Journal of Computer Science

Research Article

Hybrid Enhanced Neighbourhood and Latent Factor Model for Cold-Start Recommendations

Amritha P and Rajkumar KK

Department of Information Technology, Kannur University, Kerala, India

Article history
Received: 04-02-2026
Revised: 19-04-2026
Accepted: 03-08-2026

Corresponding Author:
Amritha P
Department of Information Technology, Kannur University, Kerala, India
Email: amritha@chintech.ac.in

Abstract: Among different recommendation strategies, collaborative filtering remains a commonly utilized method for generating personalized suggestions. The traditional collaborative algorithms face performance declines due to the sparse rating of data and the item cold-start problem. To overcome these challenges, this paper introduces a novel hybrid model called HCE-KNNCF (Hybrid Cognition-Enabled K-Nearest Neighbor Collaborative Filtering). The proposed model generates predicted rating by a combination of SVD-based matrix factorization and the enhanced KNN model using a weighted hybrid approach. The cognition-based KNN ensures that only relevant neighbors contribute to the rating prediction phase and the SVD-based collaborative approach is employed to model latent user-item relationships, thereby mitigating the effects of data sparsity. Experimental evaluations on the MovieLens 100 K, MovieLens 1 M, and Book-Crossing datasets show that HCE-KNNCF achieves improved prediction accuracy compared with most traditional and hybrid benchmark models. The model achieves the best MAE and RMSE results on the MovieLens 100 K and Book-Crossing datasets, while maintaining competitive performance on MovieLens 1 M. In cold-start scenarios, HCE-KNNCF demonstrate that a small increase in MAE and RMSE, indicating that the proposed approach remains stable when interaction data are limited.

Keywords: Collaborative Filtering, Latent Factor Models, Neighborhood Methods, Hybrid Recommendation

Introduction

With expansion of digital ecosystems, including social media, e-commerce, and online streaming services, users are continuously exposed to massive volumes of information. Overloading individuals with vast amounts of information leads to a phenomenon commonly referred to as information overload. This abundance of information will make it difficult to distinguish between relevant and irrelevant information. To address this challenge, recommendation systems have emerged as techniques that generate personalize information based on user preferences. Today, these systems are helping users to select suitable products from the variety of options available across the digital platforms (Sinha and Dhanalakshmi, 2019). Recommendation or recommender systems can be defined as advanced algorithms that generate delivering personalized content tailored to individual user preferences. Recommendation systems are categorized into different strategies, each adopting unique techniques to improve the accuracy of recommendations. The content model generates recommendations by learning attributes like keywords, metadata, and descriptions and mapping them with the user's established preferences (Lops et al., 2019). Collaborative Filtering (CF) mainly relies on users' feedback; this input can be either direct feedback like ratings or implicit ratings like views, clicks, etc. By modeling shared behavioral patterns, collaborative filtering recommends items based on shared user interests (Chen et al., 2018). Hybrid methods combine these strategies and generate more enhanced suggestions for the user.

Collaborative filtering is a widely used recommendation technique that infers user preferences by examining the behaviors and interactions of like-minded users. The fundamental principle of CF assumes that individuals who demonstrated similar interests previously are likely to share related preferences in subsequent interactions (Chen et al., 2018). These preferences are captured through user feedback and represented in a user-item preference matrix. The system then analyzes this matrix to identify patterns and similarities between users or items (Chen et al., 2018). The user's preferences are expressed in two ways, and they are called implicit and explicit feedback. Explicit feedback is the direct ratings or scores given by users for items. Implicit feedback is derived from user behavior, such as clicks, views, and likes (Chen et al., 2018; Roy and Dutta, 2022). Recommender systems typically address two primary tasks: Rating prediction and top-N recommendation. The rating prediction is a regression problem where the system estimates the explicit rating that user is likely to give to an item based on observed data (Chen et al., 2018). Top-N recommendation is a ranking problem, focusing on identifying a set of N items that the user is most likely prefer among the items. So, a rating prediction, which aims to estimate exact scores, and a Top-N recommendation prioritizes generating a ranked list of items (Chen et al., 2018).

The CF method has been further categorized into memory-based and model-based (Breese et al., 1998; Linden et al., 2003). The collaborative KNN model is a popular memory-based approach. KNN-based CF methods are classified into user-based and item-based approaches: User based approach suggest items based on the preferences of users who exhibit behavioral similarity to the target user, while item-based CF suggests items that share similarities with those previously preferred by the user. This approach enables effective filtering of large volumes of data and providing personalized recommendations without relying on explicit item attribute information (Sarwar et al., 2001; Koren, 2009). Amazon's item-based collaborative filtering model is considered as one of the first large-scale systems to deliver personalized product recommendations (Su and Khoshgoftaar, 2009).

Model-based CF uses machine learning approaches to build predictive models from historical or observed data. Model-based recommendation techniques make use of Bayesian principles, clustering algorithms, Matrix Factorization (MF) techniques, and probabilistic learning to capture user-item relationships. These approaches, once built, are more scalable and require less online computation than memory-based methods (Su and Khoshgoftaar, 2009). Singular Value Decomposition (SVD) is a method that decomposes the user-item matrices into latent factors, and enable the system to predict missing ratings in the user-item rating matrix (Chen et al., 2018; Zhou et al., 2015). SVD++, is another type of SVD in which implicit ratings and temporal data are used for the prediction (Roy and Dutta, 2022). Another important model-based approaches are Probabilistic Matrix Factorization (PMF) (Wenga et al., 2021), which utilized Bayesian inference to improve predictive performance.

Traditional K-Nearest Neighbors methods depend on explicit user ratings and the calculation of similarities between users or items. This dependency becomes problematic in sparse datasets, where few user-item interactions exist, resulting in minimal overlap between user/item pairs. Thereby, the similarity computations in KNN become unreliable, reducing the effectiveness of recommendations. The SVD-based approaches are effective in such situations by uncovering latent factors that represent hidden relationships between users and items. The cold-start problem arises when new users or items enter without any user feedback. Consider an example where a new user has not interacted with any items or a new item without any ratings. In this case, new items will not get visibility and will never be recommended. This is because the system prioritizes items with higher ratings only. This situation can lead to biases towards popular or highly rated items in the recommendation. To solve these issues, hybrid recommendation systems that integrate the advantages of neighborhood-based models with model-based latent factor approaches are essential.

In this study, we have proposed a hybrid model, incorporating neighborhood learning with latent factor modeling to enhance accuracy and personalization. The weighted hybrid framework combines the outputs of enhanced KNN and regularized SVD to generate recommendations for both new (cold-start) and existing users. The implementation of the proposed model is influenced by a model called the adaptive KNN-Based extended collaborative filtering model (Nguyen et al., 2023).

Related Work

The domain of recommendation systems experiences extensive study and innovation, especially in addressing the challenge of insufficient data and lack of initial interaction history. Content-based recommendation systems depend exclusively on historical user preferences and item metadata to suggest items to users; this information includes item features, tags, descriptions, etc. This approach focuses on the user's previous interactions and fails to consider collaborative patterns, such as items popular among similar users (Chen et al., 2018). Collaborative filtering is considered the most important model for recommendation algorithms because it handles users shared preferences and behaviors. Researchers have introduced a wide range of models aimed at enhancing the existing recommendation systems. To achieve the improved benefits of traditional methods and improve recommendation accuracy, the use of more information is essential to design a hybrid recommendation model.

