A hybrid recommender system integrating sentiment analysis and link prediction to mitigate cold start and sparsity in social networks

Nafiseh Fareghzadeh [a*], Mahdi Bazargani [b]

a. PhD, Department of Computer Engineering, Khod.c., Islamic Azad University, Khodabandeh, Iran
b. PhD, Department of Computer Engineering, Za.C., Islamic Azad University, Zanjan, Iran
* Corresponding author: fareghzadeh@iau.ac.ir (N.Fareghzadeh)

Keywords

Hybrid recommender system;
Sentiment analysis;
Collaborative method;
Content-based method;
Link prediction.

Abstract

Despite all success of recommender systems, these systems face natural problems such as cold start and data sparseness. To face these issues, in this work a novel hybrid recommender is proposed that leverages the strengths of both collaborative and content-based methods together with sentiment analysis. The proposed system improves recommendation accuracy, alleviates cold start and sparsity issues, predicts future interactions, and adds a new context layer to recommendations, by effectively integrating demographic data, user clustering, Machine Learning (ML) models, sentiment analysis, and improved link prediction strategy. Proposed system was evaluated using the popular MovieLens dataset, and test results indicate the system significantly improves handling new users. The proposed system achieved a decline of 5.2% in Mean Absolute Error (MAE) and 6.8% in Root Mean Square Error (RMSE), compared to baseline methods. Moreover, results indicate that the proposed recommender performs significantly better under user and item cold start conditions, with a reduction of 29.6% and 27.4% in MAE, respectively. Lastly, integrating sentiment analysis and an improved link prediction approach provides a boost of 7% in Precision and 10.8% in Recall. Therefore, such hybrid approaches can be successful in alleviating cold start and sparsity problems, and enhancing overall system performance.

1. Introduction

Artificial intelligence is an important research field applied to problems of almost all real-life fields. Nowadays, it has become apparent that the need for intelligent solution approaches is a vital factor in overcoming real-world problems effectively [1]. Recently, recommender applications become an integral part of electronic service platforms, across various domains such as e-commerce, streaming services, and social networks. These systems have recommendation engines that analyzes collected data such as purchase biography, browsing history, demographics, and contextual information to deliver suggestions [2].

In order to make effective recommendations, recommender systems have different types and use different filtering methods [3]. There are mainly three major filtering methods for recommender systems as follow [4, 5]:

- Collaborative Filtering (CF): This method operates by evaluating user interactions and determining similarities between users and items. This approach faces issues like the cold start problem, sparsity, scalability problems, and popularity bias.
- Content-Based Filtering (CBF): This is a technique used to suggest items that are comparable with an item a user has shown interested in, based on the target attributes of items. One major disadvantage of this method is feature limitation. Moreover, some other weakness are low serendipity, overspecialization, lack of popularity of items [6].
- Hybrid Method: Hybrid approach can combines CF, CBF methods with other recommending approaches to overcome drawbacks of each individual method, leverage the strengths of each approach, and result in more accurate and diversified recommendations.

While these collaborative and content-based filtering methods have proven efficiency in many applications, they are not without their challenges. The cold start problem, where limited information about new users or items, hinders the system's ability to produce meaningful recommendations, remains a significant barrier [7]. Despite the significant advances in recommendation systems, a considerable gap persists, regarding the effective solution for common challenges such as cold start problem, sparsity, scalability and performance, particularly in the context of social networks [8].

On the other side, sentiment analysis is an intelligent capability, whose application in recommender systems can lead to higher performance and accuracy in suggestions and unfortunately, most existing recommenders lack this capability. Normally, the goal of the typical sentiment analysis based recommender system is to extract or analyze user's emotions, opinions, and other information regarding a certain object, and use this information to achieve more personalized and accurate suggestions [9].

In order to face the weaknesses of traditional methods, recent recommender systems have been moved toward hybrid approaches. However, still there is a lack of advanced and intelligent capabilities such as sentiment analysis in suggestions, and also lack of research that adequately explore how machine learning techniques and demographic data can further improve the performance of hybrid systems in social networks. This research aims to fill this gap by developing a novel hybrid recommender system that have major contributions as follows:

- Integrating sentiment analysis capability with CF and CBF methods to design a novel sentiment-enhanced hybrid recommender system.
- Applying user demographic information, historical data, and interaction feedbacks to improve accuracy and overall system performance.
- Utilizing the content information of items to enhance recommendation quality, and reducing the effects of the data sparsity and cold-start problems.
- Integrating user clustering, learning models, sentiment analysis, and improved link prediction strategy to predict future user interactions, provide an extra layer of context to recommendations, and optimization of suggestions.

The rest of the paper is structured as follows: Section 2 provides a comprehensive review and comparison on the research literature. Section 3 describes the proposed hybrid recommender system and its dimensions. Section 4 presents the evaluation of the proposed system. In Section 5, the results of the experiments are discussed. Finally, Section 6 concludes the article by summarizing the key findings, outlining the contributions of the research, and suggesting directions for future work.

2. Related Work

In this section, we present a detailed review on the research literature and related studies. The goal of most recommender systems is to personalize the user experience by providing highly relevant and useful recommendations. The majority of recent recommenders fit into three fundamental type with filtering methods include: collaborative, content-based, and hybrid methods.

Authors in [10] engage two CF algorithm with different attributes. CF that use the rating as attribute and CF that use the user-preferred genres as attribute. However, challenges of using only CF method was a limitation for their implementation. Research [11] used ML algorithms to design online education courses recommender system. However, not enough consideration has been given to sentiment analysis and user feedbacks on recommendations. In [12] CF and sentimental aspects have been integrated into the hybrid recommender approach. However, CF recommender still encounters difficulties such as the cold-start problem and the gray sheep problem. Ali et al. in [13] proposed a hybrid model where genomic tags of the movie were used with CBF to recommend movies with similar tastes. In addition, hybrid techniques attenuate the deficiency of individual strategies due to the blend of both recommendation strategies. Research [14] propose a hybrid music recommendation system that combines CBF and CF methods. Authors in [15] present a hybrid approach combining CF and demographic data in social networks. Sentiment analysis is one of the most attractive research areas in the recent recommender systems [16]. In [9], authors proposed a recommendation system model that combined sentiment analysis of a user's social media data with collaborative filtering to achieve better accuracy. The most important achievements and challenges of related research can be summarized according to the information in Table 1. This research aims to fill the research gap by developing a system that integrates CF, CBF, sentiment analysis, user clustering, and predictive algorithms to enhance recommendation accuracy and solve the cold start problem and sparsity, more effectively. In the next section, different dimensions of the proposed recommender system has been described, in detail.

