Artificial Intelligence Applications And Innovations Vol. 2 No. 1 (2025)

Received 16 November 2025
Revised 25 February 2026
Accepted 10 May 2026

Modeling Temporal Dynamics of User Preferences through Multi-Level Similarity in Recommender Systems

Shaho. Kariminejad[1], Mohsen. Ramezani[2*], Shahram. Saeidi[3]

[1] Ph.D. Candidate in Mathematics, Faculty of Science, University of Kurdistan, Sanandaj, Iran
[2] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran
[3] Department of Mathematics, Faculty of Science, University of Kurdistan, Sanandaj, Iran

*** Corresponding author email address**: m.ramezani@uok.ac.ir

Article Info

Article type:
Original Research

How to cite this article:
Kariminejad, S., Ramezani, M., & Saeidi, S. (2025). Modeling Temporal Dynamics of User Preferences through Multi-Level Similarity in Recommender Systems. *Artificial Intelligence Applications and Innovations, 2*(1), 47-64.
https://doi.org/10.61838/jaiai.2.1.4

ABSTRACT

Traditional collaborative filtering systems, which rely on user behavioural similarities, often suffer from fundamental limitations the most significant being their neglect of temporal aspects in data analysis. These systems assume that user preferences remain stable over time, assigning equal weight to both old and recent ratings. However, user tastes can change considerably over time. This paper proposes a time-aware movie recommendation approach that addresses these challenges by intelligently integrating both direct and indirect user relationships. Directly similar users are identified based on historical rating data, while indirectly simi-lar users are discovered through dominant opinion pattern mining. A temporal weighting mechanism is applied to dynamically reduce the influence of outdated interactions, aligning recommendations with evolving user preferences. The incorporation of dominant opinion patterns and the analysis of the target user's preferences further enhance the identification of indirectly similar users. Facilitating interconnections and mediation among users helps to mitigate data sparsity issues. Moreover, incorporating time as a key factor enables the system to effectively manage dynamic user behaviour and reduce its negative impacts. Scalability concerns are also addressed through the utilization of dominant opinion patterns. Ultimately, by analysing each target user's preferences, the proposed model delivers more personalized and ac-curate movie recommendations. Experimental results on several datasets demonstrate that the proposed approach significantly re-duces the Mean Absolute Error (MAE) and improves prediction accuracy compared to conventional methods.

Keywords: *Temporal Dynamics, Recommender Systems, Multi-Level Similarity, User Preference Modeling, Time-Aware Recommendation, Dominant Opinion Patterns*

1. Introduction

In today’s digital world, data is generated at an extraordinary rate, exposing users to an overwhelming amount of information. This abundance creates significant challenges in decision-making, particularly when it comes to selecting relevant content. Recommender systems have emerged as effective tools for addressing this issue by helping users identify content aligned with their interests amid the vast sea of digital information [1]. These systems enhance user experience and foster greater interaction by analyzing user behavior and the characteristics of consumed content.

One of the most prominent applications of recommender systems lies in the media and entertainment industry, where they are widely used to suggest movies, music, and other digital content. Recommender systems are generally categorized into three main types: content-based filtering, collaborative filtering, and hybrid approaches [2]. Content-based systems recommend items by analyzing features such as genre, director, and cast [2]. In contrast, collaborative filtering methods rely on user interactions, such as ratings, to identify similar items based on collective user experiences. These systems are typically divided into user-based and item-based approaches, with user-based methods receiving greater attention and serving as the primary focus of this study.

More advanced collaborative filtering techniques incorporate neural networks and deep learning to model complex user preferences and perform textual analysis of user reviews [3]. However, the results achieved by these methods often do not justify their computational complexity. Hybrid systems, which combine multiple recommendation strategies, have been introduced to improve accuracy and adaptability [3]. A common and critical limitation shared by these approaches, especially in collaborative filtering, is their lack of attention to temporal factors in the recommendation process.

Collaborative filtering algorithms generally rely on static user-provided data such as item ratings. However, they often overlook the gradual evolution of user preferences and the time-sensitive nature of content, both of which can significantly degrade recommendation accuracy. For instance, certain content types, such as news articles, purchase suggestions, or movies (the focus of this paper), have limited lifespans. Ignoring temporal dynamics in such cases can negatively affect the overall user experience. Incorporating time into recommendation models has been shown to improve both accuracy and relevance. Time-aware methods, including those based on Long Short-Term Memory (LSTM) networks, have been employed to model changes in user preferences over time [4]. Nevertheless, these approaches have not consistently delivered satisfactory results, often due to inadequate structures for defining user similarity and modeling opinion shifts over time.

Beyond the neglect of time, recommender systems also face other major challenges, most notably data sparsity. This issue arises when there is insufficient data about users or items, such as when users provide few ratings or when items receive limited feedback. In such scenarios, systems may struggle to accurately capture individual user preferences. One potential remedy is to identify direct similar users and use them as intermediaries to uncover indirect similar users, thereby alleviating the sparsity problem [5]. Scalability is another common challenge: as new users or items are introduced, the system must recalculate similarity relationships, resulting in increased computational demands. Although recent efforts involving multi-objective optimization and reinforcement learning have aimed to improve recommendation diversity, they have yielded only marginal improvements in overall recommendation quality [6].

This paper addresses a key limitation of most existing models, namely, their reliance solely on direct user-item interactions, regardless of time. This limitation also affects approaches like the one proposed by Ramezani et al. [5], which relies on identifying direct and indirect similar users. Accordingly, the goal of this research is to develop an optimized model for rating prediction that improves recommendation accuracy through more precise temporal analysis of user preferences.

Our proposed solution is a hybrid, multidimensional approach that consists of two main components: (1) a collaborative filtering mechanism to identify user similarity patterns, including both direct and indirect relationships, and (2) content-based filtering that incorporates movie features such as genre and user age preferences. In this model, user similarity patterns are extracted from temporal review data. Reference users, those with a larger history of interactions, serve as mediators to connect users who have no overlapping items, thus helping to discover indirect similar users and mitigate the sparsity problem. The model also employs a dynamic weighting strategy, incorporating criteria such as elapsed time, users' affinity to learned opinion patterns, and genre compatibility. By combining these elements, we construct a comprehensive, multidimensional model that enables more accurate prediction of ratings for previously unseen items through holistic analysis of user and item data.

This approach seeks to address several key challenges in movie recommendation systems. It alleviates the cold-start problem by incorporating genre compatibility with age groups and analyzing content features such as genre and release year. Temporal dynamics are addressed through the use of time-based indexing in selecting both direct and indirect similar users, with more recent ratings given greater weight in the prediction process. Data sparsity is tackled by leveraging reference users to mediate indirect similarity discovery, based on the preferences of the target user. Lastly, scalability is enhanced by clustering users into dominant opinion patterns and assigning them to these clusters for similarity calculation, reducing computational complexity. The main contributions of this work are as follows:

1. **Multi-level similarity modeling via dominant behavioral patterns:** We introduce a multi-level user similarity model based on the extraction of dominant opinion/behavior patterns. Unlike conventional approaches that compute only pairwise similarity using measures such as cosine similarity or Pearson correlation, the proposed method operates at the level of shared behavioral structures. Instead of relying directly on raw rating values, it exploits the overlap between users' preferred and non-preferred item sets to capture common behavior patterns. This design yields higher stability and effectiveness, particularly in sparse rating environments.
2. **Structured discovery of indirectly similar users through bridge users:** We develop a structured mechanism for identifying indirectly similar users. Users who simultaneously belong to multiple dominant patterns act as intermediary (bridge) users, creating links between users associated with different patterns. In this way, indirect similarity is systematically discovered based on the pattern structure. Furthermore, the influence of indirect neighbors is controlled via a dedicated weight parameter (τ), ensuring that expanding the neighborhood does not introduce excessive noise or degrade prediction accuracy.
3. **Unified and controllable weighting framework for similarity and context:** We propose an integrated weighting framework that jointly incorporates four components: direct similarity, indirect similarity, temporal dynamics, and age-based contextual compatibility. These components are combined through a normalized convex formulation ($\alpha + \beta + \theta = 1$), which enables fine-grained and balanced adjustment of each factor's contribution to the final prediction.
4. **Enhanced robustness to data sparsity via set-based overlap modeling:** By relying on structured, set-based overlap among item sets, rather than solely on correlations between rating values, the proposed approach exhibits increased robustness in the presence of sparse data. This design choice allows the system to extract meaningful similarity signals even when explicit rating interactions are limited.

The rest of the paper is organized as follows: Section 2 reviews related literature and background concepts. Section 3 introduces the proposed methodology in detail. Section 4 presents experimental results and compares the model's performance with existing approaches. Finally, Section 5 concludes the paper by highlighting key findings and discussing potential directions for future research.

2. Related works

One of the most widely used approaches in recommender systems is collaborative filtering, which predicts user preferences based on the behavior and interests of similar users. Collaborative filtering does not require explicit information about item attributes; instead, it relies solely on user interactions such as ratings or feedback to generate recommendations. The main collaborative filtering techniques are typically categorized into three groups: similarity-based methods, matrix factorization methods, and learning-based methods [7].

