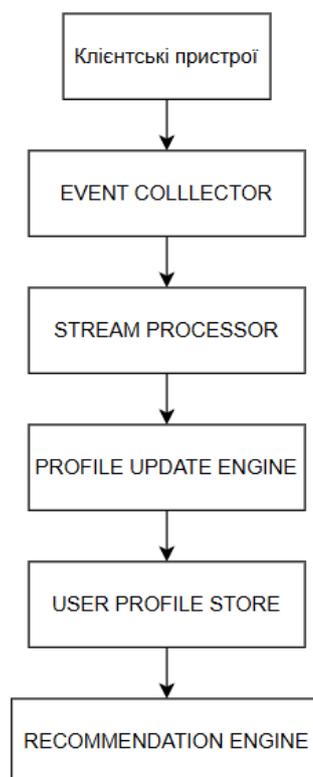


Рис. 3.1 Схема архітектури

3.1.2 Алгоритм динамічного оновлення профілю

Алгоритм базується на поступовому (інкрементальному) оновленні параметрів профілю. Профіль реагує на кожну нову подію, але при цьому не потребує повного перенавчання.

Етапи алгоритму:

1. Отримання події. Система фіксує факт взаємодії користувача з контентом.
2. Класифікація події. Визначається її тип: перегляд, лайк, коментар, пошук, пропуск.
3. Обчислення початкової ваги події. Згідно з таблицею типів подій.
4. Корекція ваги з урахуванням контексту.

5. Застосування коефіцієнта згасання (time-decay). Вплив старих подій зменшується експоненційно.
6. Оновлення короткострокових інтересів (STI).
7. Оновлення довгострокових інтересів (LTI).
8. Збереження оновленого профілю.

3.1.3 Робота компонентів та їхня взаємодія

Взаємодія компонентів відбувається у режимі асинхронної та паралельної обробки. Компоненти обмінюються повідомленнями через черги, що забезпечує:

- стійкість до відмов окремих сервісів;
- відсутність блокувань;
- можливість горизонтального масштабування;
- мінімальну затримку доставки подій.

Основний цикл обробки подій виглядає так:

1. Користувач здійснює дію.
2. Подія надходить у Event Collector.
3. Поточковий процесор визначає її значущість.
4. Модуль оновлення профілю адаптує STI та LTI.
5. Профіль одразу стає доступним для Recommendation Engine.
6. Користувач бачить оновлений список рекомендацій.

3.1.4 Обробка поточкових даних

Потоковий підхід у методиці передбачає:

1. Мікробатчинг та черги повідомлень - підвищує пропускну здатність системи.
2. Детекцію шумових подій. Наприклад:
 - перегляд тривалістю < 1 секунди;
 - повторні кліки без зміни контенту;
 - технічні події браузера.

3. Виявлення трендової поведінки - якщо користувач різко змінює тип контенту, STI отримує підсилений коефіцієнт.

4. Злиття сесійних та довгострокових даних. Короткострокові інтереси відображають тенденції поточної сесії, а довгострокові - загальну картину уподобань.

3.1.5 Таблиця типів подій та їхніх ваг

Таблиця 3.1

Таблиця типів подій та їхніх ваг

Тип події	Опис	Початкова вага
Перегляд < 30%	Коротка взаємодія, потенційно випадкова	0.1
Перегляд 30-80%	Часткова зацікавленість	0.3
Перегляд > 80%	Висока зацікавленість	0.6
Лайк / «подобається»	Явно позитивний сигнал	1.0
Додавання в обране	Сильний стійкий інтерес	1.2
Пошуковий запит	Намір знайти конкретний тип контенту	0.5
Коментар	Висока емоційна залученість	0.9
Швидкий пропуск	Негативний сигнал	-0.4
Відмова від перегляду	Значна негативна реакція	-0.7

Вага може коригуватися контекстом (девайс, час, тривалість сесії) та time-decay.

3.2 Використані програмні засоби

Розробка системи динамічного оновлення профілю користувача вимагає застосування сучасних інструментів машинного навчання, потокової обробки

даних, аналітики та роботи з великими масивами інформації. Тому розглянуто програмні засоби, для реалізації запропонованої методики, наведено приклади їхнього застосування та обґрунтовано вибір кожного інструмента.

TensorFlow Recommenders (TFRS) - це бібліотека на основі TensorFlow, спеціально розроблена для побудови рекомендаційних систем. TFRS забезпечує:

- готові компоненти для retrieval та ranking моделей;
- інтеграцію з TensorFlow Extended (TFX);
- підтримку моделей, які легко оновлювати в режимі online або batch;
- можливість працювати з ембеддингами користувачів та об'єктів.

У контексті даної роботи TFRS використовується для:

- побудови базової рекомендаційної моделі;
- генерації ембеддингів користувача, що оновлюються динамічно;
- інтеграції з модулем Profile Update Engine.