3. Proposed recommender system

In this section, we describe the important dimensions of the proposed system, and major workflows. Figure 1 presents the proposed system framework. The functional details of system components, executive phases and workflows in the proposed recommender engine are as follows:

3.1. Content-based component

After loading the data into the system, the first component of the proposed recommender system is executed. In order to classify new users, the content based component supports preprocessing and extended user clustering for the initial clustering of users' dataset according to demographic information. The profiles which are generated will allow the system to link users with similar items. In this phase, all users are clustered based on demographic information such as age, occupation and gender, and assigned to categories.

3.1.1 Preprocessing

After the dataset is entered into the system, the desired data is preprocessed, outliers and unused samples are removed from it. Then, the data that is converted into a coherent form and acceptable format for simulation. Considering the needs of our problem, in this research, we used RapidMiner data mining tool and data cleaning method for preprocessing. The strategy is to analyze the data and identify if any row or column has null or unused values. Then, values examined before and after the sample that has a null or unused value, and calculate their average. Finally, null values have been replaced with the obtained averages. Therefore, outliers eliminated and more coherent data is produced.

Note that, during the preprocessing phase, the mean imputation approach was used to handle missing values. This choice was made because, the impacted variables were continuous numerical features rather than categorical ones, and because the missing instances were comparatively middle class and randomly distributed. Our choice balances practical constraints and acceptable accuracy in imputations without introducing significant computational overhead. Therefore, given the numerical nature of the variables and the dispersion of incomplete data, replacing with the mean maintains data consistency and improves the accuracy of learning models in later stages of development process.

3.1.2 User clustering

In order to classify new users, this phase utilizes CBF method and clustering is used for the initial categorizing user's dataset, according to demographic information. By organizing users into meaningful clusters, the recommendation system can handle sparsity and improve recommendation performance. In this phase, all users are first clustered based on the demographic information such as age, occupation, and gender, and preferences using the extended clustering algorithm. For all users in the system, a label is specified as a cluster. First, the sample size is entered into the clustering algorithm after applying the execution process and data cleaning steps to obtain the optimal K value, and in the next steps, the extended X-Means clustering algorithm is used with the optimal K value for clustering. To increase

the speed of the process, the optimal K value is determined online on the central system server. It must be noted that, the K-Means algorithm is a basic clustering algorithm that performs the clustering process of samples based on a number of clusters called K. One of the most important drawbacks of this algorithm is that the number of clusters must be determined by the researcher or programmer and the clustering process should be performed based on this number. Determining the K value had a high error and often did not provide the desired and correct clustering.

Moreover, to assess the efficacy of the extended X-Means clustering algorithm, we conducted exploratory analysis, using two contrasting methods as alternate clustering approaches, clustering based on density method, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical agglomerative clustering. While DBSCAN has the capability to identify arbitrary-shaped clusters, it proved to be highly sensitive to its parameters (ε *and MinPts*), and ultimately struggled with the variability in density, with respect to user demographic data. Hierarchical clustering provided interpretable results, but it was limited by high computational costs, and scalability for larger samples. Therefore, we found that X-Means method was viable, allowing for automatic determination of the optimal cluster number by minimizing BIC. In fact, empirical results suggested that X-Means produced more balanced clusters, and showed a downstream recommendation accuracy improvement, when compared with hierarchical clustering. For these reasons, the extended X-Means algorithm was proposed. Extended clustering algorithm automatically determines the optimal number of clusters, by minimizing the Bayesian Information Criterion (BIC). This method allows the system to group users into different clusters with similar demographic profiles, which helps to make better recommendations in the cold start phase [25, 26]. The extended algorithm formulated as follows:

Extended user clustering algorithm

Input: *S:* Set of user data samples*, C:* Initial cluster centroids*, τ:* Threshold for minimum distance;
Output: C_{List}*:* Final list of optimized clusters;

1. Initialize *C* with one centroid selected randomly from *S*.
2. For Each Sample $s \in S$ do:
3. For each centroid $c \in C$ do:
4. *TempDist = Direction Evaluation(s, c.direction) + Euclidean Distance(s, c);*
5. end for
6. *MinDistance = min(TempDist);*
7. *ClosestCentroid = centroid with MinDistance;*
8. If *MinDistance* $\leq \tau$ then
9. Assign s to Cluster(*ClosestCentroid*);
10. Update *Centroid* of the cluster;
11. else
12. Create a new cluster with centroid = *s*;
13. Add this new centroid to *C*;
14. end if
15. end for
16. Compute Bayesian Information Criterion (*BIC*) for each cluster split:
17. *BIC = L - (P/2) * log(N);*
18. where:
19. *L = likelihood of the model;*
20. *P = number of parameters;*
21. *N = total number of samples*;
22. If BIC improves (i.e., decreases) then split the cluster;
23. else stop the clustering process;
24. Output the final list of clusters $C_{List.}$;

Algorithm 1. The pseudo code of the extended user clustering algorithm

Given a set of *N* data points $\{x_1, x_2, \ldots, x_N\}$, where each point is a vector in R_d set, the objective is to partition these points into *K* clusters, where *K* is automatically determined by the above algorithm. The algorithm minimizes the within-cluster sum of squares (WCSS), defined as:

$$WCSS = \sum_{k=1}^{k} \sum_{x \dot{\in} c_k} x - \mu_k^{\,2} \qquad (1)$$

Where C_k is the set of points in cluster k, and μ_k is the centroid of cluster k.

