DOI https://doi.org/10.32689/maup.it.2026.1.2
УДК 004.78:004.5

ДИНАМІЧНА АДАПТАЦІЯ ПРОФІЛЮ КОРИСТУВАЧА В РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ НА ОСНОВІ АНАЛІЗУ ІНФОРМАЦІЇ ПРО ЙОГО ПОВЕДІНКУ

О. А. Золотухіна, О. Г. Чолишкіна, О. С. Іллюченко, Т. О. Лисенко

Dynamic adaptation of the user profile in a recommendation system based on analysis of information about his behavior

Oksana Zolotukhina, Olga Cholishkina, Oleksandr Illyuchenko, Taras Lysenko

Анотація

Зростання обсягів цифрового контенту та ускладнення поведінкових сценаріїв користувачів підвищують вимоги до точності й адаптивності сучасних рекомендаційних систем. У цих умовах особливого значення набувають методи динамічного оновлення профілю користувача, здатні забезпечити актуальність персоналізації в умовах змін інтересів, впливу контексту та нерівномірного характеру взаємодії з інформаційними сервісами. Проблема полягає у необхідності забезпечення адаптації таких систем до змін інтересів користувача в умовах динамічного середовища, де короткострокові вподобання швидко змінюються, а довгострокові зберігають інерційність, що ускладнює підтримання високої точності персоналізації.

Метою дослідження є розробка математичної моделі та архітектурного рішення для динамічної адаптації профілю користувача, які дозволяють ефективно поєднувати аналіз довгострокових та короткострокових інтересів з урахуванням контекстуальних факторів для підвищення точності персоналізації в рекомендаційних системах. Робота спрямована на подолання таких недоліків існуючих систем, як висока чутливість до випадкового «шуму» в діях користувачів, складність інтерпретації оновлень та нездатність своєчасно реагувати на згладжування змін у поведінкових моделях.

В роботі використовується механізм динамічного керування балансом між короткостроковими (STI) та довгостроковими (LTI) інтересами за допомогою адаптивного коефіцієнта, що дозволяє системі автоматично перемикатися на актуальні потреби користувача при різкій зміні поведінки або спиратися на стійкі звички при стабільній взаємодії.

Розроблено метод інтеграції контекстних чинників у процес оновлення профілю користувача, що забезпечує можливість підсилювати або послаблювати значущість подій залежно від зовнішніх умов (час, пристрій, місцеперебування), перетворюючи профіль на контекстно-чутливу структуру. Технологічний підхід базується на архітектурі конвеєрної обробки даних для потокового аналізу подій у режимі реального часу. Експериментальна валідація запропонованих рішень проведена шляхом тестування на датасеті MovieLens із порівняльним аналізом точності рекомендацій динамічних моделей відносно статичних. Характерною відмінністю запропонованого підходу є інтеграція контекстуальних параметрів у процес оновлення профілю, що перетворює його на контекстно-чутливу структуру, здатну адаптуватися до умов взаємодії.

Технічні випробування довели спроможність рекомендаційної системи стабільно працювати в умовах інтенсивного потоку даних, забезпечуючи швидке оновлення знань системи про людину без втрати загальної продуктивності. Практичне застосування можливе в рекомендаційних системах, що функціонують у режимі реального часу та працюють з інтенсивними потоками даних, зокрема в медіасервісах, електронній комерції та інформаційних платформах, за умови наявності засобів збору подій користувача та інфраструктури потокової обробки.

Ключові слова: *профіль користувача, рекомендаційна система, поведінкові патерни, короткострокові інтереси, довгострокові інтереси, динамічна адаптація, потокова обробка даних<i>.<\i>*

Abstract

The rapid growth of digital content volumes and the increasing complexity of user behavioral patterns have intensified the requirements for the accuracy and adaptability of modern recommender systems. Under these conditions, methods for dynamic user profile updating become particularly important, as they enable the maintenance of relevant personalization in the presence of changing interests, contextual influences, and the irregular nature of user interaction with information services. The problem lies in the need to ensure the adaptation of such systems to changes in user interests in a dynamic environment, where short-term preferences change rapidly, and long-term ones retain inertia, which makes it difficult to maintain high personalization accuracy.

The aim of the research is to develop a mathematical model and architectural solution for dynamic user profile adaptation, which allow to effectively combine the analysis of long-term and short-term interests taking into account contextual factors to increase the accuracy of personalization in recommender systems. The work is aimed at overcoming such shortcomings of existing systems as high sensitivity to random «noise» in user actions, complexity of interpreting updates and inability to respond in a timely manner to smooth changes in behavioral patterns.