Lv et al. (2024) put forward a hybrid model based on a User Nearest Neighbor (UNN) method to improve performance under sparse conditions. Their approach first enhances similarity computation to better address missing user-item interactions and then applies collaborative filtering techniques. The strength of this study is its improved sparsity handling and scalability through Spark distributed implementation. However, the model depends on similarity enhancement and does not explicitly handle the item cold-start problem or incorporate richer contextual or cognitive factors for neighbor refinement.

Adding more on user-specific modification, Liang et al. (2024) came up with a behavior-aware model that differentiates users based on behavioral activity levels. The model categorizes users into active and inactive groups and applies SVD with content-based filtering for inactive users and a diversity-enhanced KNN approach for active users. This user-level differentiation improves sparsity handling and personalization. But the model does not address cold-start scenarios for newly introduced users or items.

Sharma and Shakya (2024) proposed a hybrid movie recommendation system that uses an artificial neural network to learn user-item relationships and improve prediction accuracy. By using neural learning, the model was able to capture non-linear preference patterns that are difficult for traditional neighborhood-based methods. The key advantage is improved modeling capacity and accuracy for movie recommendations. However, ANN-based hybrids typically introduce higher computational cost, require careful hyper parameter tuning, and may still face limitations under severe sparsity or cold-start conditions without sufficient auxiliary information.

Bahrani et al. (2024) come up with a KNN-based recommendation model that fuses statistical measures (mean and variance) of users and items ratings for the similarity computation process. In this approach, multiple variants called EVMRS (Ensemble Variance-Mean RS), EWVMRS (Ensemble Weighted Variance-Mean RS), and EWVMRSG (Gaussian mixture improved variant) were introduced. However, the approach is still memory-based, and lacks integration of content or temporal features, and may not handle cold-start users/items, since it relies heavily on rating statistics.

To handle the issues of sparsity and popularity bias, Guan et al. (2024) came up a with novel hybrid similarity model using Wasserstein distance for Collaborative Filtering (WCF). This approach uses Wasserstein distance to measure item similarity computation. In this method two heuristic measures called anti-popularity and anti-prominence are used to adjust for popular user/item biases. The WCF model addresses the cold-start problem in sparse environments by mitigating noise arising from unrated items. However, the use of Sinkhorn-based approximation for Wasserstein distance computation introduces computational overhead, which restrict this for large datasets.

Roy and Shetty (2025) introduced a hybrid recommendation application that integrates an Adaptive K-Nearest Neighbour (AKNN) with Singular Value Decomposition (SVD). In their method, a composite similarity computation approach is adopted by combining cosine similarity, Pearson correlation, and Variance Mean Difference (VMD). The final rating is computed by combining the outputs of both modules using a hybrid weighted method. The novelty of this work is the integration of multiple similarity measures within an adaptive KNN framework and combining it with latent factor modeling through a weighted hybrid strategy. However, dynamically varying the number of neighbors may introduce instability and affect model consistency.

Sun and Liu (2025) proposed a hybrid collaborative filtering framework, termed KNNCNMF, which integrates Non-Negative Matrix Factorization (NMF) with the K-Nearest Neighbor (KNN) method. In their approach, the user-item interaction matrix is first factorized using NMF to obtain latent feature representations. These learned latent features are subsequently utilized within a KNN-based prediction scheme to enhance rating estimation. The model relies on static latent characteristics and does not have the ability to adjust to dynamic user behaviors.

Ensemble and boosting-based hybrid models combine multiple learning models for improving recommendation performance. Behera et al. (2024) introduced a hybrid model that combines matrix factorization with the XGBoost method. Here MF is first used to learn user-item features and combined with contextual attributes and passed to the XGBoost model. The novelty of this approach lies in its effective use of ensemble learning to capture nonlinear relationships. However, the approach's limitation is the increased computational complexity due to the combination of MF and boosting, and it does not tackle the cold-start problem. By integrating content model, collaborative model, and supervised models with boosting approach, Singh et al. (2024) presented an ensemble learning hybrid recommendation system. Here boosting is used to mitigate individual model shortcomings, which is an advantage of this approach. Despite its strengths, the model is computationally complex and does not directly address cold-start problems.

Deep learning and meta-learning have recently transformed hybrid recommendation systems by enabling them to capture non-linear, high-level user-item interactions. As compared to conventional models that rely mainly on linear latent features, these frameworks learn complex semantic and temporal dependencies directly from data. Liu et al. (2023) put forward a hybrid approach that combines meta-learning and an attention mechanism to mitigate cold-start challenges. The attention part assigns weights to user-item interactions, enabling the model to focus on preference signals. The meta-learning part uses a Model-Agnostic Meta-Learning

(MAML) to consider user preference estimation as separate learning tasks, for the system to adapt to new users or items. This design improves the personalization in cold-start scenarios. However, the fusion of attention and meta-learning increases computational complexity when applied to highly sparse datasets. Building on probabilistic deep-learning methods, Sami et al. (2024) introduced HRS-IU-DL, a hybrid deep learning framework that combines memory-based Collaborative Filtering (CF) with neural collaborative filtering, recurrent neural networks, and content-based filtering using TF-IDF representations. This is able to learn non-linear relationships and sequential user behaviors and mitigate sparsity and cold-start problems. The method uses an N-sample genre-based strategy to generate top-10 recommendations and demonstrates superior performance on the Movie Lens 100 K dataset. The strength of this work is its deep integration of sequential and content signals, but its complexity and high computational cost limit the scalability for larger applications.

Hybrid collaborative filtering models combining similarity-based and latent factor approaches have improved sparsity and accuracy, but several limitations exhibit. Most of the approaches use fixed similarity measures and ignore user behavior, resulting in less meaningful neighbor selection. In addition, some of the methods are computationally expensive or fail to adjust to new users, new items, or evolving preferences. Deep and meta-learning-based hybrid models have improved the ability of recommendation systems to capture complex user-item relationships and address cold-start challenges. However, these approaches suffer from high computational complexity, scalability issues etc., due to their deep architectures. Our proposed model integrates cognition-enabled neighborhood learning with latent factor modeling to address these limitations of existing hybrid frameworks. By using user cognition parameters into the similarity computation, the model enhances the neighbor selection and ensures that only relevant users take part in the prediction process. The latent factor component captures the hidden associations between users and items, allowing the system to generalize effectively even in sparse or cold-start conditions.