Extended user clustering algorithm iteratively splits clusters and computes the *BIC* score for each split to evaluate whether the additional cluster improves the model. The *BIC* score is given by:

$$BIC = L - \frac{P}{2} .log(N) \quad (2)$$

Where:

- L is the likelihood of the model.
- P is the number of parameters in the model (proportional to the number of clusters).
- N is the number of data points.

The algorithm stops when the *BIC* score no longer improves with additional clusters. This ensures that the number of clusters is optimal and not over fitted. After the extended user clustering algorithm is applied to the data about users based on their personal information, a combination of machine learning methods is used to select a category for the new user.

3.2 Collaborative component

This component is a foundational part of our proposed hybrid recommender system, leveraging the collective preferences and behaviors of users to generate personalized recommendations, by leveraging accumulated knowledge from multiple data sources in a rich repository. In our proposed recommender system, rich repository can provide a powerful foundation for training and evaluating recommender algorithms and plays a key role in improving the accuracy, variety, and personalization of recommendations. Additionally, behavioral data, such as reviews, ratings provided by users, and their interaction history with items, can be combined with the demographic data to refine the recommendations.

In this phase, CF method is incorporated to make recommendations based on the preferences of similar users. First system analyzes users who have similar tastes and then, recommends items that these users have enjoyed but the target user has not yet interacted with. This collaborative approach works particularly well for users with established interaction histories. In this research, the cross-validation method is used to evaluate the classification models. In this method, the data set is divided into two parts, which is usually divided in a ratio of 70-30 (70% training data, 30% test data). One of the two parts of the data set is considered the training data set and the model is built based on it and other data set is used to evaluate the built-in model.

Afterward, several ML models are utilized in this phase to make predictions about the user's potential preferences. First, the decision tree and random forest algorithms, have been discarded due to low accuracy. Then, other algorithms applied and compared with their error values, and the algorithms that had less error rate with more accuracy have been selected. Efficient ML models with the lowest error rate, such as decision trees (C4.5), Support Vector Machines (SVM), and Deep Neural Network (DNN) are applied to predict user preferences. The outputs from these models are then aggregated by majority voting mechanism to provide initial recommendation list. Figure 2 presents how the preference list is generated in this stage. Through efficient integration of ML algorithms such as decision trees and SVM, with sentiment analysis and CF method, the system generates predictions that are then aggregated using a voting mechanism. This multi-aspect approach ensures that recommendations are both personalized and accurate, addressing challenges such as data cold start and improving the overall user experience in the recommender system. After the training samples (70%) and test samples (30%) are separated by using the balancing method, training samples have been applied to the DNN, C4.5 decision tree, and SVM-Lib algorithms as input. Each of these algorithms generates a model based on training samples. The generated models are flexible, since have a tree structure, a vector structure, and a network structure. Then, test samples are applied to these models, and based on the existing samples, maximum vote-based classification is done. The functional details of the implemented learning algorithms are as follows:

One of the important workflows to train and model the system is dividing samples into two main parts. The first part is used to train the ML model, and the second part is used as a test case for categorizing new users. Sampling is one of the stages of data mining which considered in the proposed approach. At the next step, using the demographic information of the new users and the clusters specified user clusters, the appropriate category for the new users can be found. Afterward, the next phase initialized and the collaborative component uses the outputs of this phase in cumulative modeling to address challenges like data sparsity, and cold-start problems.

3.2.1 Decision tree model

The decision tree model uses both demographic and behavioral data to segment users within a cluster further and make specific recommendations. Moreover, it experienced the least error in comparative assessments. In C4.5, once the tree is built, it can be used to predict whether users would like a particular item based on their cluster and previous behaviors. The decision tree works by recursively partitioning the dataset based on the attributes that provide the highest information gain. The information gain for an attribute A is defined as:

$$IG(D, A) = Entropy(D) - \sum_{v \grave{o} values(A)} \frac{|D_v|}{|D|} .Entropy(D_v)$$

. (3)

Where:

- Entropy is the entropy of the dataset D;
- D_v is the subset of the dataset where attribute A has value v;

Entropy is calculated as follows:

$$Entropy(D) = -\sum_{i=1}^{m} p_i .log_2(p_i) \quad (4)$$

Where p_i is the proportion of instances in class i. The C4.5 algorithm builds the decision tree by selecting the attribute with the highest information gain at each node, recursively splitting the data until a stopping criterion is met, such as reaching a maximum tree depth, or when all instances in a node belong to the same class. The resulting decision tree is used to predict the preferences of users, based on their demographic information, interactions, and behavioral data. In order to increase the accuracy of recommendations, the decision tree (C4.5) is used beside the SVM model, which is explained in the next part.

3.2.2 SVM model

To refine the predictions made by the decision tree, the system employs a SVM model for classification. The objective of SVM is to find a hyperplane that maximizes the margin between the two classes. Given a set of training data points $\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$, where $x_i \in R_d$ and R_d and $y_i \in \{-1,1\}$ the hyper plane is defined as follows:

$$w.x + b = 0 \quad (5)$$

Where:

- w is the weight vector.
- x is the input vector.
- b is the bias term.

The target of the optimization problem is to maximize the margin, while ensuring that all data points are correctly classified as follows:

$$\min \frac{1}{2} \quad w^2 \quad subject\,to \; y_i(w.x_i + b) \geq 1, \quad \forall i \quad (6)$$

To handle non-linearly separable data, the SVM uses a kernel function ($K(x_i, x_j)$), which transforms the input data into a higher-dimensional space where a hyperplane can be applied. Common kernel functions include the linear kernel, polynomial kernel, and radial basis function BF) kernel. The SVM in the proposed hybrid system uses an RBF kernel, defined as:

$$K(x_i, x_j) = exp\left(-\gamma x_i - x_j^2\right) \quad (7)$$

Where, γ is a parameter that controls the width of the Gaussian function. The SVM is trained on the users in each cluster, using both demographic and behavioral data as features. Once trained, the SVM classifies new items as relevant or irrelevant to the user, improving the overall recommendation accuracy.