The core idea behind similarity-based methods is that if two users have assigned similar ratings to a set of items in the past, they are likely to share similar opinions on other items as well. Similarity is calculated using a variety of metrics, such as Pearson correlation coefficient, cosine similarity, Euclidean distance, Jaccard coefficient, Bhattacharyya coefficient, and other similarity measures [7]. Popular algorithms within this category include k-nearest neighbors (k-NN) and clustering-based approaches.

Singh et al. [8] employed a modified Bhattacharyya coefficient to compute similarity between items, treating each item as a probability distribution. In their approach, similarity between users was used as a weight to identify relevant neighbors. Hu et al. [9] proposed a novel similarity metric based on user-to-user relations, user-item interactions, and user interests. Their method extracted user similarity by integrating Pearson correlation and vector space similarity. Kumar et al. [10] introduced two new similarity measures called RJaccard and RJMSD, which consider the full rating vectors of users to identify relevant neighbors. These metrics were shown to improve recommendation quality by more accurately capturing user similarity. Similarly, Hayashi et al. [11] used an enhanced version of the Pearson similarity metric to improve item recommendations. Their approach involved techniques such as normalization, missing value estimation, and optimal weighting to enhance the accuracy of Pearson-based similarity.

In several studies, the k-Nearest Neighbors (kNN) method has been employed to identify the top-k users or items similar to a given user or item. Tang et al. [12] enhanced the traditional kNN by integrating it with the advanced Pearson correlation coefficient. Their method applies a similarity threshold to filter out less relevant users, aiming to retain only the most effective similar users. Ramazani et al. [5] proposed a method called WABR, which extracts diverse interest patterns to identify both direct and indirect similar users. In this approach, users who have interacted with a large number of items serve as intermediaries between users who have no common interactions. However, these methods overlook the importance of pattern relevance and consider all extracted patterns equally, potentially incorporating less reliable similar users. In a more recent study, Ramazani et al. [13] considered the influence of indirect similar users in rating prediction by assigning them lower weights compared to direct similar users. However, this method does not adequately account for the individual preferences of the target user in the recommendation process.

Clustering algorithms have also been employed in some studies to identify similar users. Yang et al. [14] introduced a novel clustering algorithm named U-k-means, an unsupervised variant of k-means, which automatically determines the optimal number of clusters without requiring initialization or parameter tuning. Another clustering technique used is K-Medoids, which is similar to k-means but utilizes actual data points as cluster centers, making it more robust to outliers. As a novel approach to collaborative filtering, Feng et al. [15] introduced a distributed Item CF algorithm that combines a distributed data processing model with a distributed data management system to enhance the efficiency and scalability of the recommendation process. Zhang et al. [16] also focused on modeling complex relationships between items. Unlike traditional approaches that consider only item-item or user-user similarities, their study highlights the significance of multifaceted item relationships such as semantic, temporal, or contextual connections.

Matrix factorization has been another prominent approach in collaborative filtering for uncovering latent patterns in user-item interaction data, such as movie ratings. Vander et al. [17] applied Bayesian Probabilistic Matrix Factorization (BPMF), which uses probabilistic modeling to detect hidden structures in sparse data. By placing probability distributions over feature vectors, this method reduces model error and improves accuracy. Chen et al. [18] introduced an Alternating Least Squares Matrix Factorization (ALS-MF) method that iteratively optimizes and updates user and item feature matrices to enhance prediction accuracy. In another approach, Deng et al. [19] proposed Bias-aware Matrix Factorization to predict missing ratings by considering inherent biases of users and items, such as a user's tendency to rate higher or lower. This method incorporates individual user and item biases into the model for more accurate predictions.

Li et al. [20] introduced an Online Collaborative Filtering approach with a dynamic regularizer, which considers varying factors such as dynamic rating averages, shifting user preferences, item characteristics, and score distributions. Berkani et al. [21] proposed NeuMF, a neural matrix factorization model that integrates Generalized Matrix Factorization (GMF) and a Multi-Layer Perceptron (MLP). GMF embeds users and items in a vector space to model linear interactions, while MLP captures complex nonlinear relationships through multiple neural layers. The outputs of both models are then combined to predict the probability of user-item interactions. Furthermore, Liu et al. [22] proposed Context-aware CF, a framework for learning context-based latent representations. This method can be seen as a general model combining Bayesian Matrix Factorization (BMF) and Bayesian Personalized Ranking (BPR).

As another direction, learning-based methods aim to train models to predict unknown ratings. Wang et al. [23] proposed GraphGAN, a framework that utilizes generative and discriminative models to learn a graph-based representation. Although it performs well in link prediction, its recommendation accuracy is unstable and not comparable with more recent methods. Due to limitations in user-item information, graph-based approaches face challenges in improving performance. To address this, Jiao et al. [24] developed a deep learning framework leveraging knowledge graphs and implicit feedback to model entities and knowledge structures. Generally, graph-based methods perform weaker than similarity-based or matrix factorization approaches. Wang et al. [25] also employed generative and discriminative retrieval models within a minimax game to generate item recommendation lists, using maximum likelihood estimation as part of their strategy.

Genetic algorithms have also been used in recommender systems to simulate and optimize user-item similarities. These algorithms aim to compute similarities more effectively for better recommendations. In this regard, Keilani et al. [26] proposed a method that evaluates recommendation lists rather than individual items. It also incorporates semantic information and item rankings, outperforming other evolutionary and item-based methods. However, this approach falls short in runtime and recommendation quality compared to newer similarity-based and learning-based methods. Among other recommendation categories, some methods rely on implicit feedback for item suggestions.

User preferences often change over time, making temporal factors crucial in recommendation, which many existing systems tend to overlook. Ahmadian et al. [27] incorporated temporal reliability and trust metrics into the recommendation process. First, user rating effectiveness is assessed using a probabilistic approach, and uninformative profiles are reinforced with implicit scores. Then, predictions are evaluated through temporal reliability, and low-trust ratings are recalculated via neighborhood updates to enhance accuracy. In another study, Harshavardhan et al. [28] proposed a Time-aware Deep Collaborative Filtering method involving two phases: dynamic modeling of user preferences and rating prediction. Here, short-term preferences are modeled using a time-aware attention mechanism and are integrated with long-term preferences. The user-item interactions are then learned through two deep learning models, DMF and MLP, to estimate matching scores. Jain et al. [29] also developed an effective recommender system that captures temporal changes in user preferences. They used exponential and power decay functions combined with similarity metrics such as cosine similarity, Pearson correlation, and Gower coefficient. However, this method did not yield significantly notable results in practice.

Compared to previous approaches, the proposed method in this study offers several key advantages. By combining direct and indirect user similarity with dynamic weight analysis, it delivers more personalized and interactive recommendations. Unlike deep learning-based models that require large datasets for training, the proposed model enhances both accuracy and efficiency with lower computational costs. It is capable of outperforming existing methods on real-world data and addresses data sparsity and dynamic user behavior challenges. In addition to analyzing direct and indirect similarities, the method considers behavioral changes over time to provide the most relevant recommendations based on users' evolving characteristics. Moreover, by leveraging dominant opinion patterns and more informed mediation by indirect similar users, it effectively alleviates data sparsity issues.

The proposed method is compared to various baseline techniques. For instance, PMF uses probabilistic modeling of user-item interaction matrices and Gaussian priors over latent factors, outperforming traditional approaches [30]. BPMF model that enhances uncertainty modeling using a Bayesian framework is another comparing method [31]. As another comparing method, NNMF applies non-negativity constraints on user and item features, improves interpretability and performance on sparse data [32]. SVD++ [33] is another proper state-of-the-art factorization based method which incorporates implicit feedback to improve prediction accuracy over traditional SVD. In neighborhood-based collaborative filtering, methods like UBBCF perform well in environments lacking explicit ratings by focusing on binary interactions (e.g., viewed or not viewed) and user similarity [34]. RMIF method uses implicit feedback (clicks or views) instead of explicit ratings to deliver more effective recommendations [7].

In leveraging graph and contextual knowledge, RippleNet method extracts contextual information by expanding user interests through knowledge graphs [35]. Models like KGAT [35] and KGCN [36] combine knowledge graphs with attention or convolutional learning to model rich semantic interactions between users and items. In deep learning-based models, ONCF [37] modeled high-order interactions through outer product instead of vector concatenation. MLP model captures nonlinear user-item features for recommendation [37]. NCF model integrates MF and MLP into a flexible framework for personalized recommendations [38]. Finally, Shirali et al. [39] applied alternating optimization to model parameters, offering an effective approach for multi-objective recommendation problems.