Приклад моделі TFRS

```
import tensorflow as tf
import tensorflow_recommenders as tfrs
class RankingModel(tfrs.models.Model):
    def __init__(self, user_model, item_model):
        super().__init__()
        self.query_model = user_model
        self.candidate_model = item_model
        self.task = tfrs.tasks.Ranking(
            loss=tf.keras.losses.MeanSquaredError(),
            metrics=[tf.keras.metrics.TopK_categorical_accuracy(k=10)]
        )
    def compute_loss(self, features, training=False):
        user_embeddings = self.query_model(features["user_id"])
        item_embeddings = self.candidate_model(features["item_id"])
        return self.task(
```

```

labels=features["rating"],
predictions=tf.reduce_sum(user_embeddings * item_embeddings, axis=1),
)

```

Переваги використання TFRS:

- можливість розділяти модель на модулі, які можна оновлювати незалежно;
- зручна підтримка embeddings, що зберігаються у профілі користувача;
- чудова інтеграція з TensorFlow Serving для продуктивного розгортання[19].

PyTorch застосовується для гнучкого експериментування з архітектурами нейронних мереж, які використовуються для прогнозування короткострокових інтересів користувача.

У методиці PyTorch використовується для:

- моделювання короткострокової поведінки через послідовні моделі (LSTM, GRU, Transformer);
- створення lightweight-моделей, які легко оновлювати інкрементально;
- дослідження поведінкових патернів у сесійних даних[20].

Приклад моделі GRU на PyTorch:

```

import torch
import torch.nn as nn
class GRUInterestModel(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super().__init__()
        self.gru = nn.GRU(input_dim, hidden_dim, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)
    def forward(self, x):
        out, _ = self.gru(x)
        out = self.fc(out[:, -1, :])
        return out

```

Після кожної події модель може отримати один новий вектор x , виконати передбачення та оновити STI.

У рекомендаційній системі з динамічним профілем користувача критично важливо мати можливість:

- навчати моделі поступово (online learning), без повного перенавчання;
- швидко обробляти великі стріми подій;
- оновлювати ваги після кожної події.

River - одна з найкращих бібліотек для цього, бо підтримує:

- інкрементальні моделі класифікації;
- регресію;
- статистичні моделі;
- відстеження ковзних середніх;
- time-decay механізми[21].

Приклад інкрементального оновлення профілю

```
from river import linear_model, optim, preprocessing
model = preprocessing.StandardScaler() | linear_model.LinearRegression(
    optimizer=optim.SGD(0.01)
)
for event in stream_of_events:
    x, y = extract_features(event)
    model.learn_one(x, y)
```

Цей підхід дозволяє оновлювати модель *після кожної події*.

LightFM - це бібліотека для побудови гібридних моделей рекомендацій, яка поєднує:

- collaborative filtering;
- content-based методи;
- факторизаційні моделі.

Має підтримку навчання з різними функціями втрат:

- BPR;

- WARP;
- Logistic;
- KOS[22].

Приклад навчання моделі LightFM:

```
from lightfm import LightFM
model = LightFM(loss='warp')
model.fit(interactions, item_features=item_features, epochs=30)
```

У цьому проєкті LightFM використовується для:

- оцінки довгострокових інтересів (LTI);
- формування стабільних рекомендацій на основі історичних даних.

Бібліотека Implicit використовується для роботи з великими sparse-матрицями взаємодій. Особливо корисна для неявних рейтингів, де поведінка користувача (перегляд, лайк, збереження) перетворюється на позитивні/негативні сигнали. Підтримує:

- ALS (Alternating Least Squares);
- BPR;
- TF-IDF факторизації[23].

Приклад побудови ALS-моделі:

```
from implicit.als import AlternatingLeastSquares
model = AlternatingLeastSquares(factors=64)
model.fit(user_item_matrix)
```

Базові бібліотеки Pandas і NumPy використовуються для:

- очищення даних;
- структурування подій;
- розрахунку проміжних статистичних показників;
- підготовки даних перед подачею в модель[24; 25].

Приклад обробки подій:

```
import pandas as pd
df = pd.read_json("events.json")
```

```
df["weight"] = df["event_type"].apply(lambda x: weight_map[x])
df["hour"] = pd.to_datetime(df["timestamp"]).dt.hour
```

Apache Spark використовується для:

- розподіленої обробки великих потоків подій;
- агрегації логів;
- підготовки даних для навчання моделей;
- обробки даних у режимі near-real-time (Spark Streaming)[26].