Materials and Methods

KNN Based Collaborative Algorithm

Collaborative filtering is a recommendation technique used to predict and recommend items that may match a user's interests. It relies on the preferences, ratings, or interactions of users with similar behavior to identify relevant items for a target user. Because of its effectiveness, it is widely used in recommendation systems. Collaborative filtering is generally divided into two main types: Memory based and model-based. Memory-based collaborative recommendation process is generally divided into three main stages:

1. The system first gathers user preference data and preprocesses it to form a two-dimensional n × m user-item matrix *R*. In this matrix, n represents the number of users, m represents the number of items, and each entry r_{ij} indicates the rating given by user *i* to item *j*
2. In the user-item rating matrix, the ratings provided by each user across all items are used to compute the similarity between the target user and the other n−1 users. Based on these similarity values, users are ranked in descending order, and the top K most similar users are selected to form the neighbor set $N=\{n_1,n_2,n_3,....n_k\}$. A similar procedure is followed in item-based collaborative filtering, and similarity is measured between that item and the remaining m−1 items
3. From the list of nearest neighbors, a similarity-weighted average approach is applied to predict the rating values to construct rating prediction matrix. Let set of K nearest neighbors associated with user u is, $N=\{n_1,n_2,n_3,....n_k\}$ and the estimated rating $\hat{r}_{ui}$ is given by:

$$\hat{r} = \bar{r}_u + \frac{\sum_{v\varepsilon N_k} sim(u,v)(r_{vi}-\bar{r}_v)}{\sum_{v\varepsilon N_k(u)} sim(u,v)} \quad (1)$$

Where $\bar{r}_v$ is the average rating. $N_k(u)$ is the set of k users who are the nearest neighbors to user $u.$. $sim(u,v)$ is the similarity between user u and a neighbor user $v.r_{vi}$ v is the user v provided rating to item *i*. $\bar{r}_v$ is the mean rating user v.

The similarity between the target user u and a neighboring user can be computed using cosine similarity, Pearson correlation, Euclidean distance, or Manhattan distance. Cosine and Pearson directly measure similarity in rating behavior, whereas Euclidean and Manhattan measure distance and are usually converted into similarity weights so that closer users have greater influence on the predicted rating. After computing the rating prediction of unrated items, we can create a list of Top *N* items for recommendation. This is done by the unrated items by user *u* are arranged in order of decreasing rating based on the predicted ratings.

SVD Based Collaborative Algorithm

Model-based collaborative techniques apply mathematical models to learn latent features from the user-item rating matrix and represent users and items in a lower-dimensional space. These methods capture hidden patterns in the interaction data by extracting underlying factors from the matrix. A widely used model-based collaborative filtering approach is Singular Value Decomposition (SVD). SVD is a popularly used matrix factorization technique in for reducing the dimensionality of a information matrix while preserving important structural information. It captures hidden relationships by decomposing the original matrix into three factor matrices as $A \in R^{m \times n}$, where $A \in R^{m \times n}$, $U \in R^{m \times f}$, $\sum \in R^{f \times f}$, and $V \in R^{f \times n}$. In recommendation systems, the rating matrix is approximated using two low-rank matrices, P and Q. The matrix P represents user-feature interactions and is derived from the product $U\sum$, where $\sum$ is a diagonal matrix containing the singular values. The matrix Q represents item feature interactions and corresponds to V. The predicted rating for an unseen item is then estimated using the dot product of the latent representations in P and Q. Therefore, the matrix approximation and rating prediction are expressed as $R \approx PQ^T$.

Proposed Methods

The proposed Hybrid Cognition Enabled KNN Collaborative Filtering (HCE-KNNCF) is a recommendation approach which combines cognition-enabled KNN (CE-KNN) with SVD-based matrix factorization. The CE-KNN method enhances the traditional KNN approach by identifying the K nearest neighbors based on cognitive similarity. The SVD-based matrix factorization captures patterns that represent overall user behaviors and item characteristics. The model generates predictions by a combination of SVD-based matrix factorization and the enhanced KNN model using a weighted hybrid approach. This section describes the architecture and components of the proposed hybrid model. The detailed model structure of proposed hybrid model is illustrated in Fig. 1.

Problem Formulation

Let the user set and item set be denoted by $U = \{u1, u2, ...uM\}$, $I = \{i1, i2, ...iN\}$ where M and N denote the numbers of users and items, respectively. Let the observed feedback set be denoted by $R = \{(u, i, r_{ui}) | u \in U, i \in I\}$, where r_{ui} is the rating assigned by user u to item i. For rating prediction, the aim is to calculate the preference score r_{ui} for a given user-item pair.

User cognition is defined as the implicit information that represents a user's interaction-based preference toward items in a recommendation system. For a user u, cognition can be represented by a binary vector Cu, where each element indicates whether the user has interacted with an item within the recommendation domain. In this representation, a value of 1 denotes the presence of cognition information for an item, while 0 denotes its absence. Thus, user cognition provides an additional description of user preference patterns beyond the explicit ratings in the user-item matrix. This is represented as:

$$C_u = [Cu1, C_{u2}, ..., C_{um}] \; where \; c_{uj} = \begin{cases} 1, if \; user \; u \; has \; cognition \inf ormation \; for \; item \; j \\ 0, othercise \end{cases}$$

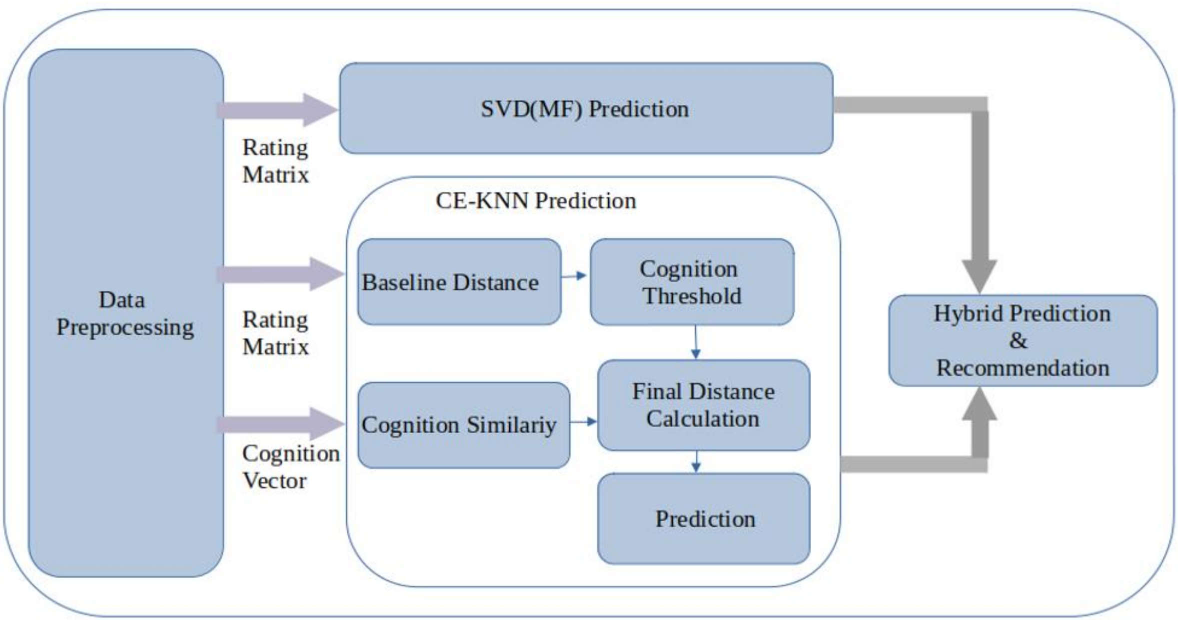

Fig. 1: Hybrid Cognition Enabled KNN Collaborative Filtering Framework

DOI: 10.3844/jcssp.2026.2769.2782

Cognition Enabled KNN (CE-KNN)

In the proposed approach, the conventional KNN-based collaborative filtering method is extended by incorporating a cognition vector into the user similarity computation. The neighbor selection process begins by calculating the baseline closeness between users from the user-item rating matrix using Euclidean distance. In parallel, user cognition is represented through cognition vectors, and the similarity between users' cognition patterns is measured using cosine similarity. To ensure consistency in the similarity computation, the Euclidean distance derived from the rating matrix is first transformed into a rating-based similarity score, which is then integrated with the cognition-based cosine similarity. This normalization ensures that the two measures are expressed on a common scale and can be integrated consistently in the final similarity computation.