In summary, the vast body of related work in recommender systems demonstrates the dynamic evolution from classical similarity-based techniques and matrix factorization to sophisticated learning-based and context-aware models. While each category has presented innovations, from metric augmentation and hybrid clustering to the integration of temporal and semantic information, existing methods often face limitations related to data fragmentation, scalability, and adaptability to changing user preferences. The proposed approach differentiates itself by integrating direct and indirect user similarities with dynamic weighting and temporal analysis, providing more adaptive and personalized recommendations. By addressing the shortcomings of previous models and effectively balancing the accuracy, efficiency, and computational cost of recommendations, this approach brings together recent advances and responds to important challenges in modern recommendation tasks.

3. Proposed Method

This section presents the proposed algorithm and its implementation, outlining the sequential steps of the method. The adopted approach combines the identification of both direct and indirect similar users, the temporal effects of rating timestamps, and users' age-related preferences. In recommender systems, similarity measures play a fundamental role in comparing and analyzing users or items. These measures enable the system to identify relevant items or similar users, thereby generating more accurate and personalized recommendations. The choice of similarity metric depends on the nature of the data, the objectives of the recommender system, and the level of complexity required.

Many similarity measures perform poorly in systems with sparse data, where user-item interactions are limited. In such scenarios, identifying overlapping items between users becomes challenging or even infeasible for a significant portion of the dataset. Moreover, conventional similarity metrics are often incapable of capturing or modeling the complex patterns of user-item interactions. To address the challenges posed by data sparsity and reduce its impact on the outcomes, our method computes user and item similarities based on set overlap and introduces mediated similarity computation by considering as many relevant variables as possible. This strategy significantly reduces dependency on large-scale data and is relatively simple to implement.

In this work, two matrices are considered, the rating matrix and the timestamp matrix, as illustrated in Table 2. Each entry $t_{i,j}$ in the timestamp matrix represents the time at which user *i* submitted their rating for movie *j*. Based on the rating matrix, a set of preferred and non-preferred items is constructed for each user. For a given user, movies rated 3 or higher are categorized as preferred items, while movies rated below 3 are treated as non-preferred items. According to the notations defined in Table 1, let U denote the set of users and M denote the set of movies, each identified by unique identifiers. Let $r_{u,i}$ be the rating assigned by user *u* to movie *i*. F_u is define as the set of preferred items for user *u*, and U_u as the set of *non-preferred items* for the same user. These two sets are formally constructed for each user *u* according to Equations (1) and (2).

$$F_u = \{I_j | r_{u,j} \geq 3\} \tag{1}$$

$$U_u = \{I_j | 1 \leq r_{u,j} < 3\} \tag{2}$$

Table 1. List of variables in the proposed method

Symbol	Explanation
U	User set
M	Movie set
m	Number of users
n	Number of movies
$\boldsymbol{r_{u,i}}$	Rating given by user *u* to movie *i*
$\boldsymbol{F_u}$	Set of liked movies of user *u*
$\boldsymbol{U_u}$	Set of disliked movies of user *u*
$\boldsymbol{\gamma}$	Overlap index
$\boldsymbol{\varphi}$	Set frequency
$\boldsymbol{\eta}$	Total number of non-empty sets in the overlap table
P	Pattern
δ	Influence of direct similar users in rating estimation
τ	Influence of indirect similar users in rating estimation
ϵ	Degree of movie overlap between indirect similar users and reference users
λ	Threshold for filtering sets
$\boldsymbol{n_{es}}$	Number of age-appropriate movies for the user
α,β,θ	Weight balancing coefficients
$\boldsymbol{w_{delta}}$	Delta weight
$\boldsymbol{w_{Time}}$	Time weight
Ω	Exponential time function parameter

In these formulations, I_j denotes item *j*, and $r_{u,j}$ represents the rating assigned by user *u* to item *j*. This categorization aims to identify the movies that are perceived as *attractive or favorable* by the user (i.e., the preferred set) versus those perceived as less appealing (i.e., the non-preferred set). Subsequently, the shared preferences between each pair of users are computed and organized into a structure referred to as the Overlap Matrix (Table 3). In this table, $CF(u_i, u_j)$ and $CU(u_i, u_j)$ represent the cardinality of the intersection between the *preferred sets* and *non-preferred sets*, respectively, of the user pair $(u_i\ ,u_j)$. This comprehensive matrix provides a clear perspective on the commonalities and divergences in users' preferences, capturing both aligned interests and disagreements across preferred and non-preferred item sets.

Table 2. Time matrix and rating matrix

TIME MATRIX	$\boldsymbol{I_1}$	$\boldsymbol{I_2}$	…	$\boldsymbol{I_N}$
$\boldsymbol{U_1}$	$t_{1,1}$	$t_{1,2}$	…	$t_{1,N}$
$\boldsymbol{U_2}$	$t_{2,1}$	$t_{1,2}$	…	$t_{2,N}$
…	…	…	…	…
$\boldsymbol{U_M}$	$t_{\mathrm{M},1}$	$t_{\mathrm{M},N}$	…	$t_{\mathrm{M},N}$

RATING MATRIX	$\boldsymbol{I_1}$	$\boldsymbol{I_2}$	…	$\boldsymbol{I_N}$
$\boldsymbol{U_1}$	$r_{1,1}$	$r_{1,2}$	…	$r_{1,N}$
$\boldsymbol{U_2}$	$r_{2,1}$	$r_{1,2}$	…	$r_{2,N}$
…	…	…	…	…
$\boldsymbol{U_M}$	$r_{\mathrm{M},1}$	$r_{\mathrm{M},N}$	…	$r_{\mathrm{M},N}$

Then, empty and duplicate entries in the overlap matrix are removed. Hereafter, the remaining item sets are sorted based on their size. Subsequently, the overlap matrix is partitioned into two submatrices: the Preferred Local Matrix and the Non-Preferred Local Matrix. In the following step, the most frequently recurring item sets across user opinions are extracted and designated as the dominant opinion sets. These dominant sets are typically identified by filtering out non-informative or low-impact subsets. To facilitate this process, a metric known as the Overlap Index is employed. The first is the Global Overlap Index, which measures how widely a particular item set is shared among all users. Let γ denote the Overlap Index of the given set, φ its frequency, and η the total number of non-empty sets in the Overlap Table. This metric is computed as shown in Equation (3). This ratio is instrumental in identifying the most popular item sets or behaviors across the entire dataset, offering a global perspective on shared user interests.

Relying only on overlapping itemsets for similarity is insufficient, as users from different age groups may coincidentally watch a few common movies despite having fundamentally different preferences. To address this, the index not only calculates itemset overlap but also accounts for user age distance, considering only pairs whose age difference is within five years. This ensures the analysis reflects genuine shared interests within similar demographic groups, such as children preferring animation and teenagers favoring action movies.

Table 3. Overlap table

$(\boldsymbol{u_i},\boldsymbol{u_j})$	$\boldsymbol{CF}(\boldsymbol{u_i},\boldsymbol{u_j})$	$\boldsymbol{CU}(\boldsymbol{u_i},\boldsymbol{u_j})$
$(\boldsymbol{u_1},\boldsymbol{u_2})$	$F_{u_1} \cap F_{u_2}$	$U_{u_1} \cap U_{u_2}$
$(\boldsymbol{u_1},\boldsymbol{u_3})$	$F_{u_1} \cap F_{u_3}$	$U_{u_1} \cap U_{u_3}$
…	…	…
$(\boldsymbol{u_2},\boldsymbol{u_3})$	$F_{u_2} \cap F_{u_2}$	$U_{u_2} \cap U_{u_3}$
$(\boldsymbol{u_2},\boldsymbol{u_4})$	$F_{u_2} \cap F_{u_3}$	$U_{u_2} \cap U_{u_4}$
…	…	…
$(\boldsymbol{u_i},\boldsymbol{u_j})$	$F_{u_i} \cap F_{u_j}$	$U_{u_i} \cap U_{u_j}$
…	…	…
$(\boldsymbol{u_{N-1}},\boldsymbol{u_N})$	$F_{u_{N-1}} \cap F_{u_N}$	$U_{u_{N-1}} \cap U_{u_N}$

$$\gamma_{\text{Total Overlap Index}} = \frac{\varphi}{\eta} \qquad (3)$$

The Age-Aware Overlap Index is an advanced metric that improves upon traditional similarity measures by considering both shared items and the age gap between users. It is based on the principle that users of similar ages share more meaningful preferences. This results in more precise user modeling and better recommendations by focusing on behavioral patterns within closer age groups..if φ_{age} denotes the frequency of item co-occurrences in the overlap table where the age difference between users at the time of rating does not exceed a predefined threshold, Equation (4) can be defined as follows:

$$\gamma_{age} = \frac{\varphi_{age}}{\eta} \qquad (4)$$

User preferences shift over time, so similarity analysis must be time-sensitive rather than relying solely on all historical ratings. A time-aware overlap index becomes crucial here, as it reflects users' current tastes and detects recent preference changes. By incorporating a time dimension, this metric tracks behavioral changes and identifies users with comparable patterns within specific time windows. It helps detect emerging trends and ensures recommendations remain relevant. For example, if two users rated the same movie similarly but years apart, ignoring this time gap could lead to inaccurate recommendations, as the earlier user's interests may have evolved. If $\varphi_{\mathrm{V}v}$ represents the frequency of co-rated items where the difference in rating timestamps does not exceed a predefined threshold, then Equation (5) is defined as follows:

$$\gamma_{vt} = \frac{\varphi_{\mathrm{vt}}}{\eta} \qquad (5)$$

In other words, to increase the accuracy of the recommendations, an overlap index is defined that is called the time-aware frequency, which is a combination of the three previous overlap indices. In this index, the item-sets are counted in the overlap table that, in addition to containing the specific itemset, correspond to two users who are approximately of the same age (i.e., in a similar age group) and who have rated that movie within a specified time window.