3.2.3 DNN model

This model also takes the features of users and items and uses them as input to a DNN model that ranks the criteria, which predicts the ranking of the criteria. One of the most relevant aspects of a neural network is its ability to generalize, that is, to predict cases that are not included in the training set [27]. The ranking of criteria form the input to the second part, which is the general neural network with the overall rating and is used to predict the overall ranking and developed based on our previous research [28].

3.2.4 Voting based aggregation:

After each model makes its prediction, the recommender system aggregates the results using a majority voting technique. In the majority voting mechanism, each model provides a prediction, and the final decision is based on which prediction has the majority value. In order to increase accuracy, if two of the three models suggest recommending a particular item, the system will recommend that item. By integrating multiple models, the system mitigates the weaknesses of individual approaches and leverages the strengths of each method. Finally, the voting mechanism ensures that the final recommendation takes into account the predictions of all capabilities and models, result in a more robust recommendation. The output of this stage is the initial list of recommended items for each user. This list is produced based on the individual's cluster, their historical interactions, and the aggregated predictions of multiple models. The system provides these recommendations as an initial ranked list, typically showing the items most likely to be of interest to the user at the top. On the other side, the sentiment analyzer component comes into play and personalizes suggestions, also optimizes similarity checking and link suggestions to new users.

3.3 Sentiment analyzer component

As an advantage over previous research, our proposed hybrid recommender system benefits from sentiment analysis capability. By analyzing sentiment expressed in reviews, proposed recommendation system can better understand user preferences and provide personalized and effective suggestions. Summary of executive phases and detailed workflows for implementing a sentiment analyzer component are as follow:

- *Data collection and preparation*
  - Gather user-generated content such as reviews, comments, and feedbacks.
  - Preprocess the textual data, remove duplicates, irrelevant entries, and noise.
  - Normalize text (stemming, lemmatization), remove stop words and handle punctuation.
  - Ensure each user has a sufficient number of reviews for statistical robustness.
- *Sentiment model fine-tuning*
  - Select a pre-trained BERT base model [29] as the foundation for sentiment analysis.
  - Fine-tune the BERT model on a large, labeled sentiment dataset (IMDb movie reviews), adapt it to the domain-specific language and sentiment nuances.
  - Validate the fine-tuned model's performance using appropriate metrics (e.g. accuracy, F1-score) on a held-out test set.
- *Sentiment scoring& similarity checking*
  - For each review r by user u, use the fine-tuned BERT model to predict the probability of positive and negative sentiment $P_{u,r}$.
  - Aggregate review-level polarity scores for each user by average s_u .
  - This averaging reduces noise and provides a stable measure of user sentiment.
  - Compute pairwise sentiment similarity between users with $Sim_{sent}(u,v)$ criterion.
  - Values range from 0 (opposite sentiment) to 1 (identical sentiment).
  - This step enables the system to quantify how closely aligned users are in terms of their expressed opinions.
- *Hybrid fusion*

- Integrate sentiment similarity with collaborative filtering and content-based similarities.
- Collaborative Filtering Similarity (Sim_{CF}) is pearson correlation between users' latent factor vectors.
- Content-Based Similarity (Sim_{CB}) is Cosine similarity between TF-IDF feature centroids for user profiles.
- Use the hybrid similarity metric $Sim_{Hybrid}(u,v)$ to improve user-user matching and recommendation relevance.
- Visualize sentiment trends and satisfaction scores to inform further tuning and insights.

In this research, a Bidirectional Encoder Representations Transformers (BERT) base model which is a powerful deep learning model is presented to elucidate the issue of sentiment analysis. The BERT model gave an improved performance with good prediction and high accuracy compared to the other methods [30]. BERT is a transformer model for performing natural language processing, question answering and sentiment analysis. First a BERT-based sentiment classifier on movie-review data has been fine-tuned to assign each review a polarity score $p_{u,r} \in [-1,1]$ and aggregate into a user-level sentiment s_u. We adopt a pre-trained BERT model fine-tuned on a large, labeled IMDb review corpus to output review-level polarity scores. Each review r by user u yields:

$$P_{u,r} = P(Positive|r) - P(Negative|r) \tag{8}$$

Thus bounded in the range [−1,1], where +1 indicates maximally positive sentiment and −1 maximally negative sentiment. Incorporating sentiment polarity from reviews enhances recommendation relevance. Therefore, user reviews are processed using a pre-trained sentiment classifier to assign polarity scores. BERT's self-attention layers capture long-range dependencies and contextual nuances far better than earlier RNN- or CNN-based classifiers [31]. To form a stable user sentiment s_u, we average over all reviews R_u by user u:

$$s_u = \frac{1}{|R_u|} \sum_{r \in R_u} P_{u,r} \tag{9}$$

Requiring a minimum of five reviews for statistical robustness. This averaging reduces noise from individual outlier reviews and reflects overall user disposition. Then the pure sentiment similarity and pairwise sentiment alignment has been measured as follows:

$$Sim_{sent}(u,v) = 1 - |s_u - s_v| \tag{10}$$

Which ranges from 0 (diametrically opposed sentiments) to 1 (identical sentiment). Finally, in order to leverage complementary signals, we fused $Sim_{sent}(u,v)$ with collaborative-filtering similarity (Sim_{CF}) and content-based similarity (Sim_{CB}) metrics via:

$$Sim_{Hybrid}(u,v) = \alpha Sim_{CF}(u,v) + \beta Sim_{CB}(u,v) + \gamma Sim_{sent} \tag{11}$$

Where, Sim_{CF} is the pearson correlation between users' latent-factor vectors learned via regularized SVD, and Sim_{CB} is cosine similarity on TF-IDF–derived item-feature centroids for each user Medium. Weights (α=β=0.4, γ=0.2) were chosen via grid search on a validation set to optimize precision. Such a sentiment analysis approach can regulated and balance rating, content, and opinion signals for improved user-user matching.