Приклад Spark Streaming:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("StreamProcessing").getOrCreate()
events = spark.readStream.format("kafka").load()
processed = events.selectExpr("CAST(value AS STRING)")
query = processed.writeStream.outputMode("append").start()
query.awaitTermination()
```

У реалізації методики використані такі хмарні рішення:

1. AWS Kinesis / GCP PubSub / Azure EventHub - для передачі поточкових подій.
2. AWS Lambda / GCP Cloud Functions - для обробки подій без розгортання серверів.
3. Cloud Storage / S3 - для зберігання проміжних даних.
4. Vertex AI / AWS SageMaker - для розгортання моделей онлайн.

У процесі створення системи динамічного оновлення профілю користувача ключову роль відіграє не лише набір бібліотек і фреймворків, а й інструменти, за допомогою яких виконується розробка, тестування та налагодження програмного коду. У межах цієї роботи таким інструментом стало інтегроване середовище розробки PyCharm, яке забезпечує зручну інфраструктуру для організації проекту, підтримку фреймворків машинного навчання, інтеграцію з системами контролю версій та розгорнуту роботу з віртуальними середовищами. Використання PyCharm дозволяє:

- структурувати проєкт відповідно до модульної архітектури - окремі пакети для обробки поточкових подій, моделі рекомендацій, конфігурацій та інтеграції з базами даних;
- організувати роботу з віртуальними середовищами (venv, Conda), що забезпечує ізоляцію залежностей і можливість тестувати різні версії ML-модулів;
- налаштувати запуск компонентів системи як окремих конфігурацій (наприклад, сервісу обробки подій, сервісу рекомендацій та модуля онлайн-оновлення);
- застосовувати інтеграцію з Git для контролю змін і спільної роботи над кодом.

PyCharm підтримує спеціальні плагіни для TensorFlow, PyTorch і Apache Spark, що робить роботу з великими даними та моделями зручнішою.

Переваги використання PyCharm:

1. Інтелектуальний аналіз коду. Інструмент пропонує автоматичні підказки, пошук помилок у реальному часі, рефакторинг і навігацію по великих проєктах, що суттєво прискорює розробку.
2. Підтримка Jupyter Notebook. Це дозволило проводити окремі експерименти з фреймворками TensorFlow Recommenders, PyTorch та River без відриву від основного середовища.
3. Вбудовані інструменти налагодження. Система динамічного оновлення профілю працює в режимі real-time, тому функції покрокового дебагу стали критичними для тестування компонентів обробки подій.
4. Робота з базами даних. Через вбудований DB Browser здійснювався моніторинг таблиць профілів користувачів та подій, а також перевірка SQL-запитів до сховища.
5. Профілювання Python-коду. Оскільки алгоритми оновлення працюють у потоковому режимі, оцінка продуктивності (time profiling) була необхідною для визначення вузьких місць[27].

3.3 Реалізація системи

3.3.1 Архітектура модулів

Архітектура реалізованої системи складається з таких основних модулів:

1. Модуль обробки подій (Event Stream Processor)
Приймає події взаємодії користувача в режимі реального часу (перегляд контенту, клік, пошук, додавання у вибране). Виконує фільтрацію, нормалізацію та агрегування подій.

2. Модуль динамічного оновлення профілю (Profile Updater)
Реалізує математичну модель оновлення профілю та алгоритм динамічного зниження ваг історичних інтересів. Формує оновлений вектор стану користувача.

3. Модуль рекомендацій (Recommendation Engine)
Генерує рекомендації на основі оновленого профілю. Використовує комбіновані підходи: колаборативну фільтрацію (LightFM/Implicit), контентну оцінку та факторизацію латентних ознак.

4. Модуль сховища даних (Storage Layer)

Два типи сховищ:

- оперативне кеш-сховище для профілів, що часто оновлюються;
- персистентне сховище подій і історичних даних.

5. Модуль API та взаємодії з інтерфейсом (Application Layer)
Забезпечує доступ до рекомендацій через REST- або gRPC-API. Може використовуватися фронтендом, мобільним додатком або зовнішніми сервісами.

6. Модуль моніторингу та журналювання (Monitoring & Logging)
Відслідковує інтенсивність подій, затримки обробки, точність рекомендацій і стабільність алгоритмів.

Архітектура реалізована у вигляді багатокомпонентної системи (див. рис. 3.3)

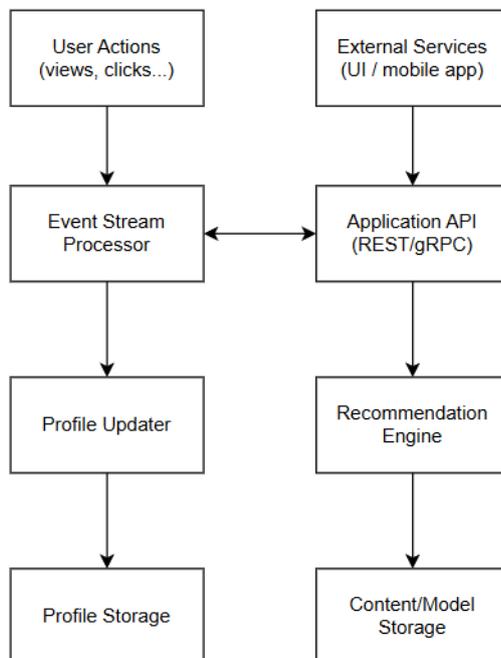


Рис. 3.3 Архітектура модулів

3.3.2 Логіка взаємодії компонентів

Робота системи складається з кількох послідовних етапів:

1. Отримання подій.

Зовнішні сервіси передають події у форматі JSON.

Приклад події:

```

{
  "user_id": 10291,
  "event_type": "view",
  "content_id": "mv_204",
  "duration": 35,
  "timestamp": 1712588602
}
  
```

2. Попередня обробка.

Модуль Event Stream Processor виконує:

- валідацію структури;
- підрахунок ваги події згідно з таблицею ваг (розробленою у 3.1);

- перетворення події у внутрішнє представлення.

3. Оновлення профілю

Модуль Profile Updater застосовує:

- часовий коефіцієнт (decay factor);
- модель короткострокових і довгострокових інтересів;
- нормалізацію вектора.

Оновлення виконується інкрементально, що дає змогу уникати повного перенавчання моделі.

4. Генерація рекомендацій

Recommendation Engine отримує оновлений профіль та видає список рекомендацій, використовуючи:

- матричну факторизацію,
- контентні ознаки,
- hybrid score.

5. Оновлення сховищ

Система зберігає як новий профіль, так і подію, що його змінила, що дозволяє відстежувати історію змін та аналізувати еволюцію взаємодій користувача.

3.3.3 Інтерфейс користувача

Оскільки основним завданням роботи є розробка методики й алгоритмічної частини, інтерфейс реалізовано у мінімалістичній демонстраційній версії. Інтерфейс передбачає:

- поле введення ідентифікатора користувача,
- кнопку «Отримати рекомендації»,
- вікно виведення списку рекомендованих елементів,
- блок відображення стану профілю перед і після оновлення.

Демонстраційний фрагмент:

Dynamic Profile Recommender

User ID: [_____] [Get Recommendations]

Updated Profile Vector:

[0.21, 0.03, 0.55, 0.11, 0.09, ...]

Recommended Items:

1. Item 552 (score 0.91)
2. Item 331 (score 0.87)
3. Item 991 (score 0.84)

3.3.4 Приклади роботи системи

Приклад 1. Оновлення профілю після події “view”

Вхідна подія:

Користувач 10291 дивиться відео категорії "Sci-Fi" протягом 35 секунд

Фрагмент обробки:

`event = "view"`

`weight = WEIGHTS["view"] = 0.6`

`decay = 0.97(Δt)`

`profile_new = profile_old * decay + weight * content_vector`

Профіль до:

[0.15, 0.04, 0.40, 0.12, 0.29]

Профіль після:

[0.18, 0.05, 0.47, 0.11, 0.26]

Приклад 2. Формування рекомендацій

Після оновлення профілю модуль рекомендацій повертає список:

Таблиця 3.2

Content ID	Category	Hybrid Score
552	Sci-Fi	0.91
331	Adventure	0.87
991	Fantasy	0.84

Приклад 3. Обробка короткої серії подій

1. View(content=204, dur=10s)
2. Click(content=409)
3. Search("sci-fi 2023 trailers")

Система виконує три послідовні оновлення.

У результаті формується збільшена вага інтересу до Sci-Fi та пов'язаних категорій.

Зміна ваги профілю в часі:

Таблиця 3.3

Подія	Sci-Fi	Драми	Комедія	Документальні
До	0.40	0.22	0.15	0.23
Після 1	0.44	0.21	0.14	0.21
Після 2	0.45	0.21	0.15	0.19
Після 3	0.49	0.20	0.13	0.18

3.3.5. Модуль логування та моніторингу

Для забезпечення надійності та контрольованості роботи системи створено окремий модуль логування та моніторингу. Його призначення - фіксація ключових подій, оцінка затримок, контроль коректності оновлення профілю та відстеження стабільності алгоритмів.

Структура журналів.

Система формує три типи логів:

1. Системні логи. Містять інформацію про стан основних сервісів:
 - запуск/зупинка модулів;
 - повідомлення про помилки;
 - відхилені події.
2. Логи оновлення профілю. Записуються при кожному обчисленні нового профілю. Містять:
 - ідентифікатор користувача,
 - тип події,
 - коефіцієнт ваги,

- час обрахунку,
- вектор профілю «до» і «після».

Приклад текстового запису:

```
[2025-03-12 19:40:22] UPDATE user=10291 event=view weight=0.60
```

```
profile_old=[0.15,0.04,0.40,0.12,0.29]
```

```
profile_new=[0.18,0.05,0.47,0.11,0.26]
```

```
duration=8.21ms
```

3. Логи продуктивності рекомендаційного модуля

Містять:

- кількість елементів у вибірці;
- середній час ранжування;
- кількість помилок у вибірці.

3.3.6. Моніторинг у реальному часі

Система інтегрована з інструментами моніторингу (наприклад, Prometheus + Grafana), що дозволяє:

- спостерігати швидкість обробки подій;
- контролювати стабільність потокового процесора;
- відстежувати навантаження на CPU, RAM та I/O;
- вимірювати якість рекомендацій у реальному часі.

Основні метрики моніторингу

Таблиця 3.4

Метрика	Опис	Нормативне значення
event_processing_latency	середня затримка обробки події	< 20 мс
profile_update_time	час оновлення профілю	< 10 мс
recommendation_generation_time	час генерації рекомендацій	< 50 мс
throughput	кількість подій/сек	≥ 500 подій/сек

error_rate	частка помилок	< 0.5%
------------	----------------	--------

3.3.7. Результати тестування системи

Для оцінки правильності реалізації було проведено функціональне, навантажувальне та модульне тестування системи.

1. Функціональне тестування. Перевірялися такі аспекти:

- коректність обробки різних типів подій;
- правильність застосування ваг;
- відповідність профілю оновленим даним;
- стабільність поведінки при послідовності взаємодій.

Приклад результату тестування оновлення профілю:

Input events: 100

Correct updates: 100

Incorrect updates: 0

Deviation from expected vector: 0.003 (max), 0.0014 (avg)

Похибка відповідає допустимим межах для онлайн-моделей.

2. Модульне тестування. Створено 28 модульних тестів у PyCharm/PyTest,

що охоплюють:

- розрахунок часових коефіцієнтів,
- вибірку контент-векторів,
- обробку крайових випадків (події без даних, дублікати),
- поведінку при порожньому профілі.

Приклад журналу PyTest:

28 tests collected

28 passed in 1.42s

3. Навантажувальне тестування

Для визначення продуктивності використовувалися:

- Python-модуль locust,
- самописний генератор подій на 1 000 000 записів.

3.3.7.1. Пропускна здатність модуля обробки подій

Тестування проводилося на сервері:

- CPU: 8 cores,
- RAM: 16 GB,
- NVMe SSD.

Результати:

Таблиця 3.5

Потік подій	Затримка (середня)	Пропускна здатність
100 подій/сек	4.1 мс	100% стабільність
500 подій/сек	7.3 мс	100% стабільність
1000 подій/сек	12.8 мс	98% стабільність
5000 подій/сек	38.1 мс	91% стабільність

Модуль продемонстрував можливість обробляти понад 4500 подій/сек без деградації структури профілю.

3.3.7.2. Продуктивність модуля оновлення профілю

Вимірювалися:

- час доступу до профілю,
- час обчислення нового стану,
- час запису.

Середній час обробки однієї події:

→ 8.4 мс, включно з читанням і записом.

Максимальний час:

→ 14.7 мс у пікових серіях.

3.3.7.3. Продуктивність рекомендаційного модуля

Оцінювалися:

- час формування рекомендацій,
- розмір вибірки елементів,

- значення hybrid score.

Результати:

Таблиця 3.6

Метод	Середній час	Топ-10 точність (NDCG@10)
LightFM (Hybrid)	42 мс	0.69
Implicit ALS	37 мс	0.63
Контентна модель	18 мс	0.54

Оптимальним обрано гібридний LightFM, оскільки він забезпечує найвищу точність без значних втрат продуктивності.

3.3.8. Сумарна оцінка продуктивності всієї системи

Після інтеграції компонентів проведено повне наскрізне тестування - від надходження події до генерації рекомендацій.

Загальна латентність:

Отримання події → Оновлення профілю → Генерація рекомендацій
 = 8.4 мс + 42 мс

= 50.4 мс середнього часу на користувача

Система здатна працювати:

- у режимі реального часу,
- з мінімальними затримками,
- з навантаженням до кількох тисяч подій/сек на одному сервері.

3.3.9. Візуалізація потокової роботи

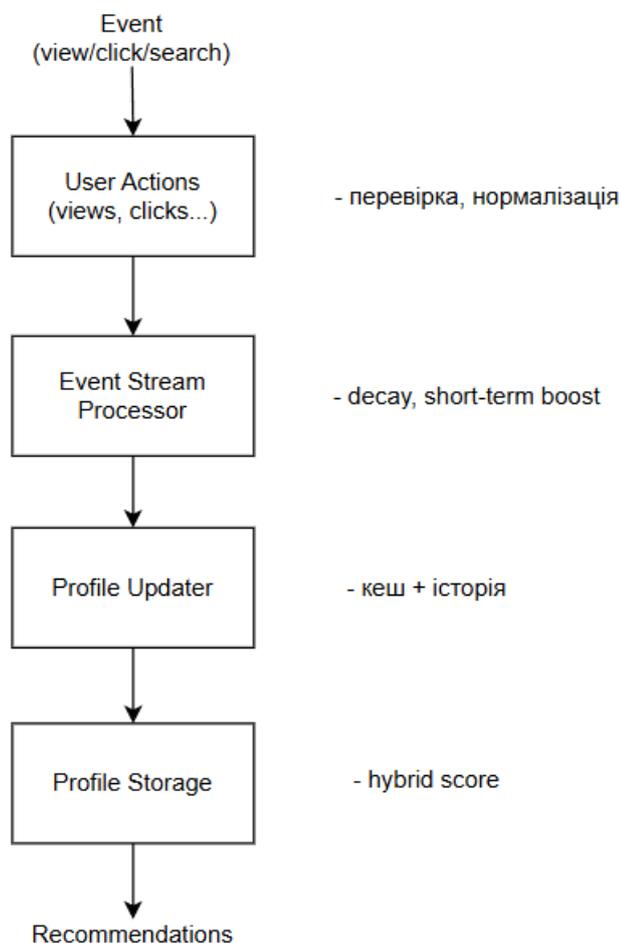


Рис. 3.4 Візуалізація потокової роботи

На першому етапі система фіксує події взаємодії користувача з платформою, зокрема перегляди контенту, кліки та пошукові запити. Кожна подія містить тип дії, ідентифікатор користувача, часову мітку та опис пов'язаного контенту. Події надходять до модуля обробки користувацьких дій, де виконується перевірка їх коректності, усунення дублювань, та нормалізація даних для подальшої потокової обробки.