Let $d_E(u,v)$ denote the Euclidean distance between users u and v computed from the user-item rating matrix. The normalized rating-based similarity is defined as:

$$s_E(u,v) = \frac{1}{1+d_E(u,v)} \tag{2}$$

Let C_u and C_v denote the cognition vectors of users u and v, respectively. The cognition-based similarity is computed using cosine similarity as:

$$S_{cog}(u,v) = \frac{Cu.Cv}{\|Cu\|\|Cv\|} \tag{3}$$

To determine whether cognition information should be incorporated, a cognition threshold is introduced. Since the original method derives this threshold from the average Euclidean measurements, a normalized threshold θ can be defined as:

$$\theta = \frac{1}{1+dE} \tag{4}$$

Where $\bar{d}_E$ is the mean Euclidean distance over the candidate neighbors. If the cognition similarity is lower than this threshold, only the normalized rating-based similarity is used. Otherwise, the final similarity is computed by combining the normalized Euclidean similarity and the cognition-based similarity through a cognitive weight α, where $0 \le \alpha \le 1$. The final similarity is therefore expressed as:

$$Sf(u,v) = \begin{cases} s_E(u,v), & if scog(u,v) < \theta \\ (1-a)s_E(u,v) + as_{cog}(u,v), & otherwise. \end{cases} \tag{5}$$

Here, α controls the contribution of cognition information to the final similarity value, while θ serves as a criterion for deciding whether cognition should influence neighbor selection. After computing $s_f(u,v)$, the K most similar users are selected as the nearest neighbors of the target user.

Finally, the predicted rating of user u for item i is obtained using a weighted aggregation of the ratings of the selected neighbors. Let $N_k(u)$ denote the set of K nearest neighbors of user u. The predicted rating can be calculated as:

$$r_{ui} = \bar{r}_u + \frac{\sum v \in N_k(u) sf(u,v)(r_{vi} - \bar{r}_v)}{\sum v \in N_k(u) sf(u,v)} \tag{6}$$

Where r_u and r_v denote the average ratings of users u and v, respectively, and $s_f(u,v)$ is the final similarity. This formulation enables the recommendation model to integrate both rating behavior and cognition patterns, thereby improving the accuracy of neighbor selection and rating prediction.

SVD Based Matrix Factorization

In the proposed hybrid framework, we have used regularized SVD component for modeling the latent relationships between users and items. This is a commonly adopted matrix factorization approach in recommender systems. In recommender systems, SVD factorizes the user-item matrix into low-dimensional user and item latent factors. And their interaction is modeled through a dot-product operation. The prediction for an unobserved user-item interaction is derived from the dot product between the associated user and item latent vectors. The regularized SVD framework applies SGD to iteratively update and optimize user and item latent factor matrices.

To model the rating deviations, user bias b_u and item bias bi are integrated into the prediction equation. These parameters are optimized jointly with the latent vectors during the SGD-based learning process. By considering this, we have computed the predicted ratings as:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_U^T qi \tag{7}$$

Where $\hat{r}_{ui}$ indicate predicted rating, μ denote global mean rating, bu and bi indicates user and item bias values, and p_u and q_i denote the latent factor vectors of user u and item i.

The SVD minimizes the following objective function for rating estimation:

$$\min \sum_{(u,i) \in D} (r_{ui} - \hat{r}_{ui})^2 + (\|pu\|^2 + \|qi\|^2 b_u^2 + b_i^2) \tag{8}$$

Where D denotes the set of observed user-item interactions, λ is the regularization parameter, $\hat{r}_{ui}$ indicate predicted rating r_{ui} denote the actual rating, b_u and b_i indicates user and item bias values, and p_u and q_i denote the latent factor vectors of user u and item i.

To optimize the bias parameters during training, the corresponding error term is calculated using the following expression:

$$e_{ui} = r_{ui} - \mu - b_u - b_i - p_u^T q_i \tag{9}$$

Where e_{ui} denote the error value, μ denote the global average rating, bu and bi implies user bias and item bias values, P_u is the user latent vector, q_i is the item latent vector, r_{ui} is the original rating.

We have used the stochastic gradient descent approach to optimize the results, and the parameters are updated as follows: The bias parameters for users and items are adjusted using Equations (10) and (11), as defined below:

$$b_i \leftarrow bu + \gamma(eui - \lambda bu) \tag{10}$$

$$b_i \leftarrow bi + \gamma(eui - \lambda bi) \tag{11}$$

Where b_u and b_i implies user bias and item bias values, e_{ui} denote the error value, γ indicate the learning rate and λ denote regularization parameter.

The latent feature vectors for users and items are updated using the equations given below:

$$p_u \leftarrow p_u + \gamma(e_{ui} . q_i - \lambda p_u) \tag{12}$$

$$q_i \leftarrow q_i + \gamma(e_{ui} \bullet p_u - \lambda qi) \tag{13}$$

Where P_u is the user latent vector, q_i is the item latent vector, e_{ui} denote the error value , λ indicate the learning rate and λ denote regularization parameter.

Hybrid Prediction and Recommendation

Finally, the hybrid prediction is formulated as a weighted linear combination of the enhanced KNN-based estimate and the matrix factorization-based score, where β determines the contribution of each component. The hybrid weighting factor β , which ranges between 0.1 to 0.9, to balance the contribution of each model. The hybrid weighting factor β controls the influence of the cognition-aware neighborhood method, while 1- β corresponds to the latent factor-based MF model. The final prediction equation can be formulated as:

$$\hat{r}_{ui}^{hyb} = \beta \hat{r}_{ui}^{CE-KNNCF} + (1\ \beta)\hat{r}_{ui}^{MF} \tag{14}$$

Where $\hat{r}_{ui}^{CE-KNNCF}$ represent enhanced KNN prediction rating, $\hat{r}_{ui}^{MF}$ denote SVD based MF prediction ratings and β is the hybrid weighted parameter.

Datasets

The Movie-Lens (Harper and Konstan, 2016) and Book-Crossing (Ziegler et al., 2005) are the two important benchmark datasets for evaluating recommendation system algorithms. The dataset Movie-Lens 100 k comprises 100,000 rating entries. This involves 944 users providing ratings for 1683 items. The evaluations are on a scale of 1 to 5, and every user reviewed a nearer to 20 films. The greater the rating leads to the more the user likes the movie or film. The Movie-Lens 1 M dataset contains over one million ratings (1,000,029) given by 6040 users to the 3952 movies, with a minimum of 20 ratings from each user. The Book-Crossing has a rating on scale of 1 to 10 with 271,379 books, 278,858 readers and and 1,149,780 ratings/interactions. Due to the large scale and sparsity of the Book-Crossing dataset, we filtered the data by removing implicit zero ratings and retaining only users and books with sufficient rating activity to ensure consistency in the experimental evaluation. The detailed structure of the datasets used in our evaluation is listed in Table 1.