The work introduces a mechanism for dynamically managing the balance between short-term (STI) and long-term (LTI) interests using an adaptive coefficient, which allows the system to automatically switch to the current needs of the user in the event of a sharp change in behavior or rely on stable habits during stable interaction.

© Zolotukhina O., Cholishkina O., Illyuchenko O., Lysenko T., 2026
Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0

A method for integrating contextual factors into the user profile update process has been developed, which provides the ability to enhance or weaken the significance of events depending on external conditions (time, device, location), transforming the profile into a context-sensitive structure. The technological approach is based on the architecture of pipeline data processing for streaming analysis of events in real time. Experimental validation of the proposed solutions was carried out by testing on the MovieLens dataset with a comparative analysis of the accuracy of recommendations of dynamic models relative to static ones. A characteristic difference of the proposed approach is the integration of contextual parameters into the profile update process, which turns it into a context-sensitive structure capable of adapting to interaction conditions.

Technical tests have proven the ability of the recommendation system to work stably in conditions of intensive data flow, ensuring rapid updating of the system's knowledge about a person without loss of overall performance. Practical application is possible in recommendation systems that operate in real time and work with intensive data flows, in particular in media services, e-commerce and information platforms, provided that there are means of collecting user events and a stream processing infrastructure.

Key words: *user profile, recommender system, behavioral patterns, short-term interests, long-term interests, dynamic adaptation, streaming data processing.*

1. Introduction. Over the past decade, recommender systems have ceased to be static mechanisms that respond only to a set of previous actions. At the heart of their development are models that are able to adapt to the dynamics of user behavior, take into account long-term and short-term interests, and also respond to changes in the context. The user profile is the central element of any recommender system, since it is it that determines the nature of personalization and the degree of correspondence of recommendations to individual interests. Regardless of the complexity of the algorithms or the architecture of the system, the effectiveness of prediction largely depends on how completely and correctly the user model is reflected. In modern information services, the profile takes the form of a multidimensional structure that combines both historical data and contextual signals, deep latent features and behavioral patterns [1, 5, 13]. In the most general sense, a user profile is a formalized representation of data that characterizes the preferences, interests, behavioral style, and individual patterns of user interaction with the platform. The profile acts as an intermediary between real human actions and the algorithmic logic of the system: it provides a transition from raw events (clicks, views, ratings) to structured representations suitable for analysis and prediction [13]. An important modern trend is contextual personalization, when recommendations depend on the current conditions of service use. For example, the algorithm takes into account that the user views content on a smartphone in the evening, when he has a limited attention span and prefers a short format [7, 11]. It is worth noting the dynamic formation of the user profile, which is critically important in environments with high changes in interests. Systems are moving from static models to adaptive ones, in which information about the user is constantly refined, updated, and weighted depending on the age and significance of signals [3]. Taken together, these trends demonstrate that recommender systems are no longer just tools for finding relevant content. They are transforming into complex adaptive behavioral analysis models capable of predicting changes in interests and actively adapting to user dynamics, which determines the relevance of research in the field of user profile updating.

In the scientific literature, the most common division of recommendation algorithms into three large groups is: content-oriented, collaborative, and hybrid systems [13]. Content-oriented models generate recommendations based on the characteristics of objects and user preferences. They compare new elements with those that the user previously preferred. This approach works well in areas where objects have pronounced properties, for example, in music services or book catalogs. At the same time, it is prone to the problem of excessive uniformity of recommendations due to "closure" within the framework of already known interests [5]. Collaborative filtering is based not on the properties of the content, but on the behavior of the community. The system analyzes the similarity between users or between objects, builds a matrix of interactions, and predicts which elements may be relevant. Its advantage is the ability to open new categories of content to the user. However, classical algorithms of this type depend on a sufficient amount of data, therefore they face the problems of "cold start" and matrix sparsity [1, 5]. Hybrid models combine both approaches, compensating for their shortcomings. Modern systems most often implement hybrid schemes, also integrating elements of machine learning, matrix factorization, graph methods, contextual modeling and deep neural networks [13, 15]. In any recommendation system, the profile is an information model of the user, which is constantly being refined and accumulates new interaction experience [1, 5]. The main functions of the profile include:

– preserving the current state of interests;
– generalizing behavior in the form of features;
– providing input data for recommendation algorithms;
– adapting the system to changes in user behavior;
– reconciling long-term and short-term interests.