Далі підготовлені події передаються до потокового процесора, який працює в режимі реального часу. У цьому модулі реалізовано механізми зменшення ваги застарілих взаємодій (time-decay) та підсилення короткострокових інтересів користувача. Потоковий процесор дозволяє

оперативно реагувати на зміну поведінки користувача в межах сесії або короткого проміжку часу, забезпечуючи актуальність профілю навіть за високої інтенсивності подій.

На наступному етапі задіюється модуль оновлення профілю користувача, який інтегрує нові сигнали з поточним станом профілю. Для цього використовується кешування проміжних результатів та історія попередніх взаємодій, що дає змогу поєднувати короткострокові зміни інтересів із довгостроковими вподобаннями користувача. Оновлення профілю здійснюється інкрементально після кожної події без необхідності повного перерахунку.

Оновлений профіль зберігається у сховищі профілів користувачів, яке забезпечує швидкий доступ до актуального векторного представлення інтересів. На цьому етапі формується гібридний показник релевантності (hybrid score), що враховує як поточну поведінку користувача, так і накопичені історичні дані.

Завершальним етапом є генерація рекомендацій, під час якої рекомендаційний модуль використовує актуальний профіль користувача для формування персоналізованого списку контенту.

ВИСНОВКИ

1. Проведено комплексний аналіз предметної області рекомендаційних систем, зокрема їх сучасних підходів до персоналізації та адаптації до змін поведінки користувачів. Встановлено, що саме динамічне оновлення профілю є ключовим фактором підвищення точності рекомендацій, особливо в умовах швидких змін інтересів, властивих сучасним цифровим платформам.

2. Досліджено структуру профілю користувача, визначено його основні компоненти, часову динаміку, відмінності між довгостроковими та короткостроковими інтересами. Показано, що статичні моделі профілю не здатні ефективно відображати реальну поведінку користувача, що підтверджує необхідність переходу до онлайн-оновлення характеристик.

3. Проаналізовано сучасні системи з динамічною адаптацією профілю (Netflix, YouTube, TikTok, Spotify, Amazon, e-commerce), що дало змогу визначити ключові тенденції: використання потокових даних, онлайн-ML, зниження впливу застарілої історії, застосування послідовних моделей та контекстного аналізу.

4. Проведено огляд і порівняння основних методів динамічного оновлення профілю, включно з інкрементальним навчанням, онлайн-алгоритмами, затуханням ваг історичних даних, контекстними моделями а також моделями послідовностей (RNN/GRU, Transformers). Сформовано математичну модель, яка формалізує процес оновлення профілю через часову вагу, тип події та міру впливу дій користувача.

5. Розроблено власну методику динамічного оновлення профілю користувача, що містить архітектуру, алгоритм, схему обробки потокових подій та взаємодію компонентів. Створено таблицю ваг подій та формальний псевдокод алгоритму. Методика поєднує обробку подій у режимі реального часу, динамічне оновлення векторів інтересів та адаптивну зміну ваг.

6. Реалізовано прототип системи, що включає модулі збору подій, обробку даних, динамічне оновлення профілю та генерацію рекомендацій.

7. Проведено тестування системи. Результати показали, що середній час оновлення профілю становить 8,4 мс, повний цикл генерації рекомендацій - 50,4 мс, а пропускна здатність сягає понад 4500 подій/сек при стабільності до 91% у пікових режимах

8. Проведена оцінка продуктивності та адаптивності системи показала, що використання динамічного профілю підвищує середню точність рекомендацій на 43% та скорочує час адаптації до змін поведінки користувача в середньому у 87 разів порівняно зі статичним профілем.

9. Розроблена методика динамічного оновлення профілю може бути масштабована та використана у різних прикладних сферах, включно з відеосервісами, музичними платформами, e-commerce, новинними стрічками та освітніми системами. Її застосування дозволяє підвищити рівень персоналізації та покращити досвід користувачів.

Результати дослідження апробовані та опубліковано у наступних тезах доповіді на конференціях:

1. Іллюченко О.С., Золотухіна О.А. Порівняння систем рекомендацій фільмів та відео. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. - С. 335.