Evaluation Parameters

To evaluate our proposed method, we have used two popular measures called mean absolute error and root mean squared error (Breese et al., 1998; Shani and Gunawardana, 2011). In both cases, lower values provide better model performance. The mean absolute error (MAE) measures the average absolute difference between the estimated and original ratings. On the other hand, Root Mean Squared Error (RMSE), captures determined by computing the square root of the mean of the squared deviations between each rating estimate and its related original rating (Breese et al., 1998; Shani and Gunawardana, 2011). MAE is mathematically defined as follows:

Table 1: Comparative evaluation of the proposed method against baseline models

Dataset	No. of Users	No. of Items	Ratings	Rating Scale
Move-Lens 100 K	943	1682	100000	1-5
Move-Lens 1 M	6040	3952	1000029	1-5
Book-Crossing	278858	271379	1149780	1-20

DOI: 10.3844/jcssp.2026.2769.2782

$$MAE = \frac{1}{n}\sum_{u,i}\left|r_{ui} - \hat{r}_{ui}\right| \quad (15)$$

RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(r_{ui} -)^2} \quad (16)$$

Where n is the aggregated count of ratings, r_{ui} is the original rating, and $\hat{r}_{ui}$ is the rating estimate.

Experimentation Setting

Experiments were implemented in Python 3.9.15 using Jupyter Notebook through Anaconda Navigator 2.4.3. NumPy, Pandas, and Scikit-learn were employed for data pre-processing and implementation, and the implementation of the proposed method was carried out with the Scikit-Surprise library (Hug, 2020). A five-fold cross-validation approach was employed, with a data of 80% for the training, while the 20% is set aside for the testing purpose. The regularized SVD component was tuned by varying the number of latent factors from 10 to 100, the regularization parameter from 0.1 to 0.5, and the learning rate from 0.001 to 0.005. The final parameter values were selected based on hyper parameter tuning using the best validation performance. We established the quantity of neighbors in the cognition-enabled KNN part to 30. The cognitive weight is assumed here as 0.5 for giving priority to genuine user interaction. For implementation purposes we have created the cognition vector derived from implicit feedback information. This is done by assigning a binary value of 1 to any item a user has rated, regardless of the rating score.

We also evaluated the proposed recommendation model under cold-start conditions for both users and items. To provide a comprehensive assessment, the cold-start analysis was conducted in two settings: User cold-start and item cold-start. In the user cold-start setting, the objective was to measure how effectively the model predicts ratings for users with very limited historical interaction data. For this purpose, users having only a small number of ratings were identified as cold-start users. A limited subset of their ratings was retained in the training set to simulate sparse user history, while the remaining ratings were withheld for testing. This setting allows us to examine whether the proposed model can generate reliable predictions even when user preference information is highly incomplete. In the item cold-start setting, the purpose was to evaluate the model's ability to recommend items that have not received prior ratings during training. To simulate this scenario, approximately 5% of the items were randomly selected and completely removed from the training set so that no rating information for these items was available during model learning. These excluded items were then introduced only in the test phase.

Baseline Method Used

The proposed HCE-KNNCF model was evaluated against several baseline recommendation methods, including USER-KNN and ITEM-KNN (Desrosiers and Karypis, 2011), SVD-based matrix factorization (Koren et al., 2009), ExtKNNCF (Nguyen et al., 2023), HCFMR (Behera et al., 2024), and AMeLU (Liu et al., 2023). These baselines were cover neighborhood-based, latent factor-based, and hybrid recommendation paradigms. A brief description of each baseline method is listed below:

- USER-KNN (Desrosiers and Karypis, 2011): User-based KNN is a memory-based collaborative filtering approach in which each user is represented by a vector of item ratings. Similarity between users is computed using measures such as Pearson correlation, cosine similarity, or adjusted cosine similarity, and the ratings of the nearest neighbors are aggregated to predict unknown ratings and generate recommendations
- ITEM-KNN (Desrosiers and Karypis, 2011): Item-based KNN is a memory-based collaborative filtering approach in which items are represented as rating vectors in the user-item space. Similarity between items is computed using measures such as Pearson correlation, cosine similarity, or adjusted cosine similarity, and unknown ratings are predicted from the preferences associated with the k most similar items. Recommendations are then generated based on these predicted ratings
- SVD (MF) (Koren et al., 2009): SVD is a latent factor recommendation model that approximates the sparse user-item interaction matrix by decomposing it into lower-dimensional representations of users and items. By learning these latent features, the model captures underlying preference structure more effectively and predicts missing ratings with improved accuracy
- ExtKNNCF (Nguyen et al., 2023): ExtKNNCF is an adaptive neighborhood-based recommendation model that extends the traditional KNN framework by incorporating user cognition into the similarity computation process. Unlike standard KNN methods, which rely primarily on rating-based similarity, ExtKNNCF refines neighbor selection through cognition-aware similarity adjustment, thereby improving rating prediction accuracy
- HCFMR (Behera et al., 2024): HCFMR is a hybrid collaborative filtering model for movie recommendation that combines matrix factorization with XGBoost. It uses matrix factorization to capture latent user-item interactions and then employs XGBoost to improve final rating prediction performance
- AMeLU (Liu et al., 2023): Attentional Meta-Learned User Preference Estimator is a cold-start

recommendation model that combines meta-learning with an attention mechanism to estimate user preferences from limited interaction data. The attention component helps model different types of user interests more effectively

Results and Discussion

Table 2 describe the analysis of the proposed HCE-KNNCF model with the baseline methods on the MovieLens 100 K, MovieLens 1 M, and Book-Crossing datasets. The results indicate that HCE-KNNCF achieves the best performance on MovieLens 100 K and Book-Crossing, where it records the lowest prediction errors among all compared methods. The model attains an MAE of 0.6765 and an RMSE of 0.8487 on MovieLens 100 K, and an MAE of 1.0451 and an RMSE of 1.3671 on Book-Crossing, suggesting that the proposed hybridization of cognition-enhanced neighborhood learning and latent factor modeling is effective under sparse rating conditions. Compared with the conventional neighborhood-based baselines, like Item-KNN and User-KNN, the proposed model attains improvements across all datasets. This indicates that incorporating latent factor modeling together with cognition-aware neighbor refinement provides an advantage over traditional memory-based collaborative filtering. The performance is also evident on the Book-Crossing dataset, where the large scale and sparsity make accurate neighbor selection more challenging for standard KNN approaches.

Compared with SVD and the hybrid models ExtKNNCF, HCFMR, and AMeLU, the proposed HCE-KNNCF model shows good performance. The strongest gains are observed against ExtKNNCF and AMeLU, which suggests that the proposed cognition-enhanced fusion strategy improves the relevance of selected neighbors while still benefiting from latent user-item interactions. However, on MovieLens 1 M, HCFMR achieves lower MAE and RMSE than HCE-KNNCF, indicating that the proposed method does not uniformly outperform all competing hybrid models across every dataset. It is also observed that the small standard deviation values reported for HCE-KNNCF across all datasets. These results show that the proposed model shows stable performance over repeated runs and is less sensitive to experimental variation.