3.4 Link prediction component

The proposed link prediction algorithm, predicts user relationships and interactions, which are incorporated into the recommendation process to further improve accuracy. To further enhance the recommendations, an extension of the FriendLink algorithm [32, 33] has been used and improved to predict future user interactions, based on the existing social relationships. The probability that a link will form between two users u_i and u_j is computed based on their proximity in the social network graph. In our improved link prediction strategy, a hybrid similarity criterion is used to predict and suggest target links. It uses proximity measures to predict potential future interactions between users, which are then used to adjust the recommendation scores for items. For example, if two users are predicted to become friends, the system can recommend items that one user has already rated highly to another user. The general procedure of the improved link prediction strategy is as follows: it forms a connection graph between the new user and the users who have the most similarity, based on the $Sim_{Hybrid}(u,v)$, and according to a pre-defined threshold (e.g. five). It should be noted that in this graph, all friends of the users extracted by the hybrid similarity metric participate in the graph and the link prediction algorithm is executed on these friends. Therefore, it is not necessary to re-navigate the entire network graph. By calculating the similarity of the new user with other friends of similar users is calculated based on the path length and the threshold, the results of this part are combined with the outputs of sentiment analyzer component and the final users with the highest score are recommended as friends to the new user. Moreover, in order to improve the accuracy of user selection and suggestion in the proposed approach, by applying the degree of neighborhood of the users, final similarity criterion is defined as follows:

$$Final\,Sim(n_j,u_j) = \left(\sum_{i=2}^{l} \frac{1}{i-1} . \frac{\left| path^{i}_{n_j,u_j} \right|}{\prod_{y=2}^{i} (n-y)} \right) * \left(\frac{D_{u_j}}{N} \right) * Sim_{Hybrid}(u,v) \tag{12}$$

Where:

- n_i: Is a new user who has just been entered in the social network graph, and u_j is the one-by-one users who are most similar to new user.
- L: Specifies the length of the path.
- n: Specifies the total number of nodes in the graph.

- $\sum_{i=2}^{l} \frac{1}{i-1}$: Is a weighting factor that is more effective for paths greater than *L=2*.
- $\left|path^{i}_{n_j,u_j}\right|$: The number of paths between the new user and the user who is connected to the new user's friends is shown according to their length.
- $\prod_{y=2}^{i}(n-y)$: Is the number of possible paths from the new user to the target user.
- N : The total number of degrees of the nodes.
- D_{u_j} : Is the degree of the target node.

Where, n_i denotes the newly added user to the social network graph, whereas u_j represents an existing user who exhibits the highest similarity to n_i, based on the hybrid similarity metric introduced earlier. The parameter *L* specifies the length of the path connecting two users within the graph, serving as an indicator of their topological proximity. The term n_i refers to the total number of nodes (i.e., users) in the social graph, while the l_w weighting factor is introduced to adjust the contribution of longer paths (L_2) during the link prediction process. This ensures that indirect connections (although weaker than direct links) still contribute proportionally to the prediction score. Furthermore, $P_{nj,ui}(L)$ indicates the number of possible paths of length *L* between the new user n_i and the target user u_j. This parameter captures how many distinct connection routes exist between two nodes at a given path length. The denominator includes D_{uj}, which denotes the degree of the target node, representing the total number of existing connections that the user u_j maintains within the network.

By incorporating these elements, the proposed similarity criterion effectively models the probability of new link formation by balancing path length, connection density, and similarity strength.

While our method is inspired by the original *FriendLink* local-path approach, it improves from important ways by Hybrid Similarity Incorporation. The traditional FriendLink algorithm computes proximity based only on graph topology. Our modified version replaces the simple path-based probability with a multi-term predictor that integrates:

- Node degree centrality (*deg(u)*),
- Hybrid similarity (Sim_{Hybrid}),
- Adjusted path-length weighting (λ^L),

The link prediction score is thus defined by Eq. (13):

$$Score(u_i,u_j) = \sum_{L=1}^{Lmax} \frac{\lambda^{L}}{Deg(u_j)} * \left|Paths_{L}(u_i, u_j)\right| * Sim_{Hybrid}(u_i, u_j) \quad (13)$$

This integrates node degree centrality, hybrid similarity, and path weighting ($\lambda=0.6$). Degree normalization avoids bias toward highly connected users, improving prediction stability. Pseudocode for the FriendLink+ algorithm is:

FriendLink+ algorithm

Input: u_{new}, *Users*, L_{max}, λ;
Output: *Score[u_j], Top-N*;

1. For each user u_j in Users do
2. *Sim = SimHybrid (u_{new}, u_j);*
3. *Score[u_j] = 0;*
4. For *L = 1* to L_{max} do
5. *PathCount = $count_{paths}$(u_{new}, uj, L);*
6. *Score[u_j] += (λ^L / deg(u_j)) * $Path_{Count}$ * Sim;*
7. End For;
8. Rank users by Score[u_j];
9. Recommend Top-N with highest scores;

Algorithm 2. The pseudo code of the FriendLink+ algorithm

This integration not only captures network proximity but also propagates sentiment-augmented similarity through indirect neighbors (a hybrid graph-regularized mechanism**)** conceptually similar to a shallow GNN layer, yet computationally lighter. In fact, it is one of the achievements and benefits of the proposed approach in this research.

In summary, similarities are calculated based on the threshold between the new user and similar users in the selected category and a hybrid similarity criterion is determined based on user attributes, reviews, and interactions. After the new user's similarity to other friends of similar users is calculated based on the path length and threshold, the results of this part are combined with the previous step and the final users with the highest score are recommended as friends to the new user. The advantage of using this strategy is that the calculation and prediction of the link is done after sentiment analysis, and finding similarities between neighboring nodes in the network graph based on

the proposed hybrid criterion is able to provide more accurate suggestions. Moreover, it is not necessary to re-navigate the entire network graph and the prediction speed increases [28].
The next section focuses on examining the simulation results of the proposed approach and comparing with related work.

4. Evaluation and results

The performance of the proposed hybrid recommender system evaluated based on the several major metrics like MAE, RMSE, Precision, Recall, and the overall system's ability to address the cold start problem. The basic setup and hardware configuration has been presented in Table 2. The dataset used for evaluation is the MovieLens dataset [34], containing over one million records. To validate the advantages of the proposed recommender system, comparisons are made between the hybrid approach and other standard recommender systems used collaborative based approach [9], [21], and also content based approach [35], [19], as baseline methods. Additionally, we analyze the impact of demographic data, clustering techniques, sentiment analysis, and ML algorithms on the accuracy of suggestions in the proposed system.