The user profile is a key element in the functioning of the recommender system at all stages of its work. At the first stage (filtering and selection of candidates), the system determines which objects may be of interest to the user. To do this, the long-term interests accumulated in the profile are analyzed, as well as their correlation with current queries [1, 5]. The ranking stage involves determining the importance of each candidate. The algorithm assesses the degree of correspondence of the object to the user's profile, taking into account the relationships between features, latent vectors and context [15, 12]. In the future, recommender systems with real-time learning constantly refine the user model: each view, rejection or interaction changes the weights of the profile parameters. It is the ability to do such updating that allows the system to adapt to rapid changes in preferences [3]. The profile helps to maintain stable preferences formed over a long period, but also takes into account fleeting intentions. For example, the user may be temporarily interested in a certain topic, and the system must respond to this without permanently changing the global profile [15].

The effectiveness of profiling directly depends on the quality of data, its completeness and relevance. Common problems include the sparseness of interactions among new users; rapid changes in interests, which makes static models inaccurate; noise in the data caused by random or forced actions; overloading the profile with old events that no longer correspond to current interests; the difficulty of taking into account the context and dependencies between events. These difficulties have become a prerequisite for the development of adaptive methods that can not only accumulate information, but also take into account the dynamics of its changes, which is key in modern generation systems [1, 3]. In most modern services, the user profile is transformed from a simple set of data into a complex dynamic object that characterizes behavior in a wide time and contextual range. It is it that determines the quality of personalization, the ability of the system to adapt, reduce information redundancy and maintain a high level of user interaction with the platform. Thus, the profile becomes not just a structured set of characteristics, but a key link between the person and the algorithm - a central part of the recommendation generation mechanism [13, 15].

Modeling user behavior is one of the key areas of development of modern recommendation systems. Unlike classical approaches that relied on static profiles and averaged ratings, modern models take into account the dynamic nature of human interaction with the platform. User behavior changes both in short periods of time (during the day) and in long periods (months, years), which creates a need for methods that can reproduce and predict these changes. Behavioral patterns can be stable, random, or cyclical, and it is their reproduction that allows us to build accurate predictions about future actions [8]. User actions are rarely random. Often, the choice of the next element depends on the previous one, and it is precisely such dependencies that neural networks that work well with time series use [2, 4, 6, 7, 8, 11]. Many users have stable habits that manifest themselves regularly: viewing at a certain time of day, prioritizing certain genres/categories, using the service on weekends, etc. Identifying such patterns allows the system to form recommendations that are consistent with the user's habits [1, 6, 7]. Behavior is influenced by external factors: season, location, current events, educational or work schedule. Contextual behavior tends to change, so it is important to consider it when updating the profile [1]. In many cases, the user does not act in accordance with long-term interests, but under the influence of the moment, for example, searching for information for work or education creates temporary changes in behavior [6, 7]. Modern recommendation platforms are increasingly moving from static user models to adaptive mechanisms that take into account changes in behavior in the short and long term. Dynamic profile updating is becoming a key element in increasing the accuracy of personalization, as it allows the system to quickly respond to changes in interests, the effects of contextual factors, and the user's interaction with the platform [1, 5-9, 11, 12, 14].

Despite active development, most systems have a number of common shortcomings related specifically to dynamic profile updates:

– insufficient separation of short-term and long-term interests, which leads to the fact that even complex models often do not distinguish temporary action from a fundamental change in preferences;

– high sensitivity to noise effects, due to which individual random interactions can disproportionately affect the profile;

– insufficient consideration of context, which leads to the interpretation of all user actions as equally significant, although they may have different natures;

– weak interpretability of profile updates, as a result of which the user does not always understand how the recommendation is formed, and the developers - what exactly changed the model;

– problems adapting to behavior that changes smoothly rather than.

The aim of the research is to develop a mathematical model and architectural solution for dynamic user profile adaptation, which allow to effectively combine the analysis of long-term and short-term interests taking into account contextual factors to increase the accuracy of personalization in recommender systems. The work is aimed at overcoming such shortcomings of existing systems as high sensitivity to random «noise» in user actions, complexity of interpreting updates and inability to respond in a timely manner to smooth changes in behavioral patterns.