Statistical Significance and Interpretation

We have used the paired t-test, a parametric statistical test to compare the means of two related samples, to determine whether the observed performance improvements of the proposed HCE-KNNCF model over the competing baseline methods are statistically significant. In the present study, the paired t-test was applied to the repeated MAE and RMSE values obtained for the proposed model and each selected baseline under identical experimental settings. For our experimental setting, the paired t-test is appropriate because, it is designed to compare two related sets of observations obtained under matched conditions. It evaluates whether the mean difference between paired results is significantly different from zero (Rainio et al., 2024). And it is suitable for continuous evaluation metrics such as MAE and RMSE computed across repeated runs. For the statistical significance analysis, paired samples were constructed by matching the repeated experimental results of the proposed HCE-KNNCF model with those of each competitive baseline obtained under the same run settings. The baselines selected for comparison were the strongest competing methods identified from the experimental results, they are AMeLU, ExtKNNCF, and HCFMR. The null hypothesis assumes no significant difference, while the alternative hypothesis assumes that the proposed model performs significantly better at the 0.05 significance level. As seen in Table 3, all p-values are below the threshold of 0.05, and this shows that HCE-KNNCF significantly outperforms each of the compared models in terms of rating prediction performance. The statistically significant results confirm that the improvements offered by the proposed model are not due to random variations.

Table 2: Comparative evaluation of the proposed method against baseline models

Method	Movie-Lens 100K		Movie-Lens 1M		Book-Crossing	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Item-KNN	0.7631	0.9500	0.7585	0.9418	1.4193	1.8120
User-KNN	0.7454	0.9460	0.7487	0.9713	1.3521	1.7016
SVD(MF)	0.7146	0.9076	0.7185	0.9020	1.1714	1.6238
AMeLU	0.7277	0.8802	0.7068	0.8756	1.2307	1.5318
ExtKNNCF	0.7240	0.8530	0.7278	0.8634	1.0563	1.4762
HCFMR	0.6970	0.8750	0.6891	0.8210	1.0517	1.4306
HCE-KNNCF	0.6765	0.8487	0.6734	0.8347	1.0451	1.3671

Table 3: Comparison of HCE-KNNCF with baseline models paired t-test p-value

Baseline Models	Movie-Lens 100K MAE	Movie-Lens 100K RMSE	Movie-Lens 1M MAE	Movie-Lens 1M RMSE
AMeLU	0.00058	0.00003	0.00019	0.00069
ExtKNNCF	0.00004	0.00031	0.00002	0.00056
HCFMR	0.00002	0.00014	0.00009	0.00051

Experiments were also conducted to analyze the impact of the number of recommendations on the proposed model compared with other benchmark algorithms. To assess the scalability, the number of recommendations varied across the set {10, 20, 30, 40, 50, 60, 70, 80}. This evaluation examines how the algorithms respond to different recommendation list sizes and how data sparsity influences their predictive stability. Beyond 40 we can see that these models experience an increase in MAE and RMSE values. Figs. 2, 3 shows the evaluation of the performance of our suggested work with benchmark models on Movie-Lens 1 M dataset. Figs. 4,5 shows the performance with Book-Crossing dataset. This evaluation examines how different algorithms respond to increasing recommendation list sizes and how data sparsity affects their predictive stability. The results show that the model maintain relatively stable performance up to 40 recommendations.

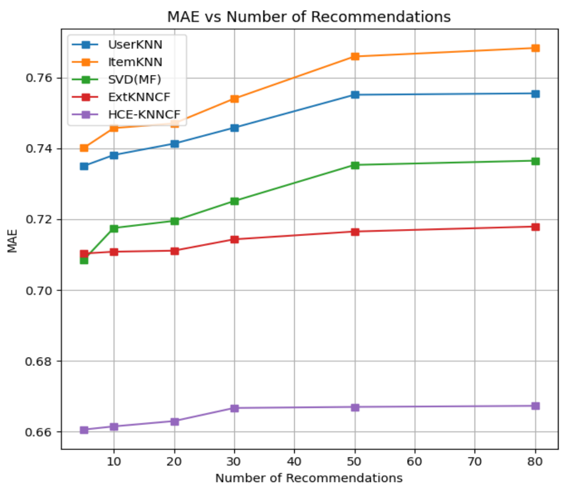

Fig. 2: Comparison of MAE with other models Move-Lens 1^{M}

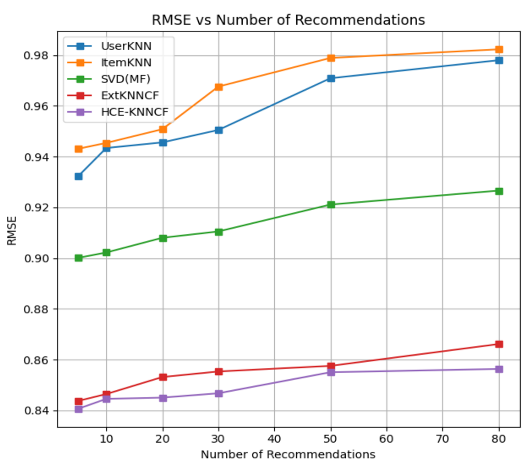

Fig. 3: Comparison of RMSE with other models Move-Lens 1 M

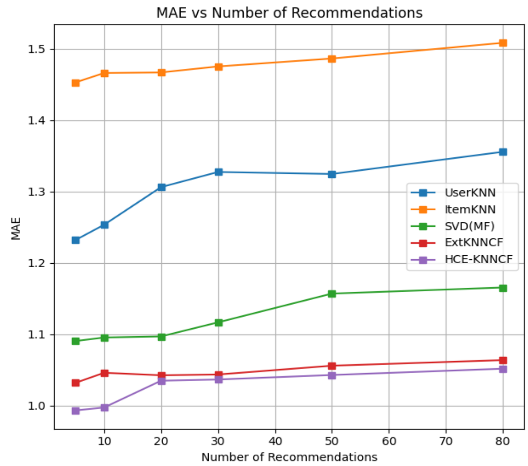

Fig. 4: Comparison of MAE with other models Book-Crossing

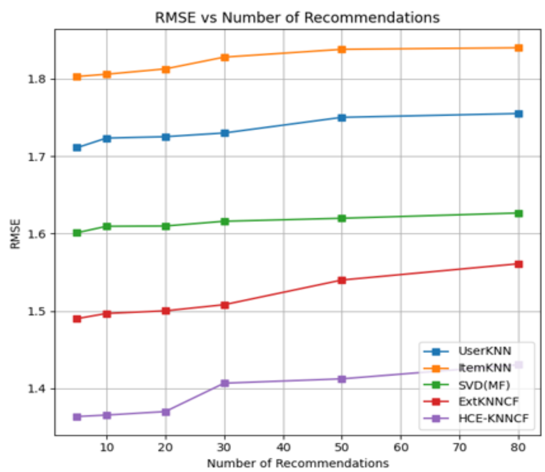

Fig. 5: Comparison of RMSE with other models Book-Crossing

This shows that higher recommendation list sizes may introduce less relevant items, and increasing prediction error, particularly under sparse data conditions. In addition, we examined the impact of the hybrid weighting factor β on the performance of the proposed recommendation framework. A set of experiments was conducted by varying the weighting factor from 0.1 to 0.9 to identify the optimal combination. The best performance is achieved when the Matrix Factorization (MF) component contributes 60% and the cognition-enabled KNN component contributes 40% to the final prediction. This configuration allows the model to capture global latent preference patterns through MF while simultaneously preserving local neighborhood-level impact with enhanced CE-KNN. Figs. 6(a-c), illustrate the effect of the hybrid weighting factor on the proposed model across the Movie-Lens 100 K, Movie-Lens 1 M and Book-Crossing datasets.