Using a threshold parameter denoted by λ, the candidate sets are filtered based on their overlap index. This hyperparameter serves to reduce and refine the sets by retaining only those with a high overlap index or those that sufficiently capture meaningful similarities. If the overlap index of a set exceeds the λ threshold, that set is selected as a dominant set. Subsequently, sets that are repeated within other sets are removed, ensuring that only independent and meaningful sets are retained. In other words, non-dominant sets are eliminated from the algorithm.

Here, by combining desirable and undesirable dominant sets, dominant patterns are extracted. These patterns are particularly useful for quickly assigning new users to one of the existing patterns without the need for intensive computations, thereby mitigating the cold-start problem and enhancing scalability. Ultimately, the extracted patterns are used to uncover relationships among users. For instance, a dominant pattern with a high overlap index influenced by age represents a group in which users with small age differences exhibit highly similar preferences.

After identifying the patterns, user assignment to the dominant opinion patterns is performed by evaluating the behavioral alignment of each user with the union of preferred and non-preferred item sets within each pattern. For new users, this assignment can alternatively be done based on their own selection. For every user–pattern pair, the number of movies watched by the user that intersect with the union of the pattern’s preferred and non-preferred item sets is computed to determine the user's alignment percentage with the pattern. Users whose alignment exceeds or meets a predefined threshold are assigned to the respective pattern. The pattern-membership threshold σ (user affiliation to a dominant pattern) was varied in {0.5, 0.6, 0.7, 0.8, 0.9}.

However, selecting an appropriate threshold is critical: if the threshold is set too high, only a small number of users may be matched with patterns, potentially omitting valuable and meaningful user relationships. Conversely, a very low threshold might lead to the inclusion of irrelevant users in the patterns, thereby significantly reducing the accuracy of the method. Thus, careful tuning of this hyperparameter is essential to optimize the assignment process.

Establishing connections among users and employing mediation for movie recommendation is an effective approach to enhance the accuracy and overall quality of recommender systems. In this framework, direct similar users, those assigned to the same dominant pattern, exhibit similar behaviors or rating tendencies and are directly connected through the shared pattern. Indirect similar users, on the other hand, are linked via reference users who serve as intermediaries. Combining direct and indirect similar users enables access to a broader range of information, improving the accuracy of rating estimations. Furthermore, indirect users can recommend items that have not yet been considered by others within a user's immediate neighborhood, significantly increasing the variety of recommendations. This approach facilitates the effective utilization of hidden user preferences and expands the system's recommendation potential.

In this study, direct similar users are first identified based on dominant opinion patterns or item sets. Among these direct similar users, if a user appears in more than one pattern, they are designated as a reference user. Reference users serve as intermediaries for connecting indirect similar users, those who are associated with patterns related to the reference user. Through these intermediaries, a group of indirect similar users is discovered.

In addition to this group, another category of indirect similar users is identified based on the interests and preferences of the target user, the user for whom the recommendation is being generated. These users are referred to as indirect similar users based on target user preferences. To identify them, the preferences of the target user in terms of genre and movie production year are first analyzed. Then, users who exhibit similar preferences, i.e., an interest in the same genres and production years, are selected. For this purpose, movies are classified into three categories based on their production year, as outlined in Table 4.

The final set of indirect similar users is obtained by taking the union of the two previously defined groups. These indirect similar users provide additional insight into the potential preferences of the target user, particularly in cases where data from direct similar users is insufficient. The inclusion of indirect similar users enhances the accuracy of rating predictions, enriches the diversity of information regarding movies, and increases data coverage, expanding the range of items for which predictions can be made.

The classification of users into groups (direct, reference, and indirect) plays a pivotal role in improving the performance of the recommender system. Such grouping facilitates a more precise identification of relationships among users and enables more effective utilization of their associated data. Moreover, this approach empowers the recommender system to make intelligent use of sparse data, uncover latent user interests, and deliver a more personalized experience to each user. To further refine the accuracy of rating estimation, reduce prediction errors, and enhance user personalization, a set of weighting factors is introduced. These weights allow the model to intelligently incorporate multiple dimensions, including the relevance of the similar user group, the similarity between users, and the appropriateness of movies with respect to user age.

One such weight is the delta coefficient (Equation 6), which quantifies the influence of direct similar users in the rating estimation process and acts as a membership strength indicator linking a user to a specific pattern. The delta value reflects the degree to which a user's behavior aligns with the associated dominant pattern. Users with higher delta values are considered more influential, as a larger portion of their preferences match the movies in the pattern. Conversely, users with lower delta values have reduced impact in the calculations, thereby helping to prevent noise from distorting the accuracy of predicted ratings.

$$\delta_{u,p} = \frac{|F_u \cap \mathrm{P}| + |U_u \cap \mathrm{P}|}{|\mathrm{P}|} \tag{6}$$

Let P denote a pattern, F_u the set of preferred items for user u, and U_u the set of non-preferred items for the same user. The weight τ (Equation 8) is introduced to regulate the influence of indirect similar users during the prediction phase of unknown ratings. This weight represents the degree of behavioral overlap between indirect similar users and the target user. It is computed based on the parameter ϵ (Equation 7), which quantifies the similarity in preferences between users. The value of τ is determined by the relationship between ϵ and the size of the item sets viewed by the users.

Users with higher τ values exert greater influence on the predicted ratings, while those with lower τ values have less impact, thereby reducing potential noise in the prediction process. Importantly, τ values are calibrated such that indirect similar users are assigned lower influence than direct similar users. In this context, $S_{\mathrm{Ref,Ind}}$ refers to the intersection of items rated by both the reference user and the indirect similar user, while $S_{\mathrm{Ref,Tar}}$ denotes the overlap between the items rated by the reference user and the target user.

$$\epsilon = |S_{\mathrm{Ref,Ind}}| + |S_{\mathrm{Ref,Tar}}| \tag{7}$$

$$\tau = \left(\frac{\epsilon}{|S_{U_{\mathrm{Ind}}}|}\right)^2 + \left(\frac{\epsilon}{|S_{U_{\mathrm{Tar}}}|}\right)^2 \tag{8}$$

Time-aware recommender systems, which are the focus of this study, incorporate temporal dynamics into the recommendation generation process. These systems analyze user behavior over different time intervals, taking into account temporal variations in user preferences and other time-dependent factors, to produce more accurate and contextually relevant recommendations. For instance, users may exhibit varying interests throughout the day, across seasons, or during specific time periods, highlighting the importance of modeling temporal user preferences.

In the proposed approach, we aim to assign greater weight to items, such as movies, that have been recently rated or watched, in order to better reflect users' current interests. Items like movies typically experience popularity cycles over time. To model this temporal influence, a time decay function is introduced in Equation (9), which captures the effect of time on user preferences and the weighting of data. In this function, Ω is a tunable decay rate parameter, and Δt represents the time difference between the current moment and the timestamp of the recorded rating. The time-decay rate Ω in the exponential function was explored in {0.005, 0.01, 0.02} in following equation:

$$w_t = e^{(-\Omega \Delta t)} \tag{9}$$

On the other hand, items, particularly movies, often vary in their appeal to different age groups based on genre or content. For instance, adolescents are typically more interested in genres such as action or animation, whereas older adults may prefer documentaries or historical films. To enhance personalization and improve the accuracy of preference prediction, this study introduces an age-appropriateness weighting factor, referred to as the contextual age-based weight. This weight assigns lower values to movies that are not age-appropriate for the target user, allowing the model to place greater emphasis on data that aligns with the user's age group. As a result, the recommendations are better tailored to the individual’s age, fostering a sense of relevance and resonance with the user’s interests and needs. Moreover, this mechanism prevents the recommendation of unsuitable content for certain age groups and prioritizes age-appropriate items. This leads to a more realistic and satisfying user experience. If we denote the number of age-appropriate movies watched by the user as n_{es}, and the total number of movies watched as $|S_k|$, then Equation (10) defines the contextual weight as follows:

$$w_{es}=0.2 + 0.8 * \frac{n_{es}}{|S_k|} \qquad (10)$$

In the above formulation, the values 0.2 and 0.8 are adjustable parameters that can be tuned based on the characteristics of the dataset. This method assigns a minimum initial weight, set to 0.2 in this case, to all users to ensure that even those who have not watched age-appropriate movies still retain some level of influence in the recommendation process. This safeguard prevents the complete exclusion of user opinions and ensures that partially relevant or exceptional cases are not entirely disregarded.