2. Materials and methods. The object of the study is the process of forming and updating a user profile in recommender systems. In modern recommender systems, a user profile is considered as a formal structure that reflects his interests, behavioral patterns, current and long-term preferences. Unlike static models, dynamic profiles change over time as the user's behavior is updated, the context of his interaction changes, and new signals appear. In the most general form, a user profile can be represented as a vector in a multidimensional feature space:

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

where

– P_t – user profile at a point in time t;

– $p_{t,i}$ – assessment of interest in characteristic i at time point t;

– n – total number of characteristics (topics, genres, functional categories, activity types, etc.).

Every change in user behavior generates a new event I_t, which is also represented as a vector:

$$I_t = (i_{t,1}, i_{t,2}, \dots, i_{t,n}). \tag{2}$$

The task of the mathematical model is to determine the method of transition from the old profile P_{t-1} to new profile P_t taking into account both short-term and long-term changes. Therefore, a dynamic profile is interpreted as a system that evolves according to the rules:

$$P_t = F(P_{t-1}, I_t, \Theta_t), \tag{3}$$

where

– $F(\cdot)$ – update function,

– Θ_t – adaptation parameters (weights, coefficients, learning rate).

Short-term and long-term interests of a user have different nature and different impact: short-term interests change quickly and reflect the current context (search topics, trends, temporary needs); long-term interests evolve more slowly and correspond to stable preferences. Therefore, it is advisable to present the profile as a combination of two sub-profiles:

$$P_t = w_t \cdot P_t^{short} + (1 - w_t) \cdot P_t^{long}, \tag{4}$$

where

– P_t^{short} – short-term user profile,

– P_t^{long} – long-term user profile,

– w_t – short-term profile weight coefficient, which changes dynamically.

The weight w_t depends on how stable the user's behavior is over a certain period of time:

$$w_t = \sigma(\beta \cdot S_t), \tag{5}$$

where

– S_t – behavioral change indicator (e.g., rate of change in content categories or diversification of actions),

– β – sensitivity parameter,

– $\sigma(\cdot)$ – sigmoid function.

The intuitive assumption used in this work is the following: if the user changes interests abruptly, then S_t is large and $w_t \to 1$, in which case the system relies on the short-term profile; if the behavior is stable, then $w_t \to 0$, i.e. the long-term profile dominates.

The short-term profile models instantaneous user actions and is updated using an exponential decay scheme.:

$$P_t^{short} = (1 - \alpha) P_{t-1}^{short} + \alpha I_t. \tag{6}$$

Parameter $\alpha \in (0; 1)$ regulates the speed of «forgetting»: if $\alpha \to 1$, then the model responds quickly to new actions, with $\alpha \to 0$ the short-term profile is smoothed out. This mechanism well describes situations such as the following: short-term interest in a certain topic; intense changes in behavior; behavioral fluctuations.

The long-term profile changes more slowly:

$$P_t^{long} = (1 - \gamma) P_{t-1}^{long} + \gamma I_t, \tag{7}$$

where $\gamma \ll \alpha$. Typical values of coefficients obtained experimentally are $\alpha = 0.2 \div 0.5$, $\gamma = 0.01 \div 0.05$. This allows the system to not react too quickly to random actions, to capture stable interests, and to distinguish between short bursts of activity and real changes.

Substituting the formulas for short-term and long-term update into expression (4), we obtain:

$$P_t = w_t\left[\left(1-\alpha\right)P_{t-1}^{short} + \alpha I_t\right] + \left(1-w_t\right)\left[\left(1-\gamma\right)P_{t-1}^{long} + \gamma I_t\right]. \tag{8}$$

One of the key requirements for a dynamic profile update model is the ability not only to add new data, but also to correctly redistribute its influence relative to older information. This means that the system must mathematically determine which part of the historical interactions remains relevant, and which signals should be considered obsolete. For this, time weights are used. Let there be a sequence of user interactions with the system:

$$I = \{(v_1, t_1), (v_2, t_2), \ldots, (v_n, t_n)\}, \tag{9}$$

where

- v_i – vector of interaction features (e.g. genre of movie watched, video category, viewing duration),
- t_i – interaction time.

Each event is assigned a weighting factor:

$$w_i = e^{-\lambda\left(T-t_i\right)}, \tag{10}$$

where

- T – current time,
- λ – decay parameter, which determines the rate at which data loses relevance.