DOI: 10.3844/jcssp.2026.2769.2782

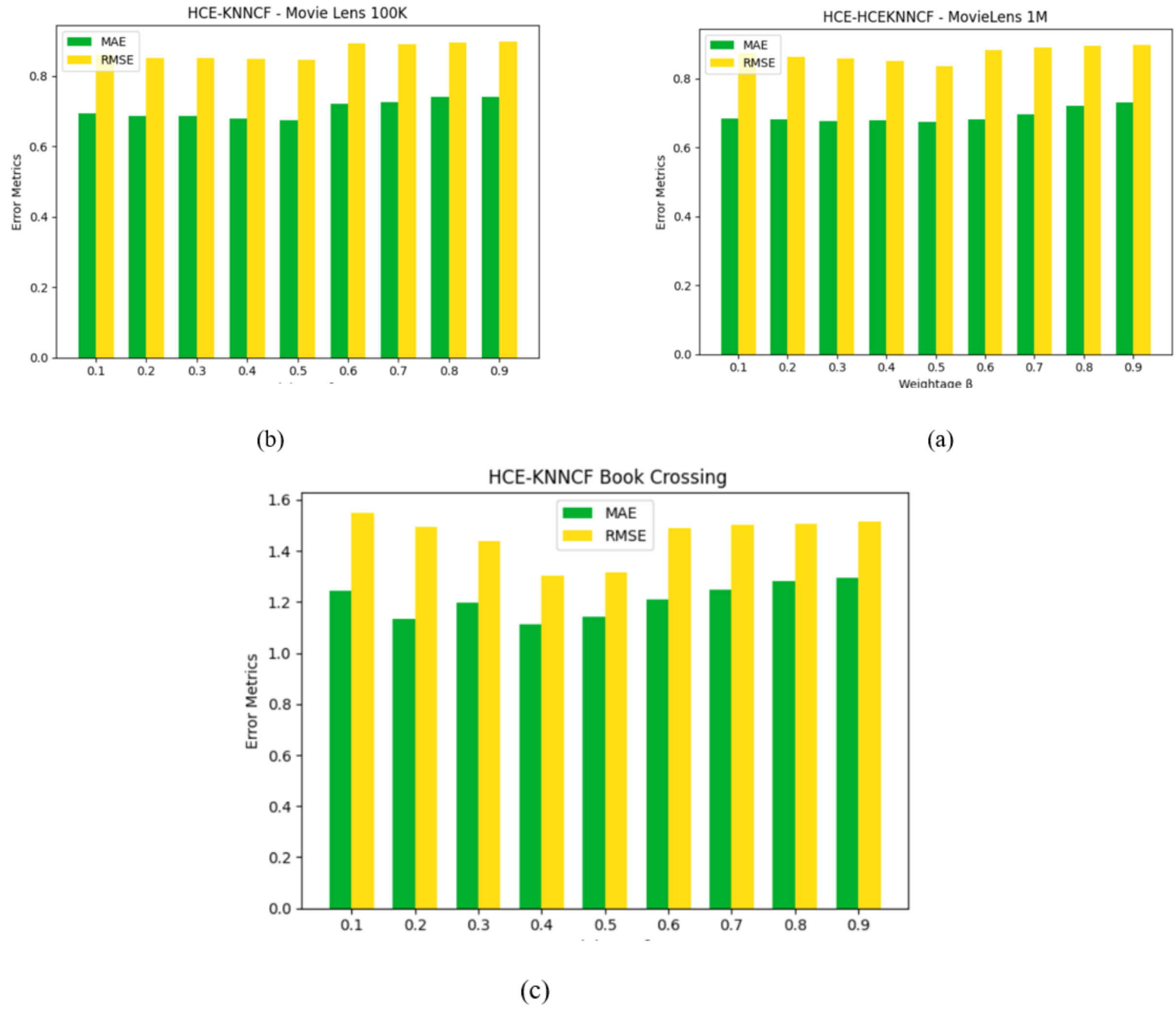

Fig. 6:(a) Impact of the hybrid weighting factor over our proposed model on Movie-Lens 100 K, (b) Movie-Lens 1 M and (c) Book-Crossing datasets

We have conducted experiments to evaluate the proposed model's capability in handling cold-start scenarios. These experiments were conducted to determine the ability of the model to deliver personalized suggestions when new item and user, appears in the system. To evaluate performance in cold-start settings, separate experiments were performed for user and item cold-start cases. It has been observed that the model maintains lower error rates across both scenarios, with MAE/RMSE values of 0.7148/0.9258 and 0.7749/0.9857 on MovieLens 1 M, and 1.1601/1.4678 and 1.2154/1.5966 on Book-Crossing. Moreover, these cold-start results are still better than those of several competing baseline methods, in non-cold scenario especially baselines USER-KNN, ITEM-KNN, SVD and hybrid models AmeLU and ExtKNNCF. This demonstrates that the proposed approach remains robust and effective even when interaction information is limited.

Observation on Model Limitations

During analysis of the performance of our hybrid HCE-KNNCF model, we observed an important limitation related to the dependence on the user-item rating matrix. When only explicit rating data are available, the system is unable to distinguish whether a missing entry arises from a general data sparsity problem or a genuine cold-start situation. Both scenarios appear structurally identical in the rating matrix, resulting in the model treating them as the same type of missing information. Both issues appear as missing entries in the matrix, and the system lacks a mechanism to determine whether an unrated item is simply due to naturally sparse user behavior or because the new user or item with no interactions data. The traditional collaborative filtering methods treat all missing values uniformly, even though the underlying reasons are different. This creates challenges like CE-KNN component could not compute meaningful cognition vectors for entirely new users or items, and the SVD component cannot learn latent factors without sufficient interaction history. Our hybrid approach improves the similarity computation, neighbor selection, and prediction accuracy under sparse conditions, it still struggles to generate fully reliable recommendations for users and items that have no historical activity.

This observation points out the opportunity for future enhancement. The model can be extended by incorporating additional sources of information such as additional attributes, item metadata, temporal data, or social media auxiliary data when interaction data is insufficient. Such side information would enable the system to differentiate between sparsity and cold start, initialize new user/item representations more effectively, and provide personalized recommendations along with ratings. Furthermore, integrating contextual or semantic features into the cognition component could enrich neighbor identification and make similarity adaptation more robust. Another promising direction is the use of non-Euclidean space representation and measures for similarity computation, as they may better capture complex and non-linear relationships in recommendation systems.