When a user has watched a higher proportion of age-appropriate movies, a greater weight is assigned, thereby increasing their influence on rating prediction. Conversely, users with a lower proportion of age-appropriate content are given reduced weight, as their behavior does not align with the expected age-based pattern. In this study, movie genres are categorized according to target age groups, as outlined in Table 4. Accordingly, similar users who have consumed a larger number of movies from genres appropriate for their age are given higher importance, exerting more influence on both the rating predictions and the recommendations generated by the system.

Ultimately, after calculating the various weights, each capturing different dimensions of user preferences, the prediction of unknown ratings is performed through a linear combination of these weights. In this approach, a parameter **α** is introduced to balance the influence of different weight components. The inclusion of α provides greater flexibility, enabling the method to adapt effectively across different scenarios while maintaining a suitable range of personalization in the predicted outcomes.

Table 4. Classification of preferred movie genres for different age groups

Preferred Genres	Description	Age Group
Action, Adventure, Comedy, Animation	This group includes teenagers who typically enjoy thrilling and entertaining genres	Ages 13–18
Action, Sci-Fi, Fantasy, Drama	This group consists of young adults who tend to prefer science fiction, emotional dramas, and fantasy stories	Ages 19–29
Drama, Romance, Thriller	This group represents middle-aged individuals with a tendency toward deeper and more complex genres	Ages 30–49
Historical, Documentary, Drama	This group includes older adults who often appreciate genres with historical depth, documentaries, or emotionally engaging narratives	Ages 50 and above

This parameter is computed based on the proportion of relevant genres to the total number of movies watched by the user, ensuring that the final recommendations are not biased by disproportionate genre exposure. Consequently, the final similarity weight assigned to each user in the rating prediction phase can be obtained as a composite of the calculated weights. For instance, the weight of a similar user can be computed by combining the delta weight and the time-aware impact weight as defined in Equation (11), or alternatively, as shown in Equation (12), by integrating temporal weight, delta, and age-relevance weight. These final similarity weights, used for estimating unknown ratings, are thus constructed as a linear combination of multiple influencing factors, each contributing to a more accurate and personalized prediction. For the linear combination of the time, delta, and age-appropriateness weights in Eq. (12), triplets (*α, β, θ*) are chosen satisfying Eq. (13) with *α, β, θ* ∈ {0.1, 0.2, …, 0.8}.

$$w_{time-delta} = \alpha * w_{delta} + (1-\alpha) * w_{time} \qquad (11)$$

$$w_{time-delta-es=} \alpha * w_{time} + \beta * w_{delta} + \theta * w_{es} \qquad (12)$$

$$\alpha + \beta + \theta = 1 \qquad (13)$$

It is worth noting that the delta weight pertains to direct similar users, while for indirect similar users, this value is equal to the weight *τ*. After calculating the weight for each similar user, the unknown score can be computed using equation 14, where the weight of the similar user can be considered using one of the weights of time, delta, and contextual, or a combination of them, as indicated in equation 11 and 12.

$$ER = \frac{\sum_{i=1}^{N} w_i r_{i,j}}{\sum_{i=1}^{N} w_i} \quad (14)$$

The overall time cost is dominated by comparing each pair of users including: computing the intersection of two item sets takes $O(l)$ time (l = maximum set length), and there are $m(m-1)/2$ pairs, so building the OverlapTable costs $O(l\,m^2)$. In practice l is very small in sparse datasets, so this is much less than $O(n^2)$. After building the table, empty rows are dropped and the frequency (support count) of each unique overlap is counted in linear time in the number of overlap rows as $O(K)$. Here, K is the number of remaining rows in the overlap table. Finally, the remaining non-empty overlaps are processed (sorted or inserted into local like/dislike tables), which adds the usual cost for those operations (for example sorting adds $O(K\ log\ K)$).

4. Experimental Results

In this section, we substantially expand the description of our experimental setup to improve reproducibility. Here, experiments rely exclusively on publicly available datasets and all preprocessing steps are explicitly specified as follows:

(i) Ratings $r_{u,i}$ are used as provided by the datasets.

(ii) For each user $u \in U$, the preferred and non-preferred movie sets F_u and U_u are constructed according to Equations (1) and (2) by thresholding $r_{u,m}$ at 3.

(iii) The Overlap Table is built by computing $F_{ui} \cap F_{uj}$ and $U_{ui} \cap U_{uj}$ for every user pair (u_i, u_j), then empty and duplicate sets are removed; the remaining sets are sorted by their set frequency φ (or size) and split into preferred and non-preferred local tables.

(iv) Dominant sets and patterns P are extracted using the overlap index γ defined in Equations (3)–(5) with threshold λ and user–pattern assignments are then performed based on the membership and alignment rules defined in Equations (6)–(8).

For each dataset, 5-fold cross-validation with an 80/20 train–test split per fold is used, where only ratings in the test portion are predicted. Within each training fold, 10% of the training data is reserved for hyperparameter selection as described in the hyperparameter tuning section, and the finally chosen values (λ=0.4, σ=0.8, Ω=0.01, α=0.5, β=0.3, θ=0.2, k=10 neighbors) are fixed for all reported test-set results. We explicitly state that the weighting factors are computed via Equations (6)–(12); that the combined similarity weights satisfy α+β+θ=1 as in Equation (13); and that unknown ratings are estimated using the weighted average in Equation (14), which also incorporates the influence of indirect similar users via τ and overlap degree ϵ. For each method and dataset, MAE, RMSE, Recall@5, and NDCG@5 are computed as defined in Equations (15)–(19) and averaged over the 5 folds. To facilitate independent replication, all hyperparameters are clearly listed in Table 16.

This section is organized into four subsections: Subsection 4.1 introduces the datasets and evaluation criteria; Subsection 4.2 presents a detailed analysis of the proposed method’s performance; Subsection 4.3 provides a comparative evaluation against alternative approaches; and finally, Subsection 4.4 discusses the analysis of the model’s parameter settings.

4.1. Datasets and Evaluation Metrics

As shown in Table 5, the user and movie data were extracted from the standard MovieLens dataset. The data encompass behavioral information (ratings), demographic details (age, gender), and content attributes (movie genres). The MovieLens 100k dataset contains 100,000 ratings from 943 users on 1,682 movies, with each user having rated at least 20 movies. This dataset is characterized by its relatively small size, making it suitable for rapid experimentation. The MovieLens 1M dataset includes 1,000,209 ratings from 6,040 users on 3,900 movies. Ratings in this dataset are on a [1, 5] scale, with a sparsity ratio exceeding 95%. In the conducted experiments, 80% of each dataset is randomly selected as the training set, while the remaining 20% is used as the test set, employing a 5-fold cross-validation procedure.

Table 5. Information of datasets

Datasets	Users	Films	Ratings	sparsity
ML-100K	943	1682	100000	0.9369
ML-1M	6040	3900	1000209	0.9575

The following formulas (15 to 19) are used to evaluate the methods. In the formulas, *r* represents the actual rating, *p* is the estimated rating, *n* is the number of absolute errors, *TP* indicates the cases correctly recommended by the system, *TN* indicates the cases correctly not recommended by the system, *FP* indicates the cases incorrectly recommended by the system, and *FN* indicates the cases that the system should have recommended but did not. IDCG is the ideal (or perfect) ranking value.

$$MAE = \frac{\sum_{i=1}^{n}|p_i - r_i|}{n} \tag{15}$$

$$RMSE = \left(\frac{\sum_{i=1}^{n}|p_i - r_i|^2}{n}\right)^{\frac{1}{2}} \tag{16}$$

$$Recall = \frac{1}{m}\sum_{u=1}^{m}\frac{TP}{TP + FN} \tag{17}$$

$$\text{Precision} = \frac{1}{m}\sum_{u=1}^{m}\frac{TP}{TP + FP} \tag{18}$$

$$NDCG = \frac{1}{IDCG}\sum_{i=1}^{r}\frac{2^{I_i} - 1}{log_2^{(i+1)}} \tag{19}$$

4.2. Analysis of the Proposed Method

In this section, the proposed method is initially evaluated on two small subsets of the ML-100K dataset, consisting of 100 and 200 users respectively, each randomly selected in five separate runs from the original dataset. The method is implemented using the delta coefficient, the time coefficient, a combined delta-time coefficient, an age-adaptive movie coefficient, and finally a combination of all these coefficients to assess how the proposed approach performs on these data subsets. The results presented in Table 6 indicate that, due to the limited size and scarcity of information in these subsets, the accuracy is not particularly satisfactory. It is also observed that the method performs better on the 200-user subset compared to the 100-user subset. Notably, the combined delta, time, and age-aware weighting scheme (DTA-PM) achieve a significant improvement in accuracy compared to other variants. This is followed by the (DT-PM) scheme, which ranks second, demonstrating the substantial impact of temporal and age-adaptive factors on performance. Generally, integrating temporal dynamics, user preference shifts, and age considerations yields considerably better results than any other combination. The outcomes of the proposed method on the full datasets, shown in Table 7, 8 follow a similar trend.