4.1 Performance comparison

To demonstrate the effectiveness of the hybrid recommender system, performance of the proposed system is compared to traditional CF and CBF based systems. The initial evaluated metrics are:

- Mean Absolute Error (MAE): Measures the average magnitude of the errors between predicted and actual ratings:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|r_i - \hat{r}_i\right| \tag{14}$$

Where, r_i is the actual rating and $\hat{r}_i$ is the predicted rating.

- Root Mean Square Error (RMSE): measures the standard deviation of the prediction errors:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \hat{r}_i)^2} \tag{15}$$

Both MAE and RMSE are calculated to assess how well the system predicts user ratings for items, with lower values indicating better performance. Table 3 presents the evaluation results for MAE and RMSE. From the results of Table 3, we observe that the proposed hybrid recommender system significantly reduces both MAE and RMSE, compared to the CF and CBF based methods. The MAE is lowered by 14.4% compared to CF method and 11.8% compared to CBF method. Figure 3 depicts the MAE values for CF, CBF, and the proposed hybrid system and illustrates the improved accuracy of the proposed recommender system and reduction in error achieved by the proposed hybrid approach. Figure 4 depicts the RMSE comparison for CF, CBF, and the proposed hybrid system. In Figure 4, the RMSE values for the three models have been presented. The RMSE is reduced by 15.9%, and 13.9%, respectively. The hybrid system presents the smallest RMSE, indicating more accurate predictions, overall.

4.2 Cold start scenario

To assess the system's performance in cold start scenarios, new users with no interaction history were added to the system. These users were clustered based on demographic data using the clustering algorithm. The results were evaluated based on the recommendations provided for these users. Figure 5 depicts the MAE values after evaluations. As it can be seen in the Figure 5, the proposed hybrid system demonstrates a 27.4% reduction in MAE for new users, compared to CF based method, showcasing its ability to handle the cold start problem, effectively. The sentiment analysis capability, decision trees and SVM models contribute significantly by classifying users based on demographic and social data. Above presented graph could illustrate how MAE decreases as our hybrid recommender system is applied to cold start users, outperforming CF and CBF methods. Moreover, for new items that had no previous user interactions, the hybrid model again leveraged clustering and decision tree predictions. Figure 6 presents the MAE metric for item cold start situation. Results in Figure 6, demonstrate that the proposed system performs significantly better in the item cold start scenarios as well, with 29.6% decrease in MAE, compared to CF method.

4.3 Link prediction performance

Our improved link prediction strategy impacts on the metrics Precision and Recall, as presented in Table 4. The improved link prediction strategy leads to 7% improvement in Precision, and 10.8% improvement in Recall. Incorporating social data and link prediction into the recommendation process helps to refine recommendations and increases the overall system's accuracy.

4.4 System scalability and performance on large datasets

To ensure the scalability of the proposed recommender system with large datasets, the performance was tested on progressively larger subsets of the MovieLens dataset. The system was evaluated based on both its accuracy (measured by MAE and RMSE), and its computation time as presented in Table 5. As it can be seen in the Table 5, the proposed recommender system maintains its accuracy even as the dataset size increases, with a slight improvement in MAE and RMSE, due to the availability of more user interaction data. The computation time increases linearly with dataset size, which is expected in systems using clustering and ML techniques.

5. Discussion and comparison

Despite the significant advances in recent recommender systems, a notable gap exists in the literature regarding the effective resolution of the cold start problem and sparsity, particularly in the scope of social networks. Some existing approaches, especially those relying on the CF methods like [9], [21], often struggle when encountering users, or items with limited historical data. These methods are heavily dependent on user-item interaction history, meaning that when data is sparse, the recommendation quality decreases.

While CBF based approaches like [35], [19] provide an alternative by leveraging item attributes to make recommendations, face limitations when the user's preferences are inadequate, or unknown. However, such systems often suffer from excessive specialization and lack of diversity. Hybrid approaches like [20], [10], and [22] integrate useful elements of both CBF and CF, and often offer promising solutions. However, still there is a lack of research that adequately explores how add-on capabilities and techniques such as demographic data, user clustering, learning models, sentiment analysis, and link prediction can further improve the accuracy and performance in recommendation systems and mitigate the cold start and data sparsity challenges. Our research aims to fill this gap by developing a novel hybrid recommender system that integrates mentioned capabilities to enhance recommendation accuracy and solve the cold start and sparsity challenges, more effectively. Table 6 presents a detailed comparison of our proposed recommender system with related work. The results of several evaluations present clear improvements in handling the cold start problem, increasing recommendation accuracy, and integrating social data through the sentiment-enhanced hybrid approach. The combination of CF, CBF, sentiment analysis, demographic-based clustering, decision trees, SVMs, and improved link prediction, offers a robust solution to traditional recommendation system challenges. The system demonstrates scalability with large datasets and maintains high accuracy across different test scenarios. The inclusion of advanced learning techniques and sentiment analysis significantly enhances both precision and recall, making the hybrid system well-suited for complex recommendation tasks in social networks.

Moreover, Table 7 includes explicit numerical improvements in error and performance metrics over key related work. We benchmarked against three representative recent systems incorporating demographics, or clustering. As can be deduced from the information in Table 7, the improvement stems from integrating (1) sentiment fusion using BERT-derived embeddings, (2) degree-normalized FriendLink+ prediction**,** and (3) Bayesian Information Criterion–optimized X-Means clustering. Finally note that, to enhance the reproducibility and transparency of the proposed framework, the complete implementation code, hyper parameter settings, and data-split scripts have been provided in the supplementary materials. The MovieLens 1M dataset was stratified into training (70%), validation (15%), and testing (15%) subsets, maintaining the user–item interaction ratios to avoid isolated entities during learning. A sensitivity analysis was further conducted to evaluate the robustness of the model under different parameter configurations. The results showed that the framework's performance remained stable across a broad range of parameter variations. Specifically, the MAE fluctuated by less than ±2% for different combinations of (α, β, γ) satisfying $\alpha + \beta + \gamma = 1$, and by ≤3% for varying cluster numbers (K). The optimal DNN learning rate ($\eta = 1e{-}4$) provided the best trade-off between convergence speed and stability. To ensure statistical consistency, all experiments were repeated using three independent random seeds (42, 77, and 113), and the performance variation was negligible (< ±0.005 RMSE). These results demonstrate that the proposed hybrid framework is robust and reproducible under different hyper parameter settings.