Thus, the model reduces to the problem of calculating a weighted set of interactions:

$$P\left(t\right) = \frac{\sum_{i=1}^{n} w_i v_i}{\sum_{i=1}^{n} w_i}. \tag{11}$$

This formula provides two important properties: current behavior dominates over historical behavior (recent interactions automatically receive greater weight), and the user profile is updated smoothly (gradual reduction of the influence of old events prevents abrupt changes in recommendations when new signals appear). For recommender systems with frequent interactions (YouTube, TikTok, Spotify), the value of the parameter λ can be relatively large, as user behavior changes quickly. For systems with slow dynamics (educational platforms, professional services) – smaller, to maintain profile stability.

3. Results. User behavior in a recommendation system is not static – it depends on external conditions that determine the context of interaction. Research in the field of personalized content shows that parameters such as time of day, day of the week, access device, location or current emotional state significantly change the nature of interaction with content. Therefore, the mathematical model of profile updating should take into account the context not as an addition, but as a separate structural component that affects the result of the calculation. In general, the context is described as a vector:

$$C(t) = [c_1(t), c_2(t), \ldots, c_m(t)], \tag{12}$$

where

- $c_j(t)$ – the value of a specific context parameter at a point in time t;
- m – number of factors taken into account by the system.

Let's define typical contextual parameters with the following list of 7 components: time of day – $c_1(t)$; day of the week – $c_2(t)$; device type (mobile/PC) – $c_3(t)$; user activity (video consumption, search, viewing recommendations) – $c_4(t)$; session duration – $c_5(t)$; place of residence – $c_6(t)$; intensity of interactions – $c_7(t)$. To incorporate context into the profile update model, a contextual influence function is introduced:

$$\phi(v_i, C(t_i)) = v_i \odot g(C(t_i)), \tag{13}$$

where

- $g(C(t_i))$ – function of modifying interaction features depending on the context,
- $\odot$ – element-wise multiplication operation.

This allows to strengthen or weaken the impact of a particular interaction depending on the conditions of its implementation. For example, viewing entertainment content in the evening may have more weight than during working hours; searching for educational materials on weekdays has different semantics than on weekends. Then the final formula for determining the user profile component taking into account the context takes the form:

$$P(t)=\frac{\sum_{i=1}^{n} w_i \phi\left(v_i, C(t_i)\right)}{\sum_{i=1}^{n} w_i}. \tag{14}$$

Thus, the user profile becomes not just a reflection of his interactions, but a context-sensitive structure capable of modeling real changes in behavior.

To take into account the processes of dynamic updating of the user profile, an approach is proposed based on the concept of continuous adaptation of the recommendation model to changes in user behavior. It combines the capabilities of streaming event processing, online updating of profile parameters and distributed computing architecture, which ensures scalability and low system response time. The system architecture consists of several components operating in the pipeline data processing mode (Fig. 1).