Table 6. Results of the proposed method with 100 and 200 users from the ML-100K dataset

Proposed Method	100 users		200 users	
	MAE	RMSE	MAE	RMSE
D-PM	0.749	0.934	0.728	0.896
T-PM	0.734	0.912	0.707	0.875
DT-PM	0.688	0.903	0.652	0.830
A-PM	0.751	0.947	0.733	0.884
DTA-PM	0.646	0.887	0.629	0.831

Table 7. Results of the proposed method on the ML-100K and ML-1M datasets

Proposed Method	ML-100k		ML-1M	
	MAE	RMSE	MAE	RMSE
D-PM	0.572	0.762	0.541	0.739
T-PM	0.565	0.749	0.532	0.724
DT-PM	0.549	0.731	0.528	0.719
A-PM	0.585	0.783	0.561	0.757
DTA-PM	0.536	0.714	0.507	0.693

Table 7 demonstrates that the proposed method performs better on the ML-1M dataset compared to ML-100K. This improved performance is attributed to the greater availability of data, enabling the model to identify more patterns, thereby enhancing prediction accuracy and reducing RMSE and MAE. In all variants of the proposed method, in addition to leveraging reference users to identify indirect similar users, the target user's preferences are also employed to discover an additional set of indirect similar users. The (D-PM) method indicates that relying solely on the Delta factor results in moderate error levels. To enhance this method, incorporating additional factors such as time or age-adaptive weighting is necessary. Results of the (T-PM) method show that adding the temporal factor improves system performance, highlighting temporal information provides valuable insights into user behavior and preferences. In the (DT-PM) variant, the combination of the Delta and time coefficients leads to error reduction and improved prediction accuracy. This approach outperforms the previous two as it integrates multiple influential factors in the prediction process. The (A-PM) variant reveals that using the age factor alone has minimal impact on accuracy, likely due to the exclusion of complex behavioral patterns. However, combining age with other factors can improve performance. Finally, the (DTA-PM) method achieves the best performance, significantly reducing prediction errors. This finding suggests that simultaneously integrating multiple effective coefficients enables the model to better capture user behaviors and preferences, resulting in more accurate predictions. Figure 1 illustrates the error rates of the proposed method across different datasets.

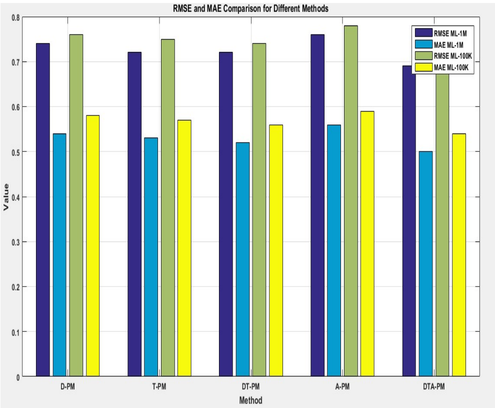

Figure 1. Comparison of the error of the proposed method on the ML-100K and ML-1M datasets

To evaluate the impact of indirect similar users identified through the target user's preferences, the (DTA-PM) method was assessed in Table 8 both with the presence of indirect similar users via the target user's preferences (N_{tar} >0) and without these indirect similar users (N_{tar} =0). The results demonstrate that incorporating these indirect similar users improves the performance of the proposed method on both the ML-100K and ML-1M datasets. According to the data in Table 8, for the ML-100K dataset, the MAE decreases from 0.652 to 0.536, and the RMSE decreases from 0.846 to 0.714, reflecting improvements of 0.116 and 0.132, respectively. Similarly, in the ML-1M dataset, the MAE is reduced from 0.629 to 0.507, and the RMSE from 0.815 to 0.693, resulting in improvements of 0.122 for both metrics. This enhancement indicates that leveraging the target user's preferences to identify indirect similar users significantly reduces prediction errors. The consistent improvements observed across both datasets highlight the robustness and effectiveness of this weighting factor within the proposed approach.

Table 8. Effect of indirect similar users via target user preferences on the error values of the proposed method

Metric	Dataset	$Ntar\neq0$	$Ntar=0$
MAE	ML100k	0.536	0.652
RMSE		0.714	0.846
MAE	ML1M	0.507	0.629
RMSE		0.693	0.815

Furthermore, in an additional experiment to examine the effect of the number of similar users on the accuracy of the method, we denote k as the number of neighboring users for the target user. Table 9 presents the error metrics obtained by considering all weighting factors (DTA-PM). Results reveal an increasing trend in both MAE and RMSE values as the number of similar users grows across both datasets. The performance on the ML-1M dataset is notably better and more stable compared to ML-100K. The best MAE and RMSE values are achieved when only the top 10 similar users are used for rating prediction. This trend can be attributed to the inclusion of users with lower similarity to the target user as k increases, indicating that the method effectively identifies the most relevant similar users. In other words, as k increases, users who are less similar to the target user are also incorporated into the calculations, leading to a slight decrease in prediction accuracy. However, this decline is minimal, supporting the conclusion that the method overall successfully selects appropriate direct and indirect similar users. Figure 2 illustrates line plots of MAE and RMSE values for varying k on both the ML-100K and ML-1M datasets.

Table 9. Effect of the number of similar users

Dataset	Metric	$k = 40$	$k = 30$	$k = 20$	$k = 10$
ML100k	MAE	0.597	0.592	0.586	0.581
	RMSE	0.751	0.749	0.733	0.728
ML 1M	MAE	0.578	0.573	0.571	0.565
	RMSE	0.728	0.72	0.718	0.714

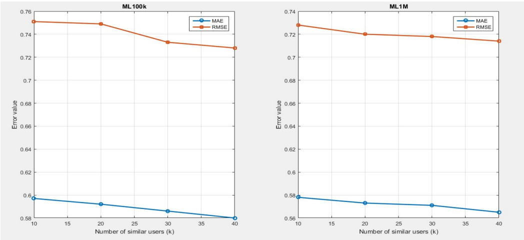

Figure 2. Effect of the number of similar users

Here, the performance of the proposed method is compared to various well-established and recent approaches. The proposed method is evaluated against a diverse range of similarity-based methods, matrix factorization techniques, model-based methods, and graph-based approaches for rating prediction and recommendation. Specifically, comparisons are made to PMF [30], BPMF [31], NNMF [32], SVD++ [33], UBBCF [34], WWBR [5], WABR [5], RMIF [34], GCF-HCR [13], ONCF [38], MLP [37], NCF

[37], SmoothRecNet [39], Alt.optim [39], RippleNet [36], KGAT [36], and KGCN [36]. In these comparisons, the metrics MAE and RMSE are employed to assess the prediction accuracy on the ML-100K and ML-1M datasets. The error comparison results of the proposed method against other approaches are summarized in Table 10, where the best values in each column are highlighted in bold and the second-best values are underlined. As observed, the proposed method achieves the lowest prediction errors. Other strong results primarily come from similarity-based approaches, indicating that contemporary similarity-based methods can deliver competitive rating prediction performance. On average, the proposed method outperforms GCF-HCR by approximately 19.7%. Conversely, the WABR method exhibits weaker performance, as it considers only users' popular items for similarity calculation. By effectively leveraging dominant and precise patterns discovered in the data, the proposed method accurately identifies and selects both direct and indirect similar users. Consequently, the ability to detect similar users, whether direct or indirect, enables the estimation of a greater number of unknown ratings. Moreover, the incorporation of a controlling coefficient to moderate the influence of indirect similar users reduces the impact of unreliable users in the rating computation, ultimately enhancing prediction accuracy. Figure 3 illustrates the comparative error charts of the proposed method versus other approaches.

Table 11 compares the performance of various methods on the ML-1M and ML-100k datasets using two evaluation metrics: NDCG@5 and Recall@5. These metrics reflect the quality of recommendations among the top 5 suggested movies for users. According to the data in Table 11, the proposed model demonstrates remarkable performance with an NDCG@5 score of 0.959 and a Recall@5 score of 0.768, outperforming all other models. This performance highlights the method's strong predictive coverage and accuracy. Based on the table, SmoothRecNet achieves the second-best performance. Conversely, the PMF model records the lowest NDCG@5 and Recall@5 scores among all methods, indicating weaker predictive coverage and accuracy compared to others. On the ML-100k dataset, the proposed method again attains the best performance, while BPMF and NNMF show strong Recall@5 results but fall slightly behind the proposed model in terms of NDCG@5. Models such as PMF and ONCF yield the weakest results across both datasets. Figure 4 further facilitates an intuitive comparison of these methods using the Recall and NDCG metrics.