6. Conclusion and future work

In light of identified research gaps and implementation of the proposed hybrid recommender system, a clear inference emerges, suggesting that our proposed system, involving the flexible integration of CF, and CBF methods in conjunction with sentiment analysis capability, presents a promising avenue for this research scope. The novelty of this research lies in its hybrid and multi-dimensional approach. This research contributes to the growing body of knowledge surrounding recommendation systems by integrating the strengths of different techniques, while incorporating demographic data, user clustering, ML models, sentiment analysis, and link prediction to improve recommendation accuracy in cold start scenarios and offer a robust solution to traditional challenges in this research area. Future work can focus on enhancing the scalability of the system for larger and real-time execution environments. Implementing distributed computing techniques such as parallel processing, or utilizing cloud-based infrastructures can optimize computation time and also increase its richness and comprehensiveness. The insights gained from this study provide a foundation for developing more sophisticated and user-friendly recommendation systems that meet the demands of diverse users and evolving data landscapes.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for prof it sectors.

Conflicts of interest

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

References

1. Kose, U., Arslan, A. "Time series prediction with a hybrid system formed by artificial neural network and cognitive development optimization algorithm," Scientia Iranica, **26**, pp. 942-58 (2019). https://doi.org/10.24200/sci.2018.20033

2. Singh, P. K., Pramanik, P. K. D., Dey, A. K., et al. "Recommender systems: an overview, research trends, and future directions," International Journal of Business and Systems Research, **15**, pp. 14-52 (2021). https://doi.org/10.1504/IJBSR.2021.111753

3. Khan, Z. Y., Niu, Z., Sandiwarno, S., et al. "Deep learning techniques for rating prediction: a survey of the state-of-the-art," Artificial Intelligence Review, **54**, pp. 95-135 (2021). https://doi.org/10.1007/s10462-020-09892-9

4. Mishra, K. N., Mishra, A., Barwal, P. N., et al. "Natural Language Processing and Machine Learning-Based Solution of Cold Start Problem Using Collaborative Filtering Approach," Electronics, **13**, pp. 4331 (2024). https://doi.org/10.3390/electronics13214331

5. Papadakis, H., Papagrigoriou, A., Panagiotakis, C., et al. "Collaborative filtering recommender systems

taxonomy," Knowledge and Information Systems, **64**, pp. 35-74 (2022). https://doi.org/10.1007/s10115-021-01628-7

6. Stitini, O., Kaloun, S., and Bencharef, O. "Towards a robust solution to mitigate all content-based filtering drawbacks within a recommendation system," International Journal of Systematic Innovation, **7**, pp. 89-111 (2023). https://doi.org/10.6977/IJoSI.202309_7(7).0006

7. Eliyas, S., Ranjana, P., "Recommendation systems: Content-based filtering vs collaborative filtering," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE); IEEE; Published (2022). https://doi.org/10.1109/ICACITE53722.2022.9823730

8. Parthasarathy, G., Sathiya Devi, S. "Hybrid recommendation system based on collaborative and content-based filtering," Cybernetics and Systems, **54**, pp. 432-53 (2023). https://doi.org/10.1080/01969722.2022.2062544

9. Dang, C. N., Moreno-García, M. N., Prieta, F. D. l. "An approach to integrating sentiment analysis into recommender systems," Sensors, **21**, pp. 5666 (2021). https://doi.org/10.3390/s21165666

10. Yanto, H., Isa, S. M. "Hybrid Approach for Movie Recommendation System Using Double Collaborative Filtering and Sequential Pattern Analysis," Journal of Theoretical and Applied Information Technology, **101**, (2023). https://doi.org/10.1145/361219.361220

11. Mondal, B., Patra, O., Mishra, S., et al. "A course recommendation system based on grades," 2020 international conference on computer science, engineering and applications (ICCSEA); IEEE; Published (2020). https://doi.org/10.1109/52.582976

12. Madani, Y., Ezzikouri, H., Erritali, M., et al. "Finding optimal pedagogical content in an adaptive e-learning platform using a new recommendation approach and reinforcement learning," Journal of Ambient Intelligence and Humanized Computing, **11**, pp. 3921-36 (2020). https://doi.org/10.1007/s12652-019-01627-1

13. Ali, S. M., Nayak, G. K., Lenka, R. K., et al. "Movie recommendation system using genome tags and content-based filtering," Advances in Data and Information Sciences: Proceedings of ICDIS-2017, Volume 1; Springer; Published. (2018). https://doi.org/10.1007/978-981-10-8360-0_8

14. Bablani, D., Gupta, R., and Gokhale, T. "Hybrid Approach to Music Recommender Systems," Int J Res Appl Sci Eng Technol, **5**, pp. 1-6 (2022). https://doi.org/10.1145/245108.245124

15. Zare, A., Motadel, M. R., and Jalali, A. "Presenting a hybrid model in social networks recommendation system architecture development," AI & SOCIETY, **35**, pp. 469-83 (2020). https://doi.org/10.1007/s00146-019-00893-z

16. Zhang, L., Wang, S., and Liu, B. "Deep learning for sentiment analysis: A survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **8**, pp. 12-53 (2018). https://doi.org/10.1002/widm.1253

17. Ahuja, R., Solanki, A., and Nayyar, A., "Movie recommender system using k-means clustering and k-nearest neighbor," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence); IEEE; Published. (2019). https://doi.org/10.1109/CONFLUENCE.2019.8776969