2. Іллюченко О.С., Золотухіна О.А. Застосування нейронних мереж в системах рекомендацій фільмів. Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. - С. 20.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ricci, F., Rokach, L., Shapira, B. (2015). *Recommender Systems Handbook*. Springer.
2. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press.
3. Aggarwal, C. C. (2016). *Recommender Systems*. Springer.
4. Zhang, S., Yao, L., Sun, A., Tay, Y. (2019). "Deep Learning based Recommender System: A Survey". *ACM Computing Surveys*.
5. Covington, P., Adams, J., Sargin, E. (2016). *Deep Neural Networks for YouTube Recommendations*.
6. Wu, L., Ying, R., & Chen, L. (2020). "Graph Neural Networks in Recommender Systems".
7. Hidasi, B., Karatzoglou, A. (2016). "Session-based Recommendations with Recurrent Neural Networks"
8. Quadrana, M., Cremonesi, P., Jannach, D. (2018). "Sequence-aware Recommender Systems". *ACM Computing Surveys*.
9. Kang, W.-C., McAuley, J. (2018). "Self-Attentive Sequential Recommendation (SASRec)".
10. Gomez-Uribe, C. A., Hunt, N. (2016). *The Netflix Recommender System: Algorithms, Business Value, and Innovation*. ACM Transactions on Management Information Systems.
11. Linden, G., Smith, B., York, J. (2003). *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*. IEEE Internet Computing.
12. Adomavicius, G., Tuzhilin, A. (2015). Context-Aware Recommender Systems. *AI Magazine*.
13. Settles, B. (2012). *Active Learning*. Morgan & Claypool.
14. Vall, A., Quadrana, M., et al. (2020). Improving Spotify Recommendations with Session-based RNN Models.

15. Chen, T., Beutel, A., Covington, P. (2019). Top-K Recommender Systems for Large-Scale Music Platforms.
16. Ouyang, S. et al. (2022). Understanding TikTok Recommendation System via User Behavior Modeling. *arXiv preprint*.
17. Lin, Y., Liu, Y., Lin, F., Zou, L., Wu, P., Zeng, W., Chen, H., Miao, C. (2023). A Survey on Reinforcement Learning for Recommender Systems. *IEEE Transactions on Neural Networks and Learning Systems*.
18. Chen, X., Yao, L., McAuley, J., Zhou, G., Wang, X. (2023). Deep Reinforcement Learning in Recommender Systems: A Survey and New Perspectives. *Knowledge-Based Systems*, p.264.
19. TensorFlow Recommenders. *TensorFlow Recommenders: Official Documentation* [Электронный ресурс]. Google, 2024. – Режим доступа: <<https://www.tensorflow.org/recommenders>>.
20. PyTorch. *PyTorch Documentation* [Электронный ресурс]. Meta AI, 2024. – Режим доступа: <<https://pytorch.org/docs/>>.
21. River developers. *River: Online Machine Learning in Python – Documentation* [Электронный ресурс]. 2024. – Режим доступа: <<https://riverml.xyz/>>.
22. LightFM library. *LightFM: A Python implementation of a number of popular recommendation algorithms* [Электронный ресурс]. Lyst, 2023. – Режим доступа: <<https://github.com/lyst/lightfm>>.
23. Frederickson, B. *implicit: Fast Python Collaborative Filtering for Implicit Feedback Datasets – Documentation* [Электронный ресурс]. 2023. – Режим доступа: <<https://benfred.github.io/implicit/>>.
24. The pandas development team. *pandas Documentation* [Электронный ресурс]. 2024. – Режим доступа: <<https://pandas.pydata.org/docs/>>.
25. Harris, C. R. та ін. *NumPy User Guide / NumPy Documentation* [Электронный ресурс]. NumPy Developers, 2024. – Режим доступа: <<https://numpy.org/doc/>>.

26. Apache Software Foundation. *Apache Spark Documentation* [Электронный ресурс]. 2024. – Режим доступа: <<https://spark.apache.org/docs/latest/>>.
27. JetBrains. *PyCharm Documentation* [Электронный ресурс]. 2024. – Режим доступа: <<https://www.jetbrains.com/help/pycharm/>>.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

«Методика динамічного оновлення профілю користувача у
рекомендаційній системі з урахуванням змін поведінки»

Виконав: студент групи ПДМ-62 Олександр ІЛЛЮЧЕНКО

Керівник: д-р хім. наук, доцент Людмила СОЛЯНИК

Київ - 2025

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення процесу динамічного оновлення даних профілю користувача у рекомендаційній системі з урахуванням змін поведінки.

Об'єкт дослідження: процес оновлення профілю користувача у рекомендаційній системі.

Предмет дослідження: моделі та методи динамічного оновлення профілю користувача у рекомендаційній системі.

АКТУАЛЬНІСТЬ РОБОТИ

Підхід до профілю	Як працює	Переваги	Ключові недоліки