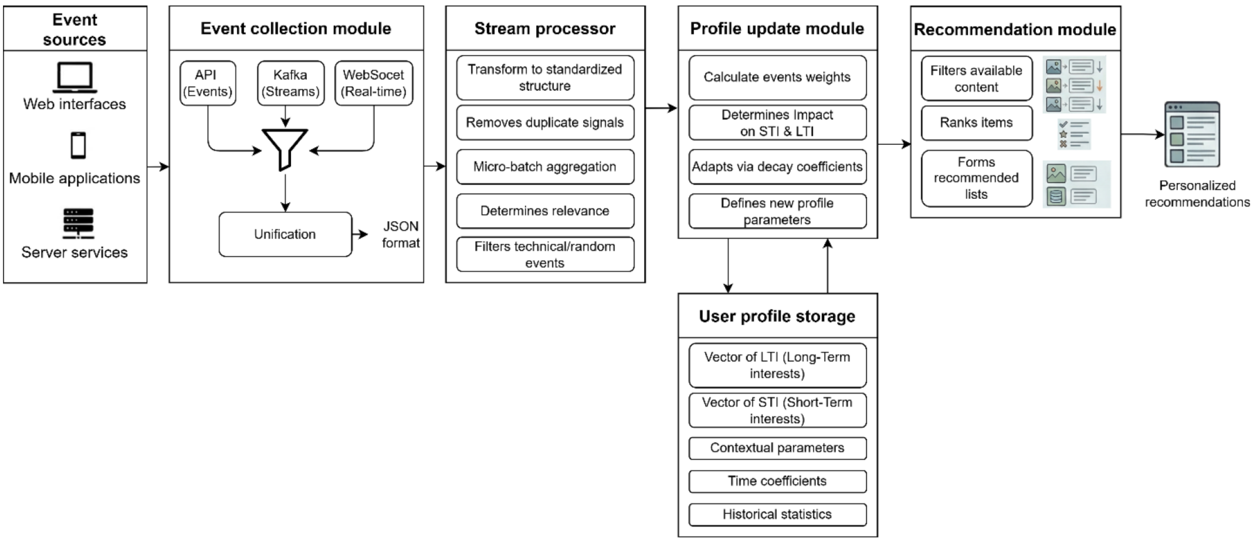

Fig. 1. Data processing pipeline architecture

The event collection module receives events from client applications (web interface, mobile applications or server services). Event collection can be implemented via HTTP-API, WebSocket connections, message queues (Kafka, RabbitMQ), server loggers. Each event has a unified form: {*user_id, item_id, event_type, timestamp, session_id, context*}.

The stream processor performs primary processing of input data: transformation of events into standardized structures; removal of duplicate signals; aggregation in microbatch; determination of event relevance; filtering of technical and random events.

The user profile repository contains a set of parameters describing user preferences: long-term interest vector (LTI); short-term interest vector (STI); contextual parameters; time coefficients; historical statistics. Storage can be organized in the form of Redis / MongoDB (for fast updates), Cassandra / Bigtable (for distributed data).

The profile update module uses the mathematical model proposed in the article. Its tasks include: calculating the weight of a new event; determining how the event affects STI and LTI; adapting the profile according to the attenuation coefficients; determining new values of the profile parameters.

The logging and monitoring module tracks the intensity of events, processing delays, accuracy of recommendations and stability of algorithms. The system generates three types of logs:

1. System logs. Contains information about the status of the main services: starting/stopping modules; error messages; rejected events.

2. Profile update logs. Recorded each time a new profile is calculated. Contains user ID, event type, weight factor, calculation time, “before” and “after” profile vector.

Example of a profile update log entry:

```
[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. Recommendation module performance logs – contain the number of items in the sample; average ranking time; number of errors in the sample.

The recommendation module, after each profile update, filters available content, ranks items, and generates recommended lists.

The training data for the experimental study was generated based on the open MovieLens dataset [10], which contains user interactions with objects (movies) in the form of tuples of the form ⟨user_id, item_id, rating, timestamp⟩. Based on this data, an event flow of interactions was generated, which is used to build and dynamically update the user profile. Additionally, explicit ratings were transformed into implicit behavioral signals and session sequences of interactions were generated, which allows modeling short-term and long-term user interests. An example of modeling the recommendation formation process is given below.

Example 1. Updating a profile after a “view” event and generating a recommendation.