Table 10. Error of the methods

Category of Method	Methods	ML100k		ML 1M	
		MAE	RMSE	MAE	RMSE
Factorization Based	PMF	0.747	0.96	0.687	0.876
	BPMF	0.726	0.92	0.67	0.852
	NNMF	0.743	0.94	0.675	0.858
	SVD++	0.715	0.909	0.667	0.853
Similarity Based	UBBCF	0.775	0.984	–	–
	WWBR	0.751	0.996	–	–
	WABR	0.811	0.998	0.782	0.961
	RMIF	0.703	0.898	0.673	0.851
	GCF-HCR	0. 688	0.895	0.619	0.849
Model-base	ONCF	0.754	0.949	0.686	0.879
	MLP	0.731	0.929	0.69	0.878
	NCF	0.729	0.925	0.685	0.87
	SmoothRecNet	0.723	0.914	0.675	0.858
	Alt. optim	0.73	0.92	0.679	0.863
Graph-based Models	RippleNet	0.714	0.914	0.733	0.937
	KGAT	0.713	0.908	0.732	0.938
	KGCN	0.711	0.908	0.731	0.937
Proposed Method (DTA-PM)		0.536	0.714	0.507	0.693

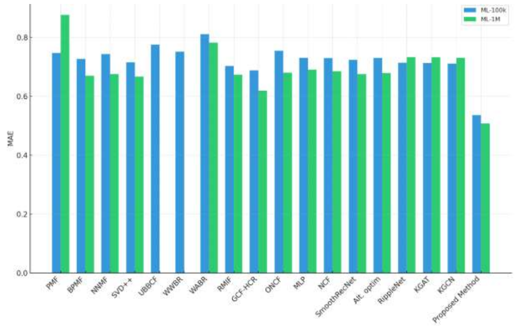

Figure 3. Error comparison of the methods

Table 11. Comparison of Recall@5 and NDCG@5 between the proposed method and other methods

Methods	ML100k		ML 1M	
	Recall@5	NDCG@5	Recall@5	NDCG@5
PMF	0.625	0.858	0.692	0.892
BPMF	0.649	0.873	0.716	0.901
NNMF	0.62	0.865	0.708	0.901
ONCF	0.634	0.855	0.703	0.898
MLP	0.643	0.871	0.693	0.894
NCF	0.648	0.871	0.695	0.895
SmoothRecNet	0.656	0.875	0.71	0.9
Alt.optim	0.648	0.872	0.703	0.899
DTA-PM	0.691	0.914	0.768	0.959

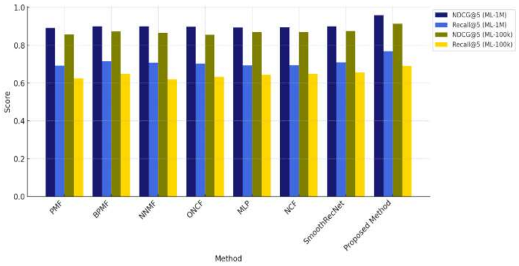

Figure 4. Comparison chart of Recall and NDCG metrics

We employ a paired t-test to confirm the performance improvement of the proposed method. First, we compute the difference for each fold, and then obtain the sample mean and sample standard deviation of these differences using formulas 20 and 21, respectively. Finally, we calculate the paired t-test statistic using formula 22:

$$\bar{d} = \frac{1}{k}\sum_{i=1}^{k} d_i \tag{20}$$

$$s_d = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(d_i - \bar{d})^2} \tag{21}$$

The paired t-statistic is then calculated as

$$t = \frac{\bar{d}}{s_d / \sqrt{k}} \tag{22}$$

In this 5-fold cross-validation framework, the test statistic follows a t-distribution with k−1=4 degrees of freedom. For each comparison, a two-tailed paired t-test is performed under the null hypothesis H_0:$\bar{d}$=0 (no difference between methods) against the alternative hypothesis H_1: $\bar{d}$≠0 (a non-zero mean difference). The corresponding p-value is obtained from the t-distribution with 4 degrees of freedom. If p<0.05, H0 is rejected and statistically significant difference at the 5% level is concluded; but if p<0.01, it is highly significant. Based on Tables 11 and 12, this procedure systematically is applied to (i) compare the full DTA-PM model with its ablated variants (D-PM, T-PM, DT-PM, A-PM) on both ML-100K and ML-1M for MAE and RMSE, and (ii) compare DTA-PM with the strongest baselines (e.g., GCF-HCR and SmoothRecNet) on both datasets. Across all key comparisons where claimed an improvement, the calculated t-statistics are large in magnitude and the corresponding p-values are below 0.05 (and most below 0.01), indicating that the observed gains of DTA-PM are not due to random variation across folds. As an illustrative example, when comparing DTA-PM against DT-PM on ML-1M for MAE, a negative $\bar{d}$ (reflecting the lower error of DTA-PM), a small s_d, and a resulting (|t|) value greater than the critical value $t_{0.05,4}$, leading to p<0.01 are obtained. Thus, a highly significant improvement is achieved. The same pattern holds for RMSE on ML-1M and for both MAE and RMSE on ML-100K, as well as for comparisons between DTA-PM and the best competing baseline methods. This rigorous paired t-test analysis, based on fold-wise results and explicitly defined formulas and hypotheses, confirms that the performance improvements reported for the proposed method are statistically significant and robust. For example, Table 12 presents the paired t-test statistics and p-values for three comparisons on the MovieLens 1M dataset.

Table 12. Paired t-test results in ML 1M

Metric	Comparison	t-value	p-value
MAE	GCF-HCR vs. DTA-PM	6.27	0.0030
MAE	SmoothRecNet vs. DTA-PM	7.84	0.0014
MAE	D-PM vs. DTA-PM	5.47	0.0050

4.3. *Parameter Analysis*

The optimal values of the parameters used in the proposed method, empirically determined to find the best results, are examined in Table 16. To find these optimal values, the error results of the method were evaluated using various parameter values. For example, the value of λ, which indicates the threshold for detecting dominant versus non-dominant sets, is evaluated in Table 13. Various values of the parameter σ, which indicates the threshold of user affiliation to a pattern, are discussed in Table 14. Additionally, Table 15 presents the results of the proposed method for different values of α, β, and θ, which represent the control coefficients for the weights of time, delta, and age, respectively.

Table 13. Results of the proposed method for different values of the parameter λ

Dataset	Metric	λ=0.2	λ=0.3	λ=0.4	λ=0.5	λ=0.6
MAE	ML100k	0.551	0.547	0.536	0.549	0.562
	ML1M	0.523	0.516	0.507	0.511	0.536
RMSE	ML100k	0.744	0.739	0.714	0.728	0.749
	ML1M	0.722	0.704	0.693	0.698	0.735

Table 14. Results of the proposed method for different values of the parameter σ

Dataset	Metric	σ=0.5	σ=0.6	σ=0.7	σ=0.8	σ=0.9
MAE	ML100k	0.577	0.561	0.549	0.536	0.54
	ML1M	0.562	0.537	0.523	0.507	0.518
RMSE	ML100k	0.766	0.741	0.73	0.714	0.725
	ML1M	0.748	0.726	0.711	0.693	0.703

Table 15. Results of the proposed method for different values of the parameters α, β, and θ

Metric	Parameter	Dataset	value
MAE	α=0.4 β=0.5 θ=0.1	ML 100K ML 1M	0.548 0.519
MAE	α=0.5 β=0.3 θ=0.2	ML 100K ML 1M	0.536 0.507
MAE	α=0.6 β=0.2 θ=0.4	ML 100K ML 1M	0.554 0.527

Table 16. Optimal parameters used in the proposed method

Parameter	Value
λ	0.4
σ	0.8
α	0.5
β	0.3
θ	0.2
Ω	0.01

5. Conclusion

This paper proposes a novel approach to enhancing the performance of movie recommender systems by leveraging multidimensional analyses and integrating dynamic weighting factors, including delta, temporal, and genre-age compatibility, to effectively address major challenges such as data sparsity, cold-start problems, temporal dynamics, and scalability. Experimental results demonstrate that the proposed model achieves significantly lower MAE and RMSE values on the ML-100k and ML-1M datasets, outperforming traditional methods such as PMF, BPMF, NCF, as well as other recent approaches. By combining influential factors related to user preferences when scoring an item, this approach enables more accurate prediction of user tastes and facilitates personalized recommendations.

Moreover, the incorporation of reference users and analysis of dominant behavioral patterns mitigates the effects of sparse data and improves the identification of indirect similar users. This enhances the flexibility of rating predictions for items not yet rated by users similar to the target user, thereby expanding the coverage of the recommendation scope. A distinctive feature of this method lies in its adaptability to integrate multiple factors while accounting for temporal variations and evolving user needs. Beyond reducing prediction errors, the proposed framework is extensible to other domains such as e-commerce and social networks.

Evaluation results reveal that the proposed method exhibits superior performance compared to a wide range of baseline and state-of-the-art techniques in both prediction accuracy and recommendation quality. It consistently achieves the lowest MAE and RMSE values across both ML-100k and ML-1M datasets. Furthermore, regarding ranking metrics NDCG@5 and Recall@5, which assess the quality and coverage of the top-5 item predictions, the proposed model attains the highest scores of 0.959 and 0.768, respectively, outperforming all compared methods. This advantage is especially pronounced against models like GCF-HCR and SmoothRecNet, which despite their competitive performance, lag behind the proposed approach. These results underscore the effectiveness of integrating multidimensional analysis and leveraging dominant user behavior patterns in enhancing both accuracy and coverage of recommendations.

