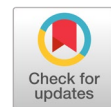


Optimization hybrid weighted switching filtering (OHWSF) using SVD and SVD++ for addressing data sparsity



Malim Muhammad ^{a,b,1,*}, Gunardi ^{b,2}, Danardono ^{b,3}, Dedi Rosadi ^{a,c,4}

^a Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Gadjah Mada, 55281, Indonesia

^b Department of Mathematics Education, Faculty of Teacher Training and Education, Universitas Muhammadiyah Purwokerto, 53182, Indonesia

^c Statistics RnD, Yogyakarta, Indonesia

¹ malimmuhammad1986@mail.ugm.ac.id; ² gunardi@ugm.ac.id; ³ danardono@ugm.ac.id; ⁴ dedirosadistat91@gmail.com

* corresponding author

ARTICLE INFO

Article history

Received October 11, 2024

Revised July 7, 2025

Accepted July 30, 2025

Available online August 31, 2025

Keywords

Recommender systems

Data sparsity

OHWSF

Hybrid weighted filtering

MAE improvement

ABSTRACT

Recommender systems are crucial for filtering vast amounts of digital content and providing personalized recommendations; however, their effectiveness is often hindered by data sparsity, where limited user-item interactions lead to reduced prediction accuracy. This study introduces a novel hybrid model, Optimization Hybrid Weighted Switching Filtering (OHWSF), to overcome this challenge by integrating two complementary strategies: Hybrid Weighted Filtering (HWF), which linearly combines predictions from SVD and SVD++ using a weighting parameter (α), and Hybrid Switching Filtering (HSF), which dynamically selects predictions based on a threshold rating (θ). The OHWSF framework introduces a tunable optimization mechanism governed by the parameter σ_1 to adaptively balance weighting and switching decisions based on actual rating deviations. Unlike existing static or manually tuned hybrid methods, the proposed model combines dynamic switching with weight optimization to minimize prediction error effectively. Extensive experiments on four benchmark datasets (ML-100K, ML-1M, Amazon Cell Phones Reviews, and GoodBooks-10K) demonstrate that OHWSF consistently outperforms traditional collaborative filtering (UBCF, IBCF), matrix factorization techniques (SVD, SVD++), and standalone hybrid models across all evaluation metrics (MAE, MSE, RMSE). The model achieves optimal performance within the range of $\alpha = 0.6$ – 0.9 and $\theta = 1.0$ – 1.5 , demonstrating robustness across varying sparsity levels. Notably, OHWSF achieves up to 742.16% MAE improvement over the UBCF model, with significantly reduced training time compared to SVD++. These findings confirm that OHWSF significantly improves prediction accuracy, scalability, and adaptability in sparse data environments. This research contributes a flexible, interpretable, and efficient hybrid recommendation framework suitable for real-world applications.



© 2025 The Author(s).

This is an open access article under the CC-BY-SA license.



1. Introduction

In the digital era, users are constantly exposed to an overwhelming abundance of information, giving rise to a condition known as information overload [1]. While such access is generally advantageous, it creates difficulties in identifying content that aligns with individual preferences, often resulting in cognitive fatigue, time inefficiencies, and diminished user satisfaction [2]. To address this challenge, Recommender Systems (RSs) have emerged as critical tools that leverage user behavior and preferences

to provide personalized content suggestions [3], [4]. These systems are now deeply embedded across diverse application domains, including movies [5], music [6], online shopping [7], food [8], tourism [9], books [10], news [11], healthcare systems [12], social networks [13], and online education platforms [14].

Among the foundational techniques in RSs, Collaborative Filtering (CF) remains the most widely adopted approach due to its capacity to uncover latent patterns in user-item interactions without requiring explicit domain knowledge [15]. CF techniques are broadly classified into neighborhood-based and model-based methods [16], [17]. The former exploits user-user or item-item similarities, such as user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF), as well as latent feature models, including Singular Value Decomposition (SVD) and SVD++, which capture user and item characteristics. However, both variants suffer critically from the data sparsity problem, which occurs when users rate only a limited subset of items, resulting in an incomplete and sparse user-item matrix [18]–[20]. This sparsity substantially hinders the system's ability to learn meaningful representations and deliver accurate recommendations, particularly in large-scale domains like movie platforms, online retail, and digital libraries [2], [4], [21]–[24].

Several methods have been proposed to mitigate sparsity, including K-Nearest Neighbors (the UBCF and IBCF) [23], [25]–[27], matrix factorization (SVD, SVD++, and NMF) [24], [28]–[30], and Co-Clustering [31]. Nevertheless, these approaches often fall short in highly sparse conditions, mainly when they rely solely on either explicit or implicit feedback. Hybrid recommender systems offer a promising direction by integrating multiple algorithms to leverage their complementary strengths [32]–[36]. However, existing hybrid methods typically use static weight combinations or fixed decision rules, which lack adaptability to dynamic interaction patterns and fail to generalize across varying sparsity levels.

To address this gap, this study introduces the Optimization Hybrid Weighted Switching Filtering (OHWSF) model, a novel and adaptive hybrid recommendation framework that dynamically integrates Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) mechanisms. It employs a Lagrangian-based weighting mechanism for linear combination (via α in HWF) and a threshold-based decision rule (via θ in HSF) to determine when to switch models based on predicted confidence. These strategies are then fused and optimized through the OHWSF architecture using an adaptive weight parameter (σ_1), which automatically adjusts the influence of HWF and HSF according to the actual error distribution.

The main contributions of this research include:

- Introduction of OHWSF, a novel optimization-based hybrid recommendation model that adaptively integrates Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) through a dynamic weighting mechanism (σ_1), effectively balancing linear combination and threshold-based selection to minimize prediction error.
- Empirical validation of its performance across diverse datasets (ML-100K, ML-1M, Amazon Cell Phones Reviews, and GoodBooks-10K), which differ in scale and sparsity, to validate the model's generalizability across diverse recommendation scenarios.
- An ablation study examining the impact of tuning α , θ , and σ_1 and demonstrating OHWSF's superiority over traditional collaborative filtering methods (UBCF, IBCF), matrix factorization techniques (SVD, SVD++), and standalone hybrid models (HWF, HSF, OHWF) using standard evaluation metrics (MAE, MSE, RMSE) and training time, highlighting both its predictive accuracy and computational efficiency.

2. Method

2.1. k-NN collaborative filtering

K-nearest neighbors (k-NN) collaborative filtering is a memory-based collaborative filtering technique that utilizes the entire user-item interaction dataset to generate predictions. As one of the

earliest automated CF methods, k-NN uses the rating matrix to compute item similarities and recommend relevant items. There are two main types of k-NN models: user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). In this approach, the rating prediction for an item i by a user u is derived directly from an $m \times n$ user-item rating matrix R , where m represents the number of users and n represents the number of items.

2.1.1. User-based collaborative filtering (UBCF) model

User-Based Collaborative Filtering (UBCF) is a personalized recommendation technique grounded in the intuition that users with similar past behaviors are likely to share similar preferences in the future [37]. This method predicts the preferences of an active user by identifying other users referred to as neighbors who exhibit similar rating profiles. The process begins by representing each user's rating behavior as an n -dimensional rating vector. The rating vector of $\vec{r}_u = \{r_{u1}, r_{u2}, \dots, r_{un}\}$ represents the ratings assigned by a user u to a set of n items. Each element r_{ui} within the vector corresponds to the rating that user u has given to item i . By comparing these vectors, one can compute the similarity between users, typically using metrics such as Cosine Similarity (COS) or the Pearson Correlation Coefficient (PCC) [26], [27]. UBCF predicts the rating that a user u would give to an item i by aggregating the ratings of similar users who have also rated item i [38]. The basic prediction formula is given in equation (1).

$$\hat{r}_{ui}^{UBCF} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \quad (1)$$

where $\hat{r}_{ui}^{UBCF}$ is the predicted rating, $\text{sim}(u, v)$ is the similarity between users u and v , r_{vi} is the rating given by user v to item i , and $N_i^k(u)$ represents the set of k -nearest neighbors of user u who have rated item i . To address rating scale biases among users, a mean-centered version is commonly employed in equation (2).

$$\hat{r}_{ui}^{UBCF} = \bar{r}_u + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \bar{r}_v)}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \quad (2)$$

where $\bar{r}_u$ and $\bar{r}_v$ represent the average ratings of users u and v , respectively.

2.2. Item-based collaborative filtering (IBCF) model

Similarly, the item-based collaborative filtering (IBCF) model is a memory-based recommendation approach that focuses on the similarity between items rather than users. This model predicts a user's rating of a particular item by analyzing the ratings the user has given to other similar items. The fundamental principle of IBCF is that if a user has rated an item highly, they are likely to rate other similar items similarly. To achieve this, item-item similarity is calculated using measures such as the COS and PCC, with the COS being commonly favored due to its geometric interpretability and empirical performance [39]. The prediction in IBCF is calculated using a weighted sum of the user's ratings for similar items, adjusted by the mean rating of each item. The prediction formula is defined as equation (3).

$$\hat{r}_{ui}^{IBCF} = \bar{r}_i + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - \bar{r}_j)}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)} \quad (3)$$

where $\hat{r}_{ui}^{IBCF}$ represents the predicted rating for user u and item i , $\bar{r}_i$ and $\bar{r}_j$ represent the average ratings of items i and j , respectively, r_{uj} denotes the rating given by user u to item j , $\text{sim}(i, j)$ represents the similarity between items i and j , $N_u^k(i)$ denotes the set of k most similar items to item i that have been rated by user u .

2.3. Matrix Factorization (MF)

Matrix Factorization (MF) is one of the most widely adopted techniques in model-based collaborative filtering, particularly effective in handling the sparsity problem in user-item rating matrices. Matrix Factorization is a fundamental model-based collaborative filtering technique that decomposes the user-item rating matrix $R \in \mathbb{R}^{n \times m}$ into the product of two lower-dimensional latent matrices. The central assumption is that both users and items can be represented in a shared latent factor space of dimension k , where $k \ll \min(n, m)$. This dimensionality reduction captures the hidden features underlying user preferences and item characteristics, enabling the model to generalize more effectively to unseen data. Formally, the MF aims to approximate the original matrix R as in (4).

$$R \approx Q^T \cdot P \quad (4)$$

where $P \in \mathbb{R}^{k \times n}$ is the user latent matrix, each column $p_u \in \mathbb{R}^k$ representing user u , $Q \in \mathbb{R}^{k \times m}$ is the item latent matrix, each column $q_i \in \mathbb{R}^k$ representing item i , k is the number of latent factors. Therefore, the predicted rating $\hat{r}_{ui}$ is defined in equation (5).

$$\hat{r}_{ui} = q_i^T \cdot p_u \quad (5)$$

where q_i is the vector of each item i , and p_u is the vector of each user u .

2.3.1. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization technique widely used in recommender systems to reduce the dimensionality of large user-item interaction matrices. By projecting the original high-dimensional rating matrix into a lower-dimensional latent feature space, SVD effectively captures the most significant patterns in user behavior and item characteristics [40]. Given a user-item rating matrix $R \in \mathbb{R}^{n \times m}$ of size $m \times n$, where m is the number of users and n is the number of items, SVD approximates R as the product of three matrices in equation (6) [23].

$$R \approx P \cdot S \cdot Q^T \quad (6)$$

where $S \in \mathbb{R}^{k \times k}$ is a diagonal matrix of single values.

In practical implementations, particularly in collaborative filtering, SVD is often extended to incorporate bias terms that capture user and item rating tendencies. The predicted rating $\hat{r}_{ui}^{SVD}$ for user u and item i is computed in equation (7).

$$\hat{r}_{ui}^{SVD} = \mu + b_u + b_i + q_i^T \cdot p_u \quad (7)$$

where μ is the global mean rating. The parameters b_u and b_i represent the user and item biases, respectively, and are defined in equations (8) and (9).

$$b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu)}{|R(u)|} \quad (8)$$

$$b_i = \frac{\sum_{u \in R(i)} (r_{ui} - \mu)}{|R(i)|} \quad (9)$$

where $R(u)$ is the set of all items that are rated by user u and $R(i)$ is the set of all users who rated item i . If a user u is unknown (i.e., not present in the training data), then their bias term b_u and latent factor vector p_u are assumed to be zero. Similarly, for an unseen item i , both the item bias b_i and the latent factor vector q_i are also set to zero.

2.3.2. Singular Value Decomposition Plus Plus (SVD++)

SVD++ is an improved version of SVD and has been widely applied in recommender systems due to its superior prediction performance compared to SVD [41]. SVD++ is an advanced matrix factorization-based recommendation algorithm that builds upon the traditional SVD approach by integrating both

explicit user-item ratings and implicit user interactions such as views, clicks, and browsing behavior. This additional information enhances the accuracy of recommendations by considering user biases and implicit preferences. The predicted rating for user u and item i is computed using the following equation (10) [28].

$$\hat{r}_{ui}^{SVD++} = \mu + b_u + b_i + q_i^T \cdot \left(p_u + \frac{1}{\sqrt{|N(\mu)|}} \sum_{j \in N(\mu)} y_j \right) \quad (10)$$

where $N(\mu)$ is the set of items with implicit feedback from user u , y_j represents the latent factor vector for item j that captures the implicit feedback effect. The term $\frac{1}{\sqrt{|N(\mu)|}} \sum_{j \in N(\mu)} y_j$ models the aggregate effect of implicit interactions. If a user u is unknown (not present in the training set), their bias b_u and latent factor vector p_u are assumed to be zero. Similarly, for an unseen item i , the item bias b_i , the latent factor vector q_i , and the implicit feedback factor y_j are also initialized to zero. This defaulting mechanism ensures that the model provides a baseline prediction of μ when no prior information is available.

2.4. Hybrid Filtering (HWF)

Hybrid filtering techniques combine two or more distinct basic recommender techniques, such as collaborative filtering, content-based filtering, knowledge-based filtering, and demographic filtering, aiming to produce a robust system by mitigating the weaknesses of one method through the strengths of another [42]–[45]. Aggarwal [46] categorizes hybrid filtering recommendation systems into three categories: ensemble design, monolithic design, and mixed system. The ensemble design is divided into two categories, namely parallel and sequential. Parallel methods include Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF), while sequential methods include Hybrid Feature Augmentation Filtering (HFAF) and Hybrid Cascade Filtering (HCF). Moreover, monolithic designs include Hybrid Feature Combination Filtering (HFCF) and Hybrid Meta-Level Filtering (HMLF). The last category is mixed, providing Hybrid Mixed Filtering (HMF).

2.4.1. Hybrid Weighted Filtering (HWF) Model

The Hybrid Weighted Filtering (HWF) model is a technique that combines the outputs of multiple recommendation algorithms to enhance predictive accuracy by leveraging their complementary strengths. Rather than relying on a single model, HWF computes a weighted average of predictions from two or more algorithms, balancing their contributions based on their individual performance or domain relevance. In this case, the HWF model combines the predictions of the best two models. The hybrid prediction $\hat{r}_{ui}^{HWF}$ for user u and item i is given by equation (11) [44].

$$\hat{r}_{ui}^{HWF} = \alpha \cdot \hat{r}_{ui}^1 + (1 - \alpha) \cdot \hat{r}_{ui}^2 \quad (11)$$

where $\hat{r}_{ui}^{HWF}$ is the predicted rating using the HWF model, $\hat{r}_{ui}^1$ is the predicted rating using the first model, $\hat{r}_{ui}^2$ is the predicted rating using the second model, $\alpha \in [0, 1]$ is the weight for the first model predictions, and $(1 - \alpha)$ is the weight assigned to the second model predictions.

2.4.2. Hybrid Switching Filtering (HSF) Model

The Hybrid Filter Switching Model (HSF) dynamically switches between different recommendation algorithms based on specific criteria. The idea is to select the most suitable algorithm based on the context, such as the number of user interactions, the availability of implicit feedback, or other factors that influence prediction accuracy [47]. Unlike HWF, which blends model outputs, HSF switches entirely between models based on a predetermined rating threshold θ , which determines when to use the first or second model based on the rating threshold. The formula can be expressed as equation (12).

$$\hat{r}_{ui}^{HSF} = \begin{cases} \hat{r}_{ui}^2, & \text{if } \hat{r}_{ui}^2 > \theta \\ \hat{r}_{ui}^1, & \text{if } \hat{r}_{ui}^1 \leq \theta \end{cases} \quad (12)$$

where θ is the rating threshold for switching between first and second model predictions.

2.5. Proposed Model

The proposed model, Optimized Hybrid Weighted Filtering (OHWF), is designed to enhance the accuracy and relevance of recommendation items to users by intelligently combining the strengths of multiple collaborative filtering (CF) algorithms. Unlike traditional hybrid models that use fixed or manually assigned weights, OHWF incorporates an optimization mechanism to dynamically determine the optimal contribution of each algorithm in the ensemble. This model enables the model to adapt its weighting scheme in a data-driven manner, thereby minimizing prediction errors across users and items.

Algorithm 1 Optimization Hybrid Weighted Filtering (OHWF)

Require:

- User-item interaction matrix R for m users and n items.
- Algorithms A_1 (first model) and A_2 (second model) with prediction outputs $\hat{r}_{ui}^{A_1}$ and $\hat{r}_{ui}^{A_2}$, respectively.
- Threshold θ for switching between the first model and the second model
- Initial weight optimization parameter σ_1, σ_2

Objective:

- Final prediction $\hat{r}_{ui}^{OHWF}$ and $\hat{r}_{ui}^{OHWSF}$, optimized weights σ_1, σ_2 , and Top-N recommendations

Step 1: Split the user-item matrix R into training (80%) and testing (20%) datasets:

- $Trainset, testset \leftarrow split(R)$

Step 2: Train the state-of-the-art models (UBCF, IBCF) and baseline models (SVD, and SVD++):

- $A_1 \leftarrow train(UBCF, trainset)$
- $A_2 \leftarrow train(BCF, trainset)$
- $A_3 \leftarrow train(SVD, trainset)$
- $A_4 \leftarrow train(SVD + +, trainset)$

Step 3: For each user-item pair (u, i) in the test set, calculate the prediction ratings:

- $\hat{r}_{ui}^{A_1} \leftarrow A_1(u, i)$
- $\hat{r}_{ui}^{A_2} \leftarrow A_2(u, i)$
- $\hat{r}_{ui}^{A_3} \leftarrow A_3(u, i)$
- $\hat{r}_{ui}^{A_4} \leftarrow A_4(u, i)$

Step 4: Implement the hybrid weighted filtering (HWF) model using SVD++ and SVD based on a predefined $\alpha = 0.5$.

- $\hat{r}_{ui}^{HWF} = \alpha \cdot \hat{r}_{ui}^{SVD++} + (1 - \alpha) \cdot \hat{r}_{ui}^{SVD}$

Step 5: Implement the hybrid switching filtering (HSF) model using SVD++ and SVD based on a predefined rating threshold $\theta = 3.0$.

- $\hat{r}_{ui}^{HSF} = \begin{cases} \hat{r}_{ui}^{SVD++}, & \text{if } \hat{r}_{ui}^{SVD++} > \theta \\ \hat{r}_{ui}^{SVD}, & \text{if } \hat{r}_{ui}^{SVD++} \leq \theta \end{cases}$

Step 6: Calculate evaluation metrics such as MAE, MSE, and RMSE to assess the best model performance:

- $MAE \leftarrow \frac{\sum_{(u,i) \in W} |r_{ui} - \hat{r}_{ui}|}{|W|}$
- $MSE \leftarrow \frac{\sum_{(u,i) \in W} (r_{ui} - \hat{r}_{ui})^2}{|W|}$
- $RMSE \leftarrow \sqrt{\frac{\sum_{(u,i) \in W} (r_{ui} - \hat{r}_{ui})^2}{|W|}}$, where W denotes the testset

Step 7: Optimize the weights σ_1 and σ_2 to minimize the error between the actual and predicted ratings:

- $\sigma_1 = \underset{\sigma_1}{\operatorname{argmin}} \sum_{u,i} (r_{ui} - \hat{r}_{ui}^{SVD})^2$ and $\sigma_2 = \underset{\sigma_2}{\operatorname{argmin}} \sum_{u,i} (r_{ui} - \hat{r}_{ui}^{SVD++})^2$
- $\sigma_1 = \underset{\sigma_1}{\operatorname{argmin}} \sum_{u,i} (r_{ui} - \hat{r}_{ui}^{HWF})^2$ and $\sigma_2 = \underset{\sigma_2}{\operatorname{argmin}} \sum_{u,i} (r_{ui} - \hat{r}_{ui}^{HSF})^2$

Step 8: Combine the predictions from both algorithms using the optimized hybrid weighted filtering.

- $\hat{r}_{ui}^{OHWF} = \sigma_1 \cdot \hat{r}_{ui}^{SVD++} + (1 - \sigma_2) \cdot \hat{r}_{ui}^{SVD}$
- $\hat{r}_{ui}^{OHWSF} = \sigma_1 \cdot \hat{r}_{ui}^{HWF} + (1 - \sigma_2) \cdot \hat{r}_{ui}^{HSF}$

Step 9: Iterate over all user-item pairs and calculate the final predictions for the test set:

- $\hat{r}_{ui} = \hat{r}_{ui}^{OHWF}$
- $\hat{r}_{ui} = \hat{r}_{ui}^{OHWSF}$

Step 10: Calculate evaluation metrics in step 6

Step 11: Return the optimized weights (σ_1 and σ_2), the final predictions of $\hat{r}_{ui}^{OHWF}$ and $\hat{r}_{ui}^{OHWSF}$ for each user-item pair.

The foundation of this model lies in the integration of multiple prediction models, such as SVD, SVD++, and others. The final predicted rating $\hat{r}_{ui}^{OHWF}$ for a given user u and item i is computed as a weighted sum of the predictions from each contributing algorithm, as in equation (13).

$$\hat{r}_{ui}^{OHWF} = \sum_{f=1}^c \sigma_f r_{ui}^{(f)} \quad (13)$$

where $r_{ui}^{(f)}$ is the predicted rating for user u and item i produced by the f^{th} recommendation algorithm, σ_f represents the weight of the f^{th} algorithm f , and c is the total number of algorithms included in the hybrid model.

The central objective of OHWF is to minimize the total squared prediction error between the actual user ratings, r_{ui} , and the predicted ratings, $\hat{r}_{ui}^{OHWF}$. This formula is expressed as the following optimization problem in equation (14).

$$E(\sigma) = \sum_{u,i} (r_{ui} - \hat{r}_{ui}^{OHWF})^2 \quad (14)$$

subject to the normalization constraint in equation (15):

$$\sum_{f=1}^c \sigma_f = 1 \quad (15)$$

This constraint ensures that the algorithm weights σ_f form a convex combination, maintaining a balanced and interpretable weighting scheme. The optimization can be efficiently solved using Lagrange multipliers, ensuring that the best-performing model combinations receive higher weight while still contributing to a unified ensemble prediction. The target function $E(\sigma)$ is the sum of squared errors between actual and predicted ratings across all user-item pairs in (16).

$$E(\sigma) = \sum_{u=1}^m \sum_{i=1}^n (r_{ui} - \sum_{f=1}^c \sigma_f r_{ui}^{(f)})^2 \quad (16)$$

to solve the optimization problem while respecting the constraint, we use a Lagrange multiplier method. The Lagrange function is constructed in (17).

$$L(\sigma, \lambda) = \frac{1}{2} E(\sigma) + \lambda (\sum_{f=1}^c \sigma_f - 1) \quad (17)$$

where λ is the Lagrange multiplier associated with the constraint.

To find the optimal σ_f , we compute the partial derivatives of $L(\sigma, \lambda)$ with respect to each σ_f and set them to zero. This formula gives us the necessary condition for optimization. The partial derivative of σ_f is $\frac{\partial L}{\partial \sigma_f} = 0$. Taking the derivative of the Lagrange function (equation 18), we get.

$$\begin{aligned} L(\sigma, \lambda) &= \frac{1}{2} \left[\sum_{u=1}^m \sum_{i=1}^n (r_{ui} - \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)})^2 \right] + \lambda (\sum_{f=1}^c \sigma_f - 1) \\ \frac{\partial L}{\partial \sigma_f} &= \frac{1}{2} (2) \left[\sum_{u=1}^m \sum_{i=1}^n (r_{ui} - \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)}) \right] + \frac{\partial \left[\sum_{u=1}^m \sum_{i=1}^n (r_{ui} - \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)}) \right]}{\partial \sigma_f} + \lambda = 0 \\ &\Leftrightarrow \sum_{u=1}^m \sum_{i=1}^n (r_{ui} - \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)}) (-r_{ui}^{(f)}) + \lambda = 0 \\ &\Leftrightarrow -\sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(f)} + \sum_{u=1}^m \sum_{i=1}^n \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)} r_{ui}^{(f)} + \lambda = 0 \\ &\Leftrightarrow \sum_{u=1}^m \sum_{i=1}^n \sum_{f=1}^c \sigma_f \cdot r_{ui}^{(f)} r_{ui}^{(f)} + \lambda = \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(f)} \\ \sum_{u=1}^m \sum_{i=1}^n (\sigma_1 r_{ui}^{(1)} + \sigma_2 r_{ui}^{(2)} + \dots + \sigma_c r_{ui}^{(c)}) r_{ui}^{(f)} + \lambda &= \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(f)} \end{aligned} \quad (18)$$

The left-hand side represents the weighted sum of ratings predicted by the c -th algorithm (including the Lagrange multiplier term). The right-hand side represents the sum of the actual ratings weighted by the corresponding predicted ratings. Next, we convert equation (18) into matrix form $A \cdot X = B$. Thus, the system of equations (19) is used to solve for the optimal weights $\sigma_1, \sigma_2, \dots, \sigma_c$.

$$\begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(c)} & 1 \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(c)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(c)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(c)} r_{ui}^{(c)} & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_c \\ \lambda \end{bmatrix} = \begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(1)} \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(2)} \\ \vdots \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(c)} \end{bmatrix} \quad (19)$$

The system of linear equations can be solved using matrix inversion $X = A^{-1} \cdot B$. The solution for X , which contains the optimal weights $\sigma_1, \sigma_2, \dots, \sigma_c$ in equation (20).

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_c \\ \lambda \end{bmatrix} = \begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(c)} & 1 \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(c)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(c)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(c)} r_{ui}^{(c)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(1)} \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(2)} \\ \vdots \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(c)} \end{bmatrix} \quad (20)$$

Equation (20) provides the optimal values for the weights $\sigma_1, \sigma_2, \dots, \sigma_c$ and the Lagrange multiplier λ . Once we have these optimal values, we can compute the final weighted predictions for the recommender system. Suppose we have two recommendation algorithms (SVD and SVD++); the final prediction $\hat{r}_{ui}^{OHWF}$ for user u and item i is a weighted sum of the predictions from these two algorithms, as shown in equation (21).

$$\hat{r}_{ui}^{OHWF} = \sigma_1 r_{ui}^{(1)} + \sigma_2 r_{ui}^{(2)} \quad (21)$$

The key constraint is that the sum of the weights must equal 1, with $\sigma_1 + \sigma_2 = 1$. Our goal is to compute the values of σ_1 and σ_2 that minimize the prediction error. To compute the weights, we solve the following matrix equation, which is derived from the optimization process in equation (22).

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(2)} & 1 \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(2)} r_{ui}^{(2)} & 1 \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(1)} & \dots & \sum_{u=1}^m \sum_{i=1}^n r_{ui}^{(1)} r_{ui}^{(2)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(1)} \\ \sum_{u=1}^m \sum_{i=1}^n r_{ui} r_{ui}^{(2)} \end{bmatrix} \quad (22)$$

The optimal values of σ_1 and σ_2 can be computed using the matrix inversion process $X = A^{-1} \cdot B$. Here are the explicit formulas for the σ_1 and σ_2 equations (23).

$$\sigma_1 = \frac{\sum_{u=1}^m \sum_{i=1}^n (r_{ui} - r_{ui}^{(2)})(r_{ui}^{(1)} - r_{ui}^{(2)})}{\sum_{u=1}^m \sum_{i=1}^n (r_{ui}^{(1)} - r_{ui}^{(2)})^2}, \text{ and } \sigma_2 = 1 - \sigma_1 \quad (23)$$

3. Results and Discussion

3.1. Experimental dataset

This study conducts a comprehensive evaluation of various recommendation algorithms using benchmark datasets, including MovieLens 100K, MovieLens 1M, Amazon Cell Phones Reviews, and GoodBooks-10K, with a focus on state-of-the-art, baseline, hybrid, and optimization-based hybrid models. The state-of-the-art models include User-Based Collaborative Filtering (UBCF) and Item-Based Collaborative Filtering (IBCF), which leverage similarity-based neighborhood methods. The baseline models, Singular Value Decomposition (SVD) and its extension SVD++, serve as fundamental matrix factorization approaches that capture latent factor interactions. To improve recommendation accuracy, two hybrid models are examined: Hybrid Weighted Filtering (HWF), which combines SVD and SVD++ through a weighted average mechanism controlled by parameter α , and Hybrid Switching Filtering (HSF), which dynamically selects between the two models based on a threshold parameter θ . Further, two optimization-based hybrid methods are proposed: Optimized Hybrid Weighted Filtering (OHWF), which determines the optimal weighting coefficient σ_1 by minimizing the prediction error between SVD and SVD++ outputs, and Optimized Hybrid Weighted Switching Filtering (OHWSF), which integrates the HWF and HSF predictions using a similarly optimized weighting scheme.

Table 1 presents the essential features of the benchmark datasets utilized in this study. These datasets vary significantly in scale, number of ratings, and sparsity levels, offering a comprehensive and diverse evaluation environment for benchmarking recommendation algorithms. The MovieLens 100K dataset comprises 943 users, 1,682 items, and 100,000 ratings, with a sparsity level of 93.69%. In contrast, the larger MovieLens 1M dataset features 6,040 users, 3,952 items, and over one million ratings, resulting in a sparsity level of 95.81%. In comparison, the Amazon Cell Phones Reviews dataset is highly sparse, featuring 720 users and 47,322 items with only 67,986 ratings, resulting in a sparsity of 98.47%. The GoodBooks-10K dataset comprises 28,906 users and 794 items, with 79,701 ratings, resulting in a sparsity of 99.65%. These datasets were selected to provide a comprehensive assessment of the recommendation models under various levels of data sparsity and scale, thereby ensuring the robustness and generalizability of the experimental results.

Table 1. Features of the dataset

Dataset	Users	Items	Ratings	Sparsity
ML 100K	943	1,682	100,000	93.69%
ML 1M	6,040	3,952	1,000,209	95.81%
Amazon Cell Phones Reviews	720	47,322	67,986	99.80%
GoodBooks-10K	28,906	794	79,701	99.65%

3.2. Evaluation Metrics Comparison

A comprehensive evaluation was conducted to compare the predictive performance of state-of-the-art, baseline, hybrid, and optimization-based models across the four diverse datasets. All models are trained and tested using an 80%:20% data split, and their predictive performance is measured using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), providing a robust basis for comparative analysis. To handle the computational requirements for modeling large datasets, we utilized Google Colab with a Python 3 runtime and a hardware accelerator CPU, leveraging shared resources to ensure consistent performance. All experiments and results of the system are carried out using an AMD Ryzen 5 4500U with Radeon Graphics (6 CPUs) processor at 2.4 GHz, paired with 32 GB of memory.

Table 2 presents a comparative evaluation of eight recommendation algorithms across four benchmark datasets, assessing both predictive accuracy (using MAE, MSE, and RMSE) and training efficiency (measured in seconds). Traditional neighborhood-based models, such as the UBCF and the IBCF, consistently recorded the highest error rates and offered no significant gains in sparse scenarios. In the ML-100K dataset, UBCF showed the worst MAE (0.7481) and RMSE (0.9500) with moderate training time (0.275s), while IBCF achieved better accuracy (MAE: 0.7320; RMSE: 0.9319) with extremely low training time (0.036s). Matrix factorization methods, such as SVD and SVD++, have significantly improved accuracy, with SVD++ achieving an RMSE of 0.9206 on ML-100K and 0.8622 on ML-1M, albeit at the highest training costs (18.752s and 368.412s, respectively).

Hybrid models, such as HWF and HSF, offered moderate enhancements over baselines with remarkably lower training times (as low as 0.001 s–0.003 s). Most notably, the optimization-based OHWF and OHWSF models consistently outperformed others, effectively balancing accuracy and efficiency. For instance, OHWF achieved the lowest RMSE on ML-1M (0.8540) and Amazon Reviews (0.0857), while OHWSF delivered the best result on GoodBooks-10K (RMSE: 0.9024) with minimal training time (0.001s). These results underscore the capability of OHWSF to adaptively learn the optimal fusion of HWF and HSF using the σ_1 parameter, providing accurate, efficient, and generalizable predictions even in high-sparsity environments. Overall, the inclusion of dynamic switching and weight optimization proves to be a valuable advancement in addressing the limitations of traditional and static hybrid recommendation models.

Table 2. Performance Comparison of All Models Across Multiple Datasets

Dataset	Models	Training Time (Seconds)	MAE	MSE	RMSE
ML-100K	UBCF	0.275	0.748054	0.902492	0.949996
	IBCF	0.036	0.731999	0.868475	0.93192
	SVD	0.968	0.73808	0.876531	0.936232
	SVD++	18.752	0.721104	0.847517	0.920607
	HWF ($\alpha = 0.5$)	0.002	0.72125	0.83937	0.916171
	HSF ($\theta = 3.0$)	0.001	0.726099	0.855603	0.924988
	OHWF ($\sigma_1 = 0.3399$)	0.007	0.719446	0.837047	0.914903
	OHWSF ($\sigma_1 = 0.8583$)	0.017	0.720922	0.838915	0.915923
ML-1M	UBCF	34.697	0.738266	0.863668	0.929337
	IBCF	8.176	0.694460	0.785728	0.886413
	SVD	10.682	0.684765	0.761609	0.872702
	SVD++	368.412	0.672110	0.74343	0.862224
	HWF ($\alpha = 0.5$)	0.003	0.669784	0.730325	0.854590
	HSF ($\theta = 3.0$)	0.003	0.675242	0.748402	0.865102
	OHWF ($\sigma_1 = 0.3976$)	0.006	0.668854	0.729394	0.854046
	OHWSF ($\sigma_1 = 0.9072$)	0.049	0.669545	0.730134	0.854479
Amazon Cell Phones Reviews	UBCF	0.033	0.227719	0.12209	0.349414
	IBCF	0.205	0.302479	0.15168	0.389461
	SVD	0.881	0.037253	0.009679	0.09838
	SVD++	23.811	0.027040	0.007348	0.085722
	HWF ($\alpha = 0.5$)	0.001	0.030343	0.008152	0.090288
	HSF ($\theta = 3.0$)	0.036	0.027666	0.007522	0.086728
	OHWF ($\sigma_1 = 0.000$)	0.027	0.027040	0.007348	0.085722
	OHWSF ($\sigma_1 = 0.000$)	0.001	0.027666	0.007522	0.086728
GoodBooks-10K	UBCF	0.025	0.762709	1.021719	1.010801
	IBCF	0.067	0.741138	0.944633	0.971922
	SVD	0.907	0.728941	0.83099	0.911587
	SVD++	1.016	0.720632	0.817129	0.903952
	HWF ($\alpha = 0.5$)	0.001	0.720966	0.815775	0.903203
	HSF ($\theta = 3.0$)	0.008	0.720537	0.817128	0.903951
	OHWF ($\sigma_1 = 0.2909$)	0.007	0.719944	0.814326	0.902400
	OHWSF ($\sigma_1 = 0.5817$)	0.001	0.719897	0.814325	0.902400

3.3. Ablation Study: Impact of Tuning Parameter for Contribution of Each Component (Switching Vs. Weighting Optimization)

The novelty of the proposed framework lies in its dynamic parameter tuning (α and θ) and the introduction of a two-level optimization strategy (OHWF and OHWSF), enabling the models to adaptively balance the contributions of latent and neighborhood-based methods. Consequently, the optimized hybrid models not only deliver better prediction accuracy but also demonstrate robustness in handling datasets with high sparsity and varying distribution characteristics.

This section presents an ablation study to investigate the individual and joint contributions of the switching and weighting mechanisms within the Optimization Hybrid Weighted Switching Filtering (OHWSF) model, which emerged as the best-performing framework in prior evaluations. The objective is to analyze how variations in the hyperparameters α (α), governing the Hybrid Weighted Filtering (HWF) mechanism, and θ (θ), determining the threshold in Hybrid Switching Filtering (HSF), affect overall recommendation performance. Specifically, α controls the weighted combination between SVD and SVD++, while θ determines the decision boundary for switching between them based on prediction confidence.

Table 3 outlines the tuning parameter space explored in this study, with α values ranging from 0.1 to 0.9 and θ ranging from 1.0 to 5.0 in increments of 0.5. Through an extensive grid search on the validation set, the optimal α and θ were identified, which were then used to compute the final OHWSF predictions via an optimized linear combination of HWF and HSF outputs. This systematic tuning enables a deeper understanding of the relative importance and synergy between the two hybridization strategies, providing empirical guidance for parameter selection in real-world applications

Table 3. Tuning Parameter

Parameters	Values
Alpha (α)	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
Threshold (θ)	[1.0, 1.5, 2.0, ... , 5.0]

Fig. 1 illustrates the performance comparison between HWF and HSF on the ML-100K dataset, revealing that HWF achieved the lowest errors at $\alpha = 0.7$ with MAE = 0.7192, MSE = 0.8372, and RMSE = 0.9150. In contrast, the HSF model attained its best performance at $\theta = 1.5$ with slightly higher error values: MAE = 0.7207, MSE = 0.8473, and RMSE = 0.9205. These results indicate that the weighted fusion strategy in HWF is more effective than the switching mechanism in HSF under this dataset's sparsity level. Additionally, the OHWSF model adaptively converged to the HWF output ($\sigma_1 = 1.000$), confirming the superior influence of weighted integration in this context.

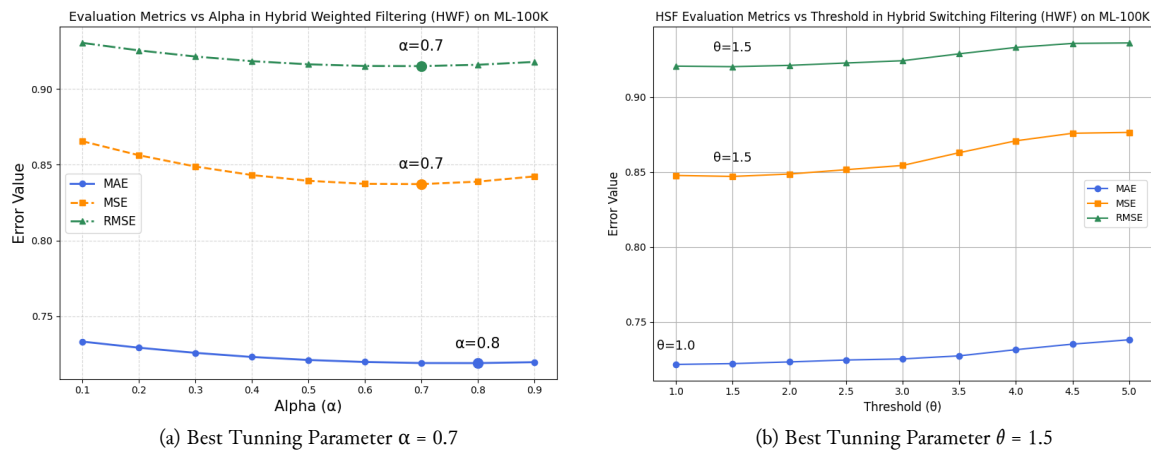


Fig. 1. Comparison of (a) and (b) based on ML-100K

Fig. 2 illustrates the evaluation of Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) on the ML-1M dataset, where tuning of the α and θ parameters significantly influences prediction accuracy. In subfigure (a), the optimal performance for HWF is achieved at $\alpha = 0.6$, yielding the lowest error values: MAE = 0.6697, MSE = 0.7308, and RMSE = 0.8549, indicating that an appropriate balance between SVD and SVD++ contributes to better generalization. Subfigure (b) demonstrates that HSF attains its best accuracy at $\theta = 1.0$, although its performance is slightly inferior to HWF, with MAE = 0.6727, MSE = 0.7443, and RMSE = 0.8627. These findings suggest that for ML-1M, the weighted fusion strategy is more effective than threshold-based switching, especially under moderately sparse conditions.

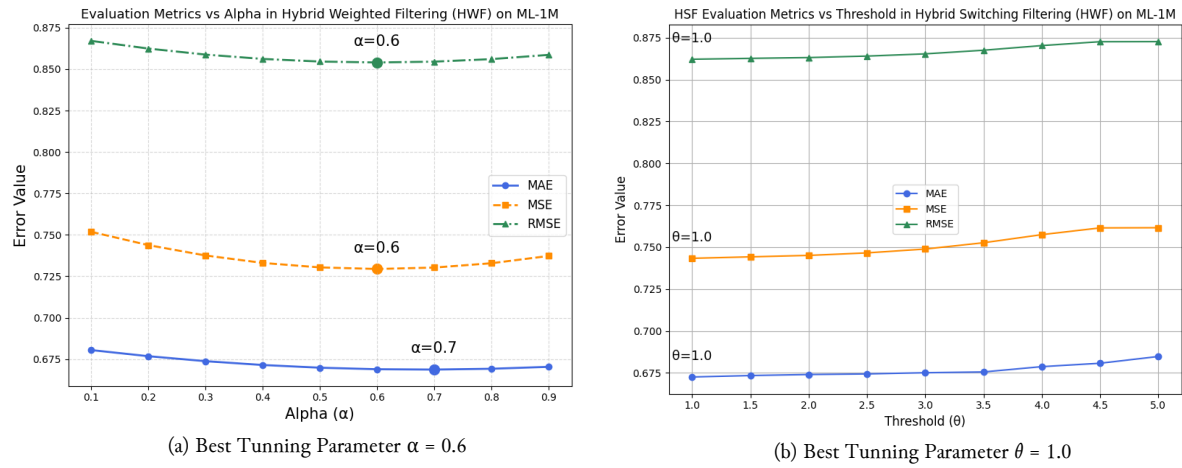


Fig. 2. Comparison of (a) and (b) based on M-1M

Fig. 3 presents a tuning analysis of Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) on the Amazon Cell Phones Reviews dataset. In subfigure (a), the HWF model shows optimal performance at $\alpha = 0.9$, resulting in the lowest error rates of MAE = 0.0272, MSE = 0.00745, and RMSE = 0.0863, indicating that a higher emphasis on SVD++ enhances predictive accuracy due to its handling of implicit feedback. Subfigure (b) demonstrates that HSF performs best at $\theta = 1.0$, where MAE = 0.0270, MSE = 0.00735, and RMSE = 0.0857, validating the effectiveness of SVD++ dominance when above the confidence threshold. These results highlight the model's sensitivity to parameter settings and confirm that tailored weighting and switching strategies significantly enhance recommendation quality in highly sparse environments.

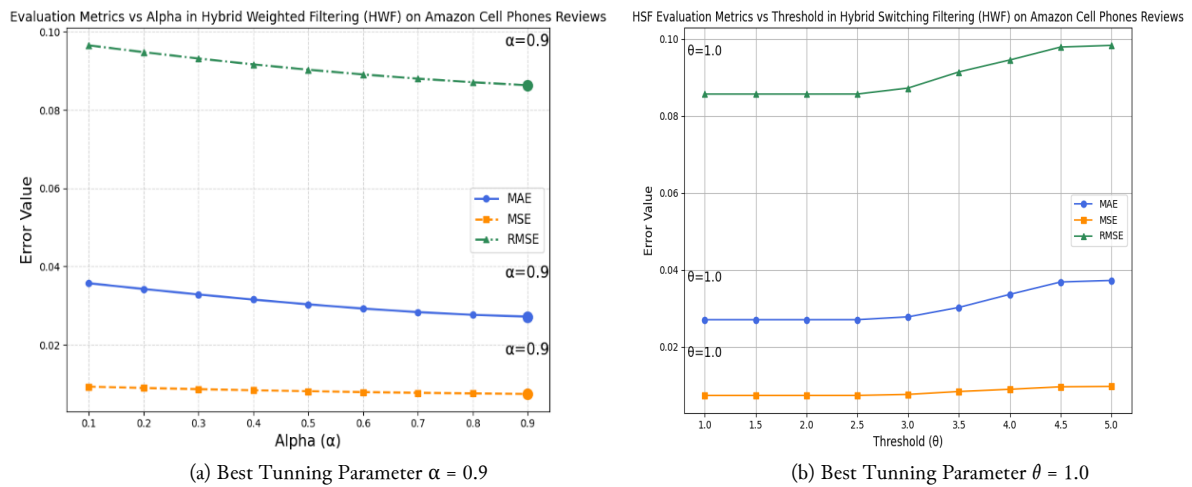


Fig. 3. Comparison of (a) and (b) based on Amazon Cell Phones Reviews

Fig. 4 presents the evaluation of parameter tuning for the Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) models on the GoodBooks-10K dataset, demonstrating the impact of optimal α and θ values on predictive performance. In subfigure (a), the HWF model attains its best performance at $\alpha = 0.8$, achieving the lowest error metrics MAE = 0.71996, MSE = 0.81433, and RMSE = 0.90240 by favoring SVD++'s contribution, which is particularly effective in addressing extreme sparsity. Subfigure (b) indicates that HSF reaches optimal results at $\theta = 1.0$. However, its performance remains slightly inferior, with higher MAE and RMSE values due to less effective switching decisions in this sparse environment. These findings confirm that emphasizing SVD++ through a weighted fusion strategy outperforms threshold-based switching, reinforcing the importance of adaptive weighting in handling datasets with limited user-item interactions.

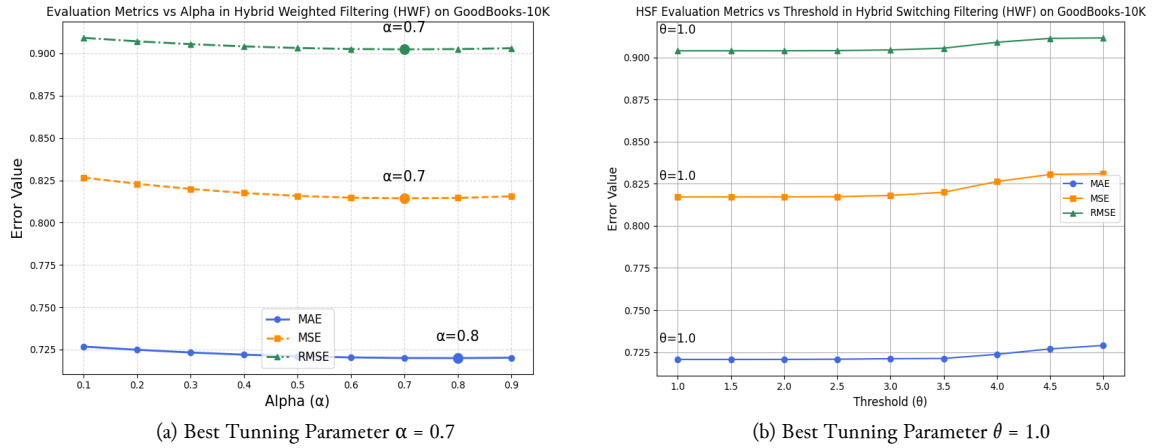


Fig. 4. Comparison of (a) and (b) based on GoodBooks-10K

The comparative analysis of best-tuned parameters for Hybrid Weighted Filtering (HWF), Hybrid Switching Filtering (HSF), and the Optimized Hybrid Weighted Switching Filtering (OHWSF) models across four diverse benchmark datasets demonstrates consistent improvements in both predictive accuracy and computational efficiency when optimal tuning is applied as show in Table 4. On the ML-100K dataset, OHWSF achieved an MAE of 0.7192, MSE of 0.8372, and RMSE of 0.9149 with just 0.034 seconds of training time, resulting in a 4.01% MAE improvement over the baseline. In the ML-1M dataset, OHWSF produced an MAE of 0.6688, MSE of 0.7294, and RMSE of 0.8540 in 0.147 seconds, achieving a 10.38% MAE gain compared to UBCF. On the Amazon Cell Phones Reviews dataset, which is highly sparse, OHWSF delivered the lowest MAE (0.0270), MSE (0.0073), and RMSE (0.0857) in just 0.007 seconds, showing a remarkable 742.16% MAE improvement. For the GoodBooks-10K dataset, OHWSF achieved an MAE of 0.7199, MSE of 0.8143, and RMSE of 0.9024 within 0.147 seconds, improving MAE by 5.94% relative to the baseline. These substantial percentage gains in MAE illustrate OHWSF's ability to adaptively optimize hybrid strategies across different data sparsity levels. Overall, the OHWSF model stands out as a robust, scalable, and accurate solution for modern recommender systems operating in diverse and sparse environments.

Table 4. Best Tuning Parameters Performance Comparison

Dataset	Models	Training Time (Seconds)	MAE	MSE	RMSE	MAE Impv (%)
ML-100K	HWF ($\alpha = 0.7$)	0.250	0.719231	0.837192	0.914982	4.01
	HSF ($\theta = 1.5$)	0.162	0.720673	0.847328	0.920504	3.80
	OHWSF ($\sigma_1 = 1.000$)	0.034	0.719231	0.837192	0.914982	4.01
ML-1M	HWF ($\alpha = 0.6$)	0.598	0.668867	0.729395	0.854046	10.38
	HSF ($\theta = 1.0$)	0.457	0.672004	0.743382	0.862196	9.86
	OHWSF ($\sigma_1 = 0.9915$)	0.147	0.668848	0.729394	0.854045	10.38
Amazon Cell Phones Reviews	HWF ($\alpha = 0.9$)	0.227	0.027209	0.007451	0.086320	736.93
	HSF ($\theta = 1.0$)	0.183	0.027040	0.007348	0.085722	742.16
	OHWSF ($\sigma_1 = 0.000$)	0.007	0.027040	0.007348	0.085722	742.16
GoodBooks-10K	HWF ($\alpha = 0.7$)	0.227	0.719961	0.814329	0.902402	5.94
	HSF ($\theta = 1.0$)	0.183	0.720632	0.817129	0.903952	5.84
	OHWSF ($\sigma_1 = 0.9695$)	0.147	0.719944	0.814326	0.902400	5.94

4. Conclusion

This study proposed and rigorously evaluated the Optimization Hybrid Weighted Switching Filtering (OHWSF) model, a novel hybrid recommender system designed to improve predictive accuracy and mitigate data sparsity challenges. By integrating Hybrid Weighted Filtering (HWF) and Hybrid Switching Filtering (HSF) with a dynamic optimization mechanism controlled by the tunable parameter σ_1 , OHWSF adaptively balances the contributions of SVD and SVD++ based on actual rating deviations.

Experimental results across four benchmark datasets, ML-100K, ML-1M, Amazon Cell Phones Reviews, and GoodBooks-10K, consistently demonstrate the superiority of OHWSF in achieving lower error rates (MAE, MSE, RMSE) compared to traditional models (UBCF, IBCF), baseline matrix factorization methods (SVD, SVD++), and standalone hybrid models. The performance gains were particularly pronounced when optimal values for α and θ were applied, with HWF achieving best results in highly sparse scenarios (e.g., $\alpha = 0.9$ on Amazon and $\alpha = 0.8$ on GoodBooks). At the same time, HSF performed better with lower thresholds (e.g., $\theta = 1.0$ across datasets). The OHWSF model effectively leveraged this tuning, dynamically adjusting its internal weighting (σ_1) to favor the more accurate strategy per dataset. Moreover, OHWSF achieved significant computational efficiency, delivering top-tier accuracy with a lower training time than complex models, such as SVD++. Future research should explore several promising directions to build more adaptive and robust recommender systems. First, incorporating context-aware factors such as temporal dynamics, user demographics, and session-based behaviors could enhance personalization. Second, extending the OHWSF framework to integrate deep learning-based embeddings or graph neural networks may enable richer user-item representations and improved scalability in large-scale, heterogeneous data environments. Lastly, reinforcement learning-based optimization mechanisms may enable real-time model updating and self-adaptive learning without full retraining. These directions offer a pathway toward developing next-generation recommender systems that are not only accurate and scalable but also dynamically responsive to user needs and evolving data conditions.

Acknowledgment

The authors gratefully acknowledge Indonesian Education Scholarship (BPI), Center for Higher Education Funding and Assessment (PPAT), and Indonesian Endowment Fund for Education (LPDP) for support and fund this research with grant number 01519/J5.2.3./BPI.06/9/2022 and all facilities provided by Department of Mathematics, Faculty Mathematics and Natural Science Universitas Gadjah Mada.

Declarations

Author contribution. Dedi Rosadi and Malim Muhammad conceived of the presented idea. Malim Muhammad developed the theory, performed the computations, and verified the analytical methods. Gunardi and Danardono supervised the method and findings of this work. All authors discussed the results and contributed to the preparation of the final manuscript

Funding statement. The authors would like to express their gratitude and appreciation to the Indonesian Education Scholarship (BPI), the Center for Higher Education Funding and Assessment (PPAT), and the Indonesian Endowment Fund for Education (LPDP)

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The datasets used in this paper can be accessed through the following two links:

- ML-100K: <https://grouplens.org/datasets/movielens/100k/>
- ML-1M: <https://grouplens.org/datasets/movielens/1m/>
- Amazon Cell Phones: <https://www.kaggle.com/datasets/grikomsn/amazon-cell-phones-reviews?select=20191226-items.csv>
- GoodBooks-10K: <https://www.kaggle.com/datasets/zygmunt/goodbooks-10k/data?select=books.csv>

References

- [1] M. Arnold, M. Goldschmitt, and T. Rigotti, "Dealing with information overload: a comprehensive review," *Front. Psychol.*, vol. 14, no. June, 2023, doi: [10.3389/fpsyg.2023.1122200](https://doi.org/10.3389/fpsyg.2023.1122200).

- [2] B. D. Okkalioglu, "A Novel Hybrid Item-Based Similarity Method to Mitigate the Effects of Data Sparsity in Multi-Criteria Collaborative Filtering," *IEEE Access*, vol. 13, no. March, pp. 64660–64686, 2025, doi: [10.1109/ACCESS.2025.3559398](https://doi.org/10.1109/ACCESS.2025.3559398).
- [3] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent Developments in Recommender Systems: A Survey [Review Article]," *IEEE Comput. Intell. Mag.*, vol. 19, no. 2, pp. 78–95, 2024, doi: [10.1109/MCI.2024.3363984](https://doi.org/10.1109/MCI.2024.3363984).
- [4] I. Saifudin and T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," *IEEE Access*, vol. 12, pp. 19827–19847, 2024, doi: [10.1109/ACCESS.2024.3359274](https://doi.org/10.1109/ACCESS.2024.3359274).
- [5] S. Peng, S. Siet, S. Ilkhomjon, D. Y. Kim, and D. S. Park, "Integration of Deep Reinforcement Learning with Collaborative Filtering for Movie Recommendation Systems," *Appl. Sci.*, vol. 14, no. 3, 2024, doi: [10.3390/app14031155](https://doi.org/10.3390/app14031155).
- [6] Z. Dong, X. Liu, B. Chen, P. Polak, and P. Zhang, "MuseChat: A Conversational Music Recommendation System for Videos," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 12775–12785, doi: [10.1109/CVPR52733.2024.01214](https://doi.org/10.1109/CVPR52733.2024.01214).
- [7] D. T. Tran and J. H. Huh, "New machine learning model based on the time factor for e-commerce recommendation systems," *J. Supercomput.*, vol. 79, no. 6, pp. 6756–6801, 2023, doi: [10.1007/s11227-022-04909-2](https://doi.org/10.1007/s11227-022-04909-2).
- [8] J. N. Bondevik, K. E. Bennin, Ö. Babur, and C. Ersch, "A systematic review on food recommender systems," *Expert Syst. Appl.*, vol. 238, p. 122166, 2024, doi: [10.1016/j.eswa.2023.122166](https://doi.org/10.1016/j.eswa.2023.122166).
- [9] J. L. Sarkar, A. Majumder, C. R. Panigrahi, S. Roy, and B. Pati, "Tourism recommendation system: a survey and future research directions," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 8983–9027, 2023, doi: [10.1007/s11042-022-12167-w](https://doi.org/10.1007/s11042-022-12167-w).
- [10] Z. Li, Y. Chen, X. Zhang, and X. Liang, "BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model," *Electron.*, vol. 12, no. 22, pp. 1–19, 2023, doi: [10.3390/electronics12224654](https://doi.org/10.3390/electronics12224654).
- [11] C. Wu, F. Wu, Y. Huang, and X. Xie, "Personalized News Recommendation: Methods and Challenges," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, pp. 1–50, Jan. 2023, doi: [10.1145/3530257](https://doi.org/10.1145/3530257).
- [12] A. Yashudas, D. Gupta, G. C. Prashant, A. Dua, D. AlQahtani, and A. S. K. Reddy, "DEEP-CARDIO: Recommendation System for Cardiovascular Disease Prediction Using IoT Network," *IEEE Sens. J.*, vol. 24, no. 9, pp. 14539–14547, 2024, doi: [10.1109/JSEN.2024.3373429](https://doi.org/10.1109/JSEN.2024.3373429).
- [13] K. Sharma *et al.*, "A Survey of Graph Neural Networks for Social Recommender Systems," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–34, Oct. 2024, doi: [10.1145/3661821](https://doi.org/10.1145/3661821).
- [14] F. L. da Silva, B. K. Slodkowski, K. K. A. da Silva, and S. C. Cazella, "A systematic literature review on educational recommender systems for teaching and learning: research trends, limitations and opportunities," *Educ. Inf. Technol.*, vol. 28, no. 3, pp. 3289–3328, 2023, doi: [10.1007/s10639-022-11341-9](https://doi.org/10.1007/s10639-022-11341-9).
- [15] Y. Koren, S. Rendle, and R. Bell, "Advances in Collaborative Filtering," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. New York, NY: Springer US, 2022, pp. 91–142, doi: [10.1007/978-1-0716-2197-4_3](https://doi.org/10.1007/978-1-0716-2197-4_3).
- [16] H. Khojamli and J. Razmara, "Survey of similarity functions on neighborhood-based collaborative filtering," *Expert Syst. Appl.*, vol. 185, p. 115482, 2021, doi: [10.1016/j.eswa.2021.115482](https://doi.org/10.1016/j.eswa.2021.115482).
- [17] H. Liu, Y. Wang, Z. Zhang, J. Deng, C. Chen, and L. Y. Zhang, "Matrix factorization recommender based on adaptive Gaussian differential privacy for implicit feedback," *Inf. Process. Manag.*, vol. 61, no. 4, p. 103720, 2024, doi: [10.1016/j.ipm.2024.103720](https://doi.org/10.1016/j.ipm.2024.103720).
- [18] Z. Fayyaz, M. Ebrahimi, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Appl. Sci.*, vol. 10, no. 7748, pp. 1–20, 2020, doi: [10.3390/app10217748](https://doi.org/10.3390/app10217748).