```
<i>Incoming event: user 10291 watches a video in the «Sci-Fi» category for 35 seconds.
Profile before: [0.15, 0.04, 0.40, 0.12, 0.29]
Profile after: [0.18, 0.05, 0.47, 0.11, 0.26]
```

Recommendation formation (Table 1).

Table 1

Example of genre recommendation after profile update

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

Example 2. Processing a short series of events.

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

To process these events, the system performs three consecutive updates. As a result, an increased weight of interest in Sci-Fi and related categories is formed. The dynamics of the change in profile weights over time is shown in Table 2.

Table 2

Dynamics of profile weight changes over time during the processing of a series of events

State	Sci-Fi	Drama	Comedy	Documentary
Initial profile	0.40	0.22	0.15	0.23
Event processing 1	0.44	0.21	0.14	0.21
Event processing 2	0.45	0.21	0.15	0.19
Event processing 3	0.49	0.20	0.13	0.18

The final test results showed a stable increase in recommendation accuracy compared to using a static profile, regardless of the type of user interests (Table 3).

Table 3

Recommendation accuracy indicators depending on profile type

User Interest Type	Static Profile	Dynamic Profile
Short-term interests	0.41	0.63
Mixed interests	0.48	0.59
New users	0.28	0.47

Testing the throughput of the event processing module, the data of which is used to update the profile, has demonstrated the ability to process over 4500 events/s without degrading the profile structure. The average processing time of one event is on average 8.4 ms, including reading and writing, the maximum time is on average 14.7 ms in peak series. The average time of the recommendation formation process after receiving the event and updating the profile is on average 50.4 ms.

Although the results demonstrated effectiveness on the MovieLens dataset (film industry), the dynamics of interest changes in media services may differ significantly from other areas, such as educational platforms or professional services, where user behavior is more stable.

Discussion. The effectiveness of the proposed model, as demonstrated in the experiments on the MovieLens dataset, stems primarily from the balancing mechanism between STI and LTI. The increase in accuracy for new users from 0.28 to 0.47 confirms that the adaptive coefficient w_t enables the system to rapidly form a profile based on initial signals, effectively mitigating the “cold start” problem. Unlike static methods that accumulate data over long periods, the developed approach provides flexibility: upon a sharp change in the interaction vector, the S_t indicator increases, prompting the system to prioritize short-term interests.

The specific integration of context via the $\phi(v_i, C(t_i))$ function allows for the filtering of behavioral “noise”. This explains the high accuracy score (0.63) for short-term interests: the system does not merely record content consumption but weighs its significance relative to time, device, and location. Such an approach renders the profile “sensitive” to situational user needs, which is critical for services with high interaction intensity. Technical metrics, specifically a throughput exceeding 4500 events/s, demonstrate that the mathematical complexity added by exponential decay does not impose a critical load on the infrastructure. This allows the solution to scale in high-load environments, provided that message queues and in-memory storage like Redis are utilized. However, the prevalence of media content in the test sample leaves open the question of the model’s applicability in domains with low interest update frequencies.

4. Conclusions. The results of the study confirmed that abandoning static models in favor of flexible adaptation of user profiles allows for better content personalization. A mathematical model has been developed that allows for the separation and effective combination of short-term and long-term interests through a system of weighting coefficients and exponential decay. A feature of the model is the integration of contextual parameters (time, device, location) directly into the profile update process, which makes it sensitive to real interaction conditions. A pipeline architecture of the data processing system has been proposed, which includes modules for event collection, streaming processing, and profile update. Testing of the model on the MovieLens dataset demonstrated an increase in the accuracy of recommendations: for short-term interests, the accuracy increased from 0.41 to 0.63, and for new users - from 0.28 to 0.47 compared to static profiles. The high throughput of the system (over 4500 events/s) with an average processing time of one event of 8.4ms has been experimentally confirmed. This proves the suitability of the developed approach for use in high-load services with rapidly changing user interests. The implementation of the proposed methods allows solving the problems of “noisy” actions and blurring of long-term preferences by temporary fluctuations.

Prospects for further research. Further scientific research can be aimed at increasing the interpretability of profile updates, which will make the logic of recommendation formation more understandable for both users and developers. The development of the proposed approach may also involve the integration of deep learning methods or graph neural networks to identify more complex hidden connections in behavioral patterns.