18. Banjac, J., Altinigne, C. Y., Kypraiou, S., et al. "Movinder: A Movie Recommendation System for Groups," (2020). https://doi.org/10.1002/wics.103

19. Cami, B. R., Hassanpour, H., and Mashayekhi, H. "A content-based movie recommender system based on temporal user preferences," 2017 3rd Iranian conference on intelligent systems and signal processing (ICSPIS); IEEE; Published. (2017). https://doi.org/10.1109/ICSPIS.2017.8311601

20. Hassan, B. A., Hassan, A. A., Lu, J., et al. "A Proposed Hybrid Recommender System for Tourism Industry in Iraq Using Evolutionary Apriori and K-means Algorithms," arXiv:250408767, pp. 1-20 (2025). https://doi.org/10.48550/arXiv.2504.08767

21. Lin, C.-H., Chi, H. "A novel movie recommendation system based on collaborative filtering and neural networks," Advanced Information Networking and Applications: Proceedings of the 33rd International Conference on Advanced Information Networking and Applications (AINA-2019) **33**; Springer; Published. (2020). https://doi.org/10.1080/09332480.2010.10739787

22. Walek, B., Fojtik, V. "A hybrid recommender system for recommending relevant movies using an expert system," Expert Systems with Applications, **158**, pp. 113452 (2020). https://doi.org/10.1016/j.eswa.2020.113452

23. Wang, Z., Liang, J., Li, R., et al. "An approach to cold-start link prediction: Establishing connections between non-topological and topological information," IEEE Transactions on Knowledge and Data Engineering, **28**, pp. 2857-70 (2016). https://doi.org/10.1109/TKDE.2016.2597823

24. Wang, Y., Wang, M., and Xu, W. "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework,"

Wireless Communications and Mobile Computing, 2018, pp. 8263704 (2018). https://doi.org/10.1155/2018/8263704

25. Mohamadrezaei, R., Ravanmehr, R. "A trust-based recommender system for e-Learning environment using fuzzy clustering," Technology of Education Journal (TEJ), **15**, pp. 439-64 (2021). https://doi.org/10.22061/tej.2021.6807.2454

26. Jaeger, A., Banks, D. "Cluster analysis: A modern statistical review," Wiley Interdisciplinary Reviews: Computational Statistics, **15**, pp. e1597 (2023). https://doi.org/10.1002/wics.1597

27. Ghorbanian, J., Ahmadi, M., and Soltani, R. "Design predictive tool and optimization of journal bearing using neural network model and multi-objective genetic algorithm," Scientia Iranica, **18**, pp. 1095-105 (2011). https://doi.org/10.1016/j.scient.2011.08.007

28. Bazargani, M., H. Alizadeh, S., and Masoumi, B. "Group deep neural network approach in semantic recommendation system for movie recommendation in online networks," Electronic Commerce Research, pp. 1-40 (2024). https://doi.org/10.1007/s10660-024-09897-4

29. Wu, Y., Jin, Z., Shi, C., et al. "Research on the application of deep learning-based BERT model in sentiment analysis," arXiv preprint arXiv:240308217, (2024). https://doi.org/10.48550/arXiv.2403.08217

30. Zhuang, Y., Kim, J. "A bert-based multi-criteria recommender system for hotel promotion management," Sustainability, 13, pp. 8039 (2021). https://doi.org/10.3390/su13148039

31. Eang, C., Lee, S. "Improving the Accuracy and Effectiveness of Text Classification Based on the Integration of the Bert Model and a Recurrent Neural Network (RNN_Bert_Based) ," Applied Sciences, **14**, pp. 8388 (2024). https://doi.org/10.3390/app14188388

32. Papadimitriou, A., Symeonidis, P., and Manolopoulos, Y. "Friendlink: link prediction in social networks via bounded local path traversal," 2011 International Conference on Computational Aspects of Social Networks (CASoN); IEEE; Published (2011). https://doi.org/10.1109/CASON.2011.6085920

33. Vahidi Farashah, M., Etebarian, A., Azmi, R., et al. "A hybrid recommender system based-on link prediction for movie baskets analysis," Journal of Big Data, **8**, pp. 1-24 (2021). https://doi.org/10.1186/s40537-021-00422-0

34. Harper, F. M., Konstan, J. A. "The movielens datasets: History and context," Acm transactions on interactive intelligent systems (tiis), 5, pp. 1-19 (2015). https://doi.org/10.1145/2827872

35. Çano, E., Morisio, M. "Hybrid recommender systems: A systematic literature review," Intelligent data analysis, **21**, pp. 1487-524 (2017). https://doi.org/10.3233/IDA-163209

36. Shi, L., Wu, W., Guo, W., et al. "SENGR: Sentiment-Enhanced Neural Graph Recommender," Information Sciences, vol. 589, pp. 655–669 (2022). https://doi.org/10.1016/j.ins.2021.12.120

37. Prottasha, N. J., Sami, A. A., Kowsher, M., et al. "Transfer Learning for Sentiment Analysis Using BERT-Based Supervised Fine-Tuning," Sensors (Basel), vol. 22, no. 11, art. 4157 (2022). https://doi.org/10.3390/s22114157

38. Liu, N., and Zhao, J. "A BERT-Based Aspect-Level Sentiment Analysis Algorithm for Cross-Domain Text," Mathematical Problems in Engineering (2022). https://doi.org/10.1155/2022/8726621

39. Li, Y., Zhai, X., Alzantot, M., et al. "CALRec: Contrastive Alignment of Generative LLMs for Sequential Recommendation," Proc. ACM RecSys (2024). https://doi.org/10.1145/3640457.3688121

40. Markchom, T., Liang, H., and Ferryman, J. "Scalable and Explainable Visually-Aware Recommender Systems," Knowledge-Based Systems, vol. **263**, art. 110258 (2023). https://doi.org/10.1016/j.knosys.2023.110258

41. Al-Hasan, T. M., Sayed, A. N., Bensaali, F., et al. "From Traditional Recommender Systems to GPT-Based Chatbots: A Survey of Recent Developments and Future Directions," Big