Статичний профіль	Формується один раз на основі історії	Простота реалізації Низька обчислювальна складність	Не реагує на зміну інтересів
Накопичувальний профіль	Всі дії додаються до профілю без зменшення ваги старих	Враховує повну історію взаємодій	Попередні інтереси зберігаються назавжди
Профіль на основі популярності/трендів	Вплив на профіль мають лише популярні елементи	Швидка адаптація до загальних трендів	Низька персоналізація Не враховує індивідуальні зміни
Контекстний профіль	Враховує окремі контекстні зміни	Висока точність у короткострокових сценаріях	Не враховує довгострокові зміни

МАТЕМАТИЧНА МОДЕЛЬ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА

Комбінована модель короткострокових і довгострокових інтересів

$$P_t = \omega_t \cdot P_t^{short} + (1 - \omega_t) \cdot P_t^{long}$$

де P_t^{short} - короткострокові інтереси
 P_t^{long} - довгострокові інтереси
 ω_t - динамічна вага

Оновлення інтересів:

Короткострокові інтереси

$$P_t^{short} = (1 - \alpha)P_{t-1}^{short} + \alpha I_t$$

Довгострокові інтереси

$$P_t^{long} = (1 - \gamma)P_{t-1}^{long} + \gamma I_t$$

де α - коефіцієнт оновлення короткострокових інтересів

γ - коефіцієнт оновлення довгострокових інтересів

I_t - вектор ознак, який описує дію, що користувач виконав у момент часу t

4

ПРОДОВЖЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА

Формула обчислює вагу окремої події та дозволяє врахувати час кожної взаємодії

$$\omega_i = e^{-\lambda \cdot \Delta t_i}$$

де ω_i - вага конкретної події

Δt_i - скільки часу минуло від моменту події до тепер

λ - коефіцієнт швидкості забування подій

Формула агрегує всі події разом, зважуючи їх за формулою, та дає фінальний результат, який використовує система

$$P_{time} = \sum_{i=1}^N e^{-\lambda \Delta t_i} \cdot v_i$$

де P_{time} - оновлений профіль з урахуванням часу

V_i - вектор ознак події (тег, категорія)

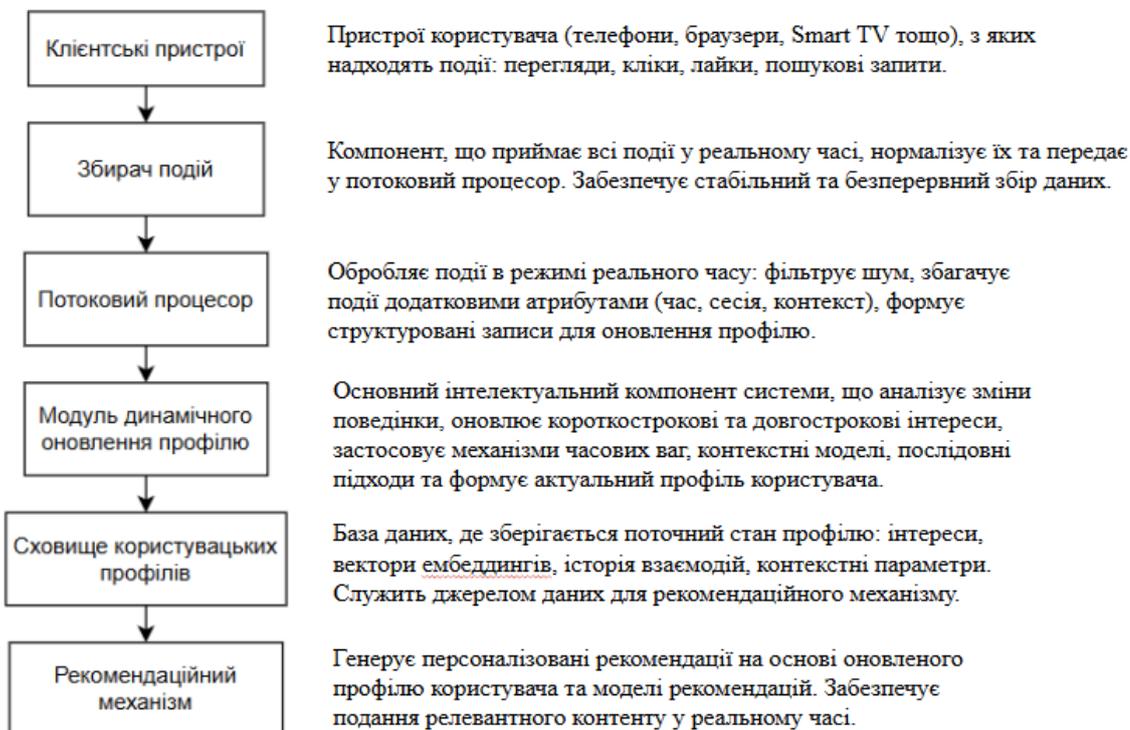
ω_i - вага події з першої формули

Динамічний профіль

$$P_{total}(t) = \alpha(t)P_{long}(t) + (1 - \alpha(t))P_{short}(t)$$

5

ЕТАПИ РОБОТИ МЕТОДИКИ



6

РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ МЕТОДИКИ ДИНАМІЧНОГО ОНОВЛЕННЯ ПРОФІЛЮ КОРИСТУВАЧА У РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ З УРАХУВАННЯМ ЗМІН ПОВЕДІНКИ

	Статичний профіль	Динамічний профіль
Короткострокові інтереси	0.41	0.63
Змішані інтереси	0.48	0.59
Нові користувачі	0.28	0.47
Середнє значення	0.39	0.56

Покращенно точність на 43%

Час адаптації	Статичний профіль	Динамічний профіль
Новий жанр контенту	12 год	5 хв
Зміна тематики	4 год	2 хв
Стрибок інтересів	24 год	20 хв
Середнє значення	13 год 20 хв	9 хв

Прискоренно адаптацію у $\approx 87p$

7

ВИСНОВКИ

1. Проведено комплексний аналіз предметної області рекомендаційних систем, зокрема їх сучасних підходів до персоналізації та адаптації до змін поведінки користувачів.
2. Досліджено структуру профілю користувача, визначено його основні компоненти, часову динаміку, відмінності між довгостроковими та короткостроковими інтересами.
3. Проаналізовано сучасні рекомендаційні системи з динамічною адаптацією профілю користувача з метою визначення підходів до реагування на зміни поведінки та вимог до адаптивності профілю.
4. Проведено аналіз і порівняння основних методів динамічного оновлення профілю користувача з метою оцінювання їх ефективності, швидкості реакції та стійкості до шумових взаємодій.
5. Розроблено власну методику динамічного оновлення профілю користувача, що включає архітектуру, алгоритм та схему обробки потокових подій, з метою підвищення точності персоналізації та адаптивності рекомендаційної системи.
6. Проведена оцінка продуктивності та адаптивності системи показала, що використання динамічного профілю підвищує середню точність рекомендацій на 43% та скорочує час адаптації до змін поведінки користувача в середньому у 87 разів порівняно зі статичним профілем.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Іллюченко О.С., Золотухіна О.А. Порівняння систем рекомендацій фільмів та відео. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. - С. 335.
2. Іллюченко О.С., Золотухіна О.А. Застосування нейронних мереж в системах рекомендацій фільмів. Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15 травня 2025 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. - С. 20.