Research limitations. Despite the experimentally confirmed positive test results, there are certain aspects that outline the limits of the application of the developed model. The fact that the conclusions are based on the analysis of user behavior within the film industry, where queries are highly dynamic, leaves open the question of how the system will behave in industries with more stable interests, for example, in professional training or scientific services. Also, the practical implementation of the model requires a powerful technological base for instant data processing. If the organization does not have tools for working with information flows in real time, the benefits of dynamic profile adaptation will be lost due to system delays. In addition, the current version of the algorithm involves the participation of specialists in the process of selecting optimal weighting factors. Creating a mechanism that would allow the system to independently and personalized select these parameters for each person is a task for future developments. It should also be taken into account that the experiment used artificially transformed data that simulates user activity. This approach, while valid for tests, may not take into account certain complex behavioral responses that occur in the real environment.

Conflict of interest. The authors declare that they have no conflict of interest regarding this research, including financial, personal, authorship, or any other nature that could influence the research and its results presented in this article.

Use of artificial intelligence. The authors confirm that they did not utilize artificial intelligence technologies in the creation of the submitted work.

Primary data and materials. The manuscript has data included as supplementary electronic material: Movielens dataset. Kaggle, URL: https://www.kaggle.com/datasets/ayushimishra2809/movielens-dataset (date of application: 10.10.2025).

Financing. The research was conducted without financial support.

Authors' contributions. Oksana Zolotukhina: conceptualization, methodology, formal analysis, writing – review & editing; Olga Cholishkina: formal analysis, validation, visualization; Oleksandr Illyuchenko: methodology, data curation, software development, conducting experiments, writing the initial draft; Taras Lysenko: resources, conducting experiments, testing software code components, validation.

REFERENCES

1. Aggarwal, C. C. (2016). *Recommender systems*. Springer. https://doi.org/10.1007/978-3-319-29659-3
2. Guo, K., & Zeng, G. (2023). Graph convolutional network and self-attentive for sequential recommendation. *PeerJ Computer Science, 9*, e1701. https://doi.org/10.7717/peerj-cs.1701
3. He, X. (2024). Graph neural networks in recommender systems. *Applied and Computational Engineering, 79*, 234–240. https://doi.org/10.54254/2755-2721/79/20241646
4. Wang, H., Li, Y., & Chen, J. (2025). Differential attentive sequential recommendation. *International Journal of High Speed Electronics and Systems*, 2540820. https://doi.org/10.1142/S0129156425408204
5. Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems*. Cambridge University Press. https://www.researchgate.net/publication/235910467_Recommender_Systems
6. Li, K., Tang, Y., Cheng, Y., Bai, Y., Zeng, Y., Wang, C., Liu, X., & Jiang, P. (2025). VQL: An end-to-end context-aware vector quantization attention for ultra-long user behavior modeling. *arXiv preprint*. https://doi.org/10.48550/arXiv.2508.17125
7. Kim, Y. S., Hwangbo, H., Lee, H. J., et al. (2024). Sequence aware recommenders for fashion e-commerce. *Electronic Commerce Research, 24*, 2733–2753. https://doi.org/10.1007/s10660-022-09627-8
8. Ma, L., & Liu, J. (2025). Relationship-enhanced session-based recommendation with graph neural networks. *Information Technology and Control, 54*(4), 1259–1270. https://doi.org/10.5755/j01.itc.54.4.42577
9. Moura, M. do C. de S., Silva, B. L., Sobral, M. F. F., & Ferko, G. P. da S. (2023). Sustainability and technology: Proposals and recommendations for the 2050 Amazon. *Revista de Gestão Social e Ambiental, 17*(1), e03166. https://doi.org/10.24857/rgsa.v17n1-020
10. Movielens dataset. (2025). *Kaggle*. Retrieved October 10, 2025, from https://www.kaggle.com/datasets/ayushimishra2809/movielens-dataset
11. Nedashkovskaya, N., & Androsov, D. (2025). System approach to multicriteria evaluation of session-based and sequential recommendation systems. *KPI Science News, 141*(4), 46–54. https://doi.org/10.20535/kpisn.2025.4.343329
12. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)* (pp. 191–198). https://doi.org/10.1145/2959100.2959190
13. Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In F. Ricci, L. Rokach, B. Shapira, & P. Kantor (Eds.), *Recommender systems handbook*. Springer. https://doi.org/10.1007/978-0-387-85820-3_1
14. Romero Meza, L., & D'Urso, G. (2024). User's dilemma: A qualitative study on the influence of Netflix recommender systems on choice overload. *Psychological Studies, 69*, 349–367. https://doi.org/10.1007/s12646-024-00807-0
15. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys, 52*(1), Article 5. https://doi.org/10.1145/3285029

Відомості про авторів

Англ.	Укр.
Zolotukhina Oksana Anatoliivna Candidate of Technical Sciences, Associate Professor, Taras Shevchenko National University of Kyiv oksana.zolotukhina@knu.ua ORCID: 0000-0002-3314-417X	Золотухіна Оксана Анатоліївна кандидат технічних наук, доцент, Київський національний університет імені Тараса Шевченка oksana.zolotukhina@knu.ua ORCID: 0000-0002-3314-417X.
Cholishkina Olga Hennadiivna Candidate of Technical Sciences, Associate Professor, Taras Shevchenko National University of Kyiv olha.cholyshkina@knu.ua ORCID: 0000-0002-0681-0413	Чолишкіна Ольга Геннадіївна кандидат технічних наук, доцент, Київський національний університет імені Тараса Шевченка, olha.cholyshkina@knu.ua ORCID: 0000-0002-0681-0413
Illyuchenko Oleksandr Serhioivych State University of Information and Communication Technologies illuchenkoo@gmail.com ORCID: 0009-0004-6358-3378	Іллюченко Олександр Сергіойвич Державний університет інформаційно-комунікаційних технологій illuchenkoo@gmail.com ORCID: 0009-0004-6358-3378
Lysenko Taras Oleksandrovych Turbo Stars ORCID: 0009-0003-1591-522X	Лисенко Тарас Олександрович компанія Turbo Stars ORCID: 0009-0003-1591-522X

Дата надходження статті: 27.03.2026
Дата надходження виправленої версії статті: 16.04.2026
Дата прийняття статті: 23.04.2026
Дата публікації статті: 01.06.2026