Additionally, controlling the influence of indirect users and reducing the impact of unreliable users contribute to improved rating prediction precision. Overall, the findings underscore the robustness and flexibility of the proposed approach in accurately identifying user preferences and delivering precise, personalized recommendations, making it a promising solution for deployment in diverse application domains.

For future work, the performance could be further enhanced by developing an intelligent deep framework utilizing deep recurrent neural networks, which can better model complex and dynamic user-item interactions and improve sequential decision-making in recommendations. Furthermore, adopting dynamic models such as tensor-based approaches that incorporate time-varying weighting factors could also improve prediction quality and is a promising direction for extending this research.

Authors' Contributions

All authors equally contributed to this study.

Declaration

None.

Transparency Statement

None.

Acknowledgments

None.

Declaration of Interest

The authors declare that they have no conflict of interest. The authors also declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The study was conducted as part of a Ph.D. dissertation project.

Ethical Considerations

Not applicable.

References

[1] J. Xu, Z. Chen, and S. Yang, "A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions," 2025, doi: 10.1109/TMM.2026.3668620.

[2] B. Kumar and N. Sharma, "Approaches, Issues and Challenges in Recommender Systems: A Systematic Review," *Indian Journal of Science and Technology,* vol. 9, no. 47, pp. 1-12, 2016, doi: 10.17485/ijst/2015/v8i1/94892.

[3] S. Sohail and J. Siddiqui, "Classifications of Recommender Systems: A review," *Engineering Science and Technology Review,* pp. 132-153, 2017. [Online]. Available: https://www.academia.edu/download/68334157/Classifications_of_Recommender_Systems.pdf.

[4] K. Zołna and B. Romanski, "User Modeling Using LSTM Networks," 2017, pp. 5025-5029, doi: 10.1609/aaai.v31i1.11068.

[5] M. Ramezani, P. Moradi, and F. Akhlaghian, "A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains," *Statistical Mechanics and its Applications,* vol. 408, pp. 72-84, 2014, doi: 10.1016/j.physa.2014.04.002.

[6] Y. Li, "Deep Reinforcement Learning: An Overview," 2018. [Online]. Available: https://arxiv.org/abs/1701.07274

[7] Y. Hu, F. Xiong, and L. Dongyuan, "Movie collaborative filtering with multiplex implicit feedbacks," *Neurocomputing,* vol. 398, pp. 485-494, 2020, doi: 10.1016/j.neucom.2019.03.098.

[8] P. Singh and K. Sinha, "An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user-user similarity as weight," *Knowl Inf Syst,* vol. 64, pp. 665-701, 2022, doi: 10.1007/s10115-021-01651-8.

[9] Y. Hu and F. Xiong, "Movie collaborative filtering with multiplex implicit feedbacks," *Neurocomputing,* vol. 398, pp. 485-494, 2020, doi: 10.1016/j.neucom.2019.03.098.

[10] S. K. Kumar, S. Bag, and M. K. Tiwari, "An efficient recommendation generation using relevant Jaccard similarity," *Information Sciences,* pp. 53-64, 2019, doi: 10.1016/j.ins.2019.01.023.

[11] K. Hayashi, "Rethinking Correlation-based Item-Item Similarities for Recommender Systems," 2022, pp. 2287-2291, doi: 10.1145/3477495.3532055.

[12] Y. Tang, K. Guo, and R. Zhang, "An effective incremental collaborative filtering based recommendation architecture for personalized web-sites," *World Wide Web,* vol. 23, pp. 1319-1340, 2020, doi: 10.1007/s11280-019-00693-x.

[13] M. Ramezani and F. Akhlaghian Tab, "A new generalized collaborative filtering approach on sparse data by extracting high confidence relations between users," *Information Sciences,* vol. 570, pp. 323-341, 2021, doi: 10.1016/j.ins.2021.04.025.

[14] M. Yang and K. Sinaga, "Unsupervised K-Means Clustering Algorithm," pp. 80716-80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[15] Y. Feng and L. Wang, "Distributed ItemCF Recommendation Algorithm Based on the Combination of MapReduce and Hive," *Electronics,* pp. 1-20, 2023, doi: 10.3390/electronics12163398.

[16] Y. Zhang, X. Xin, and X. He, "Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation," 2019, pp. 125-134, doi: 10.1145/3331184.3331188.

[17] T. Vander Aa and I. Chakroun, "Distributed Bayesian Probabilistic Matrix Factorization," *Procedia Computer*

Science, vol. 108, pp. 1030-1039, 2017, doi: 10.1016/j.procs.2017.05.009.

[18] J. Chen and J. Fang, "clMF: A fine-grained and portable alternating least squares algorithm for parallel matrix factorization," *Future Generation Computer Systems,* vol. 108, pp. 1192-1205, 2020, doi: 10.1016/j.future.2018.04.071.

[19] J. Deng, H. Li, and Q. Zhang, "Biased Multiobjective Optimization and Decomposition Algorithm," *IEEE Transactions on Cybernetics,* vol. 47, pp. 52-66, 2017, doi: 10.1109/TCYB.2015.2507366.

[20] K. Li and X. Zhou, "Sparse online collaborative filtering with dynamic regularization," *Inf. Sci.,* vol. 505, pp. 535-548, 2019, doi: 10.1016/j.ins.2019.07.093.

[21] L. Berkani and S. Zeghoud, "Neural hybrid recommendation based on GMF and hebrid MLP
Artificial intelligence and machine learning for EDGE computing," 2022, pp. 287-303.

[22] Z. Liu and X. Feng, "Self-paced learning enhanced neural matrix factorization for noise-aware recommendation," *Knowledge-Based Systems,* vol. 213, 2021, doi: 10.1016/j.knosys.2020.106660.

[23] H. Wang, J. Wang, and M. Zhao, "Graphgan, Graph representation learning with generative adversarial nets," 2017, doi: 10.1609/aaai.v32i1.11872.

[24] X. Guo and W. Lin, "DKEN: Deep knowledge-enhanced network for recommender systems," vol. 540, pp. 263-277, 2020, doi: 10.1016/j.ins.2020.06.041.

[25] J. Wang and Y. Lantao Yu, "A minimax game for unifying generative and discriminative information retrieval models," 2017, pp. 515-524, doi: 10.1145/3077136.3080786.

[26] Y. Kilani and B. Alhijawi, "A collaborative filtering recommender system using genetic algorithm," *Inf. Process. Manage,* vol. 57, no. 6, p. 102310, 2020, doi: 10.1016/j.ipm.2020.102310.

[27] S. Ahmadian and N. Joorabloo, "Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach," *Expert Systems with Applications,* p. 115849, 2022, doi: 10.1016/j.eswa.2021.115849.

[28] G. Harshvardhan and M. K. Gourisaria, "nsupervised Boltzmann machine-based time-aware recommendation system," *Computer and Information Sciences,* vol. 34, pp. 6400-6413, 2022, doi: 10.1016/j.jksuci.2021.01.017.

[29] G. Jain and T. Mahara, "Performance Evaluation of Time-based Recommendation System in Collaborative Filtering Technique," *Procedia Computer Science,* vol. 18, pp. 1834-1844, 2023, doi: 10.1016/j.procs.2023.01.161.

[30] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," 2008, pp. 1257-1264. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2007/hash/d7322ed717dedf1eb4e6e52a37ea7bcd-Abstract.html.

[31] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," 2008, pp. 880-887, doi: 10.1145/1390156.1390267.

[32] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," 2008, pp. 426-434, doi: 10.1145/1401890.1401944.

[33] D. M. Roy and G. K. Dziugaite, "Neural network matrix factorization," 2015. [Online]. Available: https://arxiv.org/abs/1511.06443

[34] B. K. Patra and R. Launonen, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," *Knowl.-Based Syst,* vol. 82, pp. 163-177, 2015, doi: 10.1016/j.knosys.2015.03.001.

[35] H. Wang and M. Zhao, "Knowledge Graph Convolutional Networks for Recommender Systems," 2019, pp. 3307-3313, doi: 10.1145/3308558.3313417.

[36] X. Wang, X. He, and Y. Cao, "Kgat: Knowledge graph attention network for recommendation," 2019, pp. 950-958, doi: 10.1145/3292500.3330989.

[37] X. He and X. Du, "Outer product-based neural collaborative filtering," 2018, pp. 2227-2233, doi: 10.24963/ijcai.2018/308.

[38] X. He and L. Liao, "Neural collaborative filtering," 2017, pp. 173-182, doi: 10.1145/3038912.3052569.

[39] A. Shirali, R. Kazemi, and A. Amini, "Collaborative filtering with representation learning in the frequency domain," *Information Sciences,* 2024, doi: 10.1016/j.ins.2024.121240.