- [19] Q. Yu, M. Zhao, and Y. Luo, "Collaborative Filtering Hybrid Recommendation Algorithm based on Optimal Weight," in *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, 2022, pp. 144–147, doi: [10.1109/ICCEAI55464.2022.00038](https://doi.org/10.1109/ICCEAI55464.2022.00038).
- [20] P. Li, J. Cao, Z. Guan, and F. Hang, "Collaborative Filtering Hybrid Recommendation Algorithm Based on Improved Score Similarity," in *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2023, pp. 1328–1332, doi: [10.1109/ICSP58490.2023.10248224](https://doi.org/10.1109/ICSP58490.2023.10248224).
- [21] A. M. Shetty, D. H. Manjaiah, M. F. Aljunid, and K. M. Yogesh, "Comparative Analysis of Memory-Based Collaborative Filtering and Deep Learning Models for Resolving Cold Start and Data Sparsity Issues in E-commerce Recommender Systems," *2024 IEEE 3rd World Conf. Appl. Intell. Comput. AIC 2024*, pp. 81–87, 2024, doi: [10.1109/AIC61668.2024.10730916](https://doi.org/10.1109/AIC61668.2024.10730916).
- [22] T. M. A. U. Gunathilaka, P. D. Manage, J. Zhang, Y. Li, and W. Kelly, "Addressing sparse data challenges in recommendation systems: A systematic review of rating estimation using sparse rating data and profile enrichment techniques," *Intell. Syst. with Appl.*, vol. 25, no. January, p. 200474, 2025, doi: [10.1016/j.iswa.2024.200474](https://doi.org/10.1016/j.iswa.2024.200474).
- [23] G. Behera, N. Nain, and R. K. Soni, "Integrating user-side information into matrix factorization to address data sparsity of collaborative filtering," *Multimed. Syst.*, vol. 30, no. 2, pp. 1–18, 2024, doi: [10.1007/s00530-024-01261-8](https://doi.org/10.1007/s00530-024-01261-8).
- [24] S. M. Choi, D. Lee, K. Jang, C. Park, and S. Lee, "Improving Data Sparsity in Recommender Systems Using Matrix Regeneration with Item Features," *Mathematics*, vol. 11, no. 2, pp. 1–26, 2023, doi: [10.3390/math11020292](https://doi.org/10.3390/math11020292).
- [25] W. Shi and Y. Zhang, "Personalized Recommendation for Online News Based on UBCF and IBCF Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 4, pp. 326–337, 2025, doi: [10.14569/IJACSA.2025.0160434](https://doi.org/10.14569/IJACSA.2025.0160434).
- [26] D. V, H. I, and V. K. K, "Enhanced Hybrid UBCF-IBCF Recommender Systems Using Pearson and Cosine Similarities for Improved Accuracy," in *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, 2024, pp. 1037–1044, doi: [10.1109/ICSCNA63714.2024.10863870](https://doi.org/10.1109/ICSCNA63714.2024.10863870).
- [27] J. Velumani, H. Mohmedmhdi, K. S., T. A. S. Srinivas, and H. D. Praveena, "Movies Recommendation System using User-Based Collaborative Filtering for Social Network Analysis," in *2025 3rd International Conference on Data Science and Information System (ICDSIS)*, 2025, pp. 1–5, doi: [10.1109/ICDSIS65355.2025.11070953](https://doi.org/10.1109/ICDSIS65355.2025.11070953).
- [28] N. Azri, A. Haddi, and A. Azri, "Enhancing Recommender Systems through Hybrid Fusion of SVD/SVD++ and k-Nearest Neighbors," in *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2023, pp. 1–7–1–7, doi: [10.1109/SITA60746.2023.10373705](https://doi.org/10.1109/SITA60746.2023.10373705).
- [29] Z. T. Yap, S. C. Haw, and N. E. Binti Ruslan, "Hybrid-based food recommender system utilizing KNN and SVD approaches," *Cogent Eng.*, vol. 11, no. 1, p., 2024, doi: [10.1080/23311916.2024.2436125](https://doi.org/10.1080/23311916.2024.2436125).
- [30] M. M. ul Haque, B. Kotaiah, and J. Ahamed, "Hybrid course recommendation system using SVD, NMF and attention-based neural network," *Int. J. Inf. Technol.*, vol. 17, no. 4, pp. 2449–2456, 2025, doi: [10.1007/s41870-025-02440-0](https://doi.org/10.1007/s41870-025-02440-0).
- [31] Aamana, N. Iltaf, H. Afzal, and Q. U. Ain, "Co-Clustering based Hybrid Collaborative Filtering Model," *4th Int. Conf. Commun. Technol. ComTech 2023*, pp. 18–27, 2023, doi: [10.1109/ComTech57708.2023.10165021](https://doi.org/10.1109/ComTech57708.2023.10165021).
- [32] D. K. Behera, M. Das, S. Swetanisha, and P. K. Sethy, "Hybrid model for movie recommendation system using content K-nearest neighbors and restricted boltzmann machine," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 445–452, 2021, doi: [10.11591/ijeecs.v23.i1.pp445-452](https://doi.org/10.11591/ijeecs.v23.i1.pp445-452).
- [33] N. M. Khairudin, N. Mustapha, T. N. M. Aris, and M. Zolkepli, "Hybrid machine learning model based on feature decomposition and entropy optimization for higher accuracy flood forecasting," *Int. J. Adv. Intell. Informatics*, vol. 10, no. 1, pp. 1–12, 2024, doi: [10.26555/ijain.v10i1.1130](https://doi.org/10.26555/ijain.v10i1.1130).
- [34] A. Sonule, H. Jagtap, and V. Mendhe, "Weighted Hybrid Recommendation System," *Int. J. Res. Anal. Rev.*, no. March, p. 402, 2024. [Online]. Available at: https://www.researchgate.net/publication/378846446_Weighted_Hybrid_Recommendation_System.

- [35] C. Song, Q. Yu, E. Jose, J. Zhuang, and H. Geng, "A hybrid recommendation approach for viral food based on online reviews," *Foods*, vol. 10, no. 8, 2021, doi: [10.3390/foods10081801](https://doi.org/10.3390/foods10081801).
- [36] S. S. E. Alqallaf, W. M. Medhat, and T. A. El-Shishtawy, "A Hybrid Recommender Framework for Selecting a Course Reference Books," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 4, pp. 1004–1014, 2022.[Online]. Available at: <http://www.jatit.org/volumes/Vol100No4/10Vol100No4.pdf>.
- [37] A. A. Amer, H. I. Abdalla, and L. Nguyen, "Enhancing recommendation systems performance using highly-effective similarity measures," *Knowledge-Based Syst.*, vol. 217, p. 106842, 2021, doi: [10.1016/j.knosys.2021.106842](https://doi.org/10.1016/j.knosys.2021.106842).
- [38] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework utilizing social networks," *Mach. Learn. with Appl.*, vol. 14, no. January, p. 100495, 2023, doi: [10.1016/j.mlwa.2023.100495](https://doi.org/10.1016/j.mlwa.2023.100495).
- [39] A. Gazdar and M. Kefi, "A Recommender System for Linear Satellite TV: Is It Possible?," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, 2020, pp. 1–8, doi: [10.1109/ICCAIS48893.2020.9096718](https://doi.org/10.1109/ICCAIS48893.2020.9096718).
- [40] Y. Chen, "A music recommendation system based on collaborative filtering and SVD," *2022 IEEE Conf. Telecommun. Opt. Comput. Sci. TOCS 2022*, pp. 1510–1513, 2022, doi: [10.1109/TOCS56154.2022.10016210](https://doi.org/10.1109/TOCS56154.2022.10016210).
- [41] S. Jiang, J. Li, and W. Zhou, "An Application of SVD++ Method in Collaborative Filtering," in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2020, pp. 192–197, doi: [10.1109/ICCWAMTIP51612.2020.9317347](https://doi.org/10.1109/ICCWAMTIP51612.2020.9317347).
- [42] A. Chaudhari, A. A. H. Seddig, A. Sarlan, and R. Raut, "A Hybrid Recommendation System: A Review," *IEEE Access*, vol. 12, pp. 157107–157126, 2024, doi: [10.1109/ACCESS.2024.3480693](https://doi.org/10.1109/ACCESS.2024.3480693).
- [43] B. Sabiri, A. Khtira, B. El Asri, and M. Rhanoui, "Hybrid Quality-Based Recommender Systems: A Systematic Literature Review," *Journal of Imaging*, vol. 11, no. 1, 2025, doi: [10.3390/jimaging11010012](https://doi.org/10.3390/jimaging11010012).
- [44] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering," *J. Appl. Data Sci.*, vol. 4, no. 3, pp. 289–302, 2023, doi: [10.47738/jads.v4i3.115](https://doi.org/10.47738/jads.v4i3.115).
- [45] O. Remadnia, F. Maazouzi, and D. Chefrour, "Hybrid Book Recommendation System Using Collaborative Filtering and Embedding Based Deep Learning," *Inform.*, vol. 49, no. 8, pp. 189–204, 2025, doi: [10.31449/inf.v49i8.6950](https://doi.org/10.31449/inf.v49i8.6950).
- [46] C. C. Aggarwal, *Recommender Systems*. Springer International Publishing AG Switzerland, 2016, p. 489, doi: [10.1007/978-3-319-29659-3](https://doi.org/10.1007/978-3-319-29659-3).
- [47] M. M. Rahman, I. A. Shama, M. S. Rahman, and M. R. Nabil, "Hybrid Recommendation System To Solve Cold," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 11, pp. 3562–3578, 2022. [Online]. Available at: <http://www.jatit.org/volumes/Vol100No11/7Vol100No11.pdf>.