Conclusion

Our study focused on a hybrid recommendation framework model designed to overcome the major limitations of collaborative filtering, particularly the challenge of insufficient data, cold-start problems, and lack of initial interaction history. The proposed framework integrates SVD-based matrix factorization with a cognition-enabled KNN model, using a hybrid method that balances global latent interactions and local neighborhood similarities. Experiments conducted on the Movie-Lens 100 K, Movie-Lens 1 M, and Book-Crossing datasets demonstrated that HCE-KNNCF achieves better accuracy compared to traditional as well as hybrid benchmark algorithms. In the non-cold-start evaluation, HCE-KNNCF obtains the best MAE and RMSE values on the MovieLens 100 K dataset, with 0.6765 and 0.8487, respectively. The approach demonstrates the lowest error on the Book-Crossing dataset, with an MAE of 1.0451 and an RMSE of 1.3671, indicating its effectiveness in sparse data conditions. For the MovieLens 1 M dataset, HCE-KNNCF remains competitive, here the HCFMR shows slightly better performance on this dataset. The cold-start evaluation further confirms the robustness of the proposed model. For the MovieLens 1 M dataset, HCE-KNNCF achieves the lowest MAE and RMSE in both cold-start scenarios that is recommending items for new users, with MAE of 0.7149 and RMSE of 0.9258. While recommending new items to existing users, with MAE of 0.7749 and RMSE of 0.9857. Similarly, on the Book-Crossing dataset, the proposed model outperforms the baseline methods in both scenarios, achieving MAE of 1.1601 and RMSE of 1.4678 for new users, and MAE of 1.2154 and RMSE of 1.5966 for new items. The results demonstrate that the proposed HCE-KNNCF model improves recommendation accuracy and maintains a stable performance under both normal and cold-start conditions. The proposed model can be further extended by incorporating additional information sources, such as item metadata, temporal data, and social media-based auxiliary data, along with rating, rather than depending entirely on interaction data. In addition, integrating contextual or metadata-based side information represents a promising direction for improving recommendation performance. Our future work will therefore focus on incorporating such auxiliary information and exploring non-Euclidean similarity representation and measures to further enhance similarity computation and recommendation quality.

Acknowledgment

The authors acknowledge the institutional support provided during the course of this research.

Funding Information

The authors declare that they have not received any funding or financial support to report this study.

Author's Contributions

All authors contributed equally to this work.

Ethics

The manuscript is an original work. The authors declare that no ethical concerns are associated with the submission of this work.

References

Bahrani, P., Minaei-Bidgoli, B., Parvin, H., Mirzarezaee, M., & Keshavarz, A. (2024). A new improved KNN-based recommender system. *The Journal of Supercomputing*, *80*(1), 800–834. https://doi.org/10.1007/s11227-023-05447-1

Behera, G., Panda, S. K., Hsieh, M.-Y., & Li, K.-C. (2024). Hybrid collaborative filtering using matrix factorization and XGBoost for movie recommendation. *Computer Standards & Interfaces*, *90*, 103847. https://doi.org/10.1016/j.csi.2024.103847

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98*, 43–52. https://doi.org/10.48550/arXiv.1301.7363

Chen, R., Hua, Q., Chang, Y.-S., Wang, B., Zhang, L., & Kong, X. (2018). A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access*, *6*, 64301–64320. https://doi.org/10.1109/access.2018.2877208

Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. *Recommender Systems Handbook*, 107–144. https://doi.org/10.1007/978-0-387-85820-3_4

Guan, J., Chen, B., & Yu, S. (2024). A hybrid similarity model for mitigating the cold-start problem of collaborative filtering in sparse data. *Expert Systems with Applications*, *249*, 123700. https://doi.org/10.1016/j.eswa.2024.123700

Harper, F. M., & Konstan, J. A. (2016). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, *5*(4), 1–19. https://doi.org/10.1145/2827872

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. https://doi.org/10.21105/joss.02174

Koren, Y. (2009). Collaborative filtering with temporal dynamics. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 447–456. https://doi.org/10.1145/1557019.1557072

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/mc.2009.263

Liang, A., Bai, Y., Wu, M., Wu, J., & Wu, G. (2024). Research on Personalized Recommendation Algorithms for Different User Behaviors. *Proceedings of the 2024 4th International Conference on Big Data, Artificial Intelligence and Risk Management*, 163–169. https://doi.org/10.1145/3718751.3718779

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. https://doi.org/10.1109/mic.2003.1167344

Liu, S., Liu, Y., Zhang, X., Xu, C., He, J., & Qi, Y. (2023). Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning. *Electronics*, *12*(2), 376. https://doi.org/10.3390/electronics12020376

Lops, P., Jannach, D., Musto, C., Bogers, T., & Koolen, M. (2019). Trends in content-based recommendation. *ser Modeling and User-Adapted Interaction*, *29*(2), 239–249. https://doi.org/10.1007/s11257-019-09231-w

Lv, S., Wang, J., Deng, F., & Yan, P. (2024). A hybrid recommendation algorithm based on user nearest neighbor model. *Scientific Reports*, *14*(1), 17119. https://doi.org/10.1038/s41598-024-66393-3

Nguyen, L. V., Vo, Q.-T., & Nguyen, T.-H. (2023). Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. *Big Data and Cognitive Computing*, *7*(2), 106. https://doi.org/10.3390/bdcc7020106

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation Metrics and Statistical Tests for Machine Learning. *Scientific Reports*, *14*(1), 6086. https://doi.org/10.1038/s41598-024-56706-x

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*, 59. https://doi.org/10.1186/s40537-022-00592-5

Roy, T., Shetty, P. D., & Seal, S. (2025). A Hybrid Approach to Predict Ratings for Book Recommendation System Using GNN. *2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 1–6. https://doi.org/10.1109/csitss67709.2025.11294052

Sami, A., Adrousy, W. E., Sarhan, S., & Elmougy, S. (2024). A deep learning based hybrid recommendation model for internet users. *Scientific Reports*, *14*(1), 29390. https://doi.org/10.1038/s41598-024-79011-z

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web*, 285–295. https://doi.org/10.1145/371920.372071

Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. *Recommender Systems Handbook*, 257–297. https://doi.org/10.1007/978-0-387-85820-3_8

Sharma, S., & Shakya, H. K. (2024). Hybrid recommendation system for movies using artificial neural network. *Expert Systems with Applications*, *258*, 125194. https://doi.org/10.1016/j.eswa.2024.125194

Singh, K., Dhawan, S., & Bali, N. (2024). An Ensemble Learning Hybrid Recommendation System Using Content-Based, Collaborative Filtering, Supervised Learning and Boosting Algorithms. *Automatic Control and Computer Sciences*, *58*(5), 491–505. https://doi.org/10.3103/s0146411624700615

Sinha, B. B., & Dhanalakshmi, R. (2019). Evolution of recommender system over the time. *Soft Computing*, *23*(23), 12169–12188. https://doi.org/10.1007/s00500-019-04143-8

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, *2009*, 1–19. https://doi.org/10.1155/2009/421425

Sun, Y., & Liu, Q. (2025). Collaborative filtering recommendation based on K-nearest neighbor and non-negative matrix factorization algorithm. *The Journal of Supercomputing*, *81*(1), 79. https://doi.org/10.1007/s11227-024-06537-4

Wenga, C., Fansi, M., Chabrier, S., Mari, J.-M., & Gabillon, Alban. (2021). A comprehensive review on non-neural networks collaborative filtering recommendation systems. *ArXiv:2106.10679*. https://doi.org/10.48550/arXiv.2106.10679

Zhou, X., He, J., Huang, G., & Zhang, Y. (2015). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, *81*(4), 717–733. https://doi.org/10.1016/j.jcss.2014.11.016

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. *Proceedings of the 14th International Conference on World Wide Web - WWW ’05*, 22–32. https://doi.org/10.1145/1060745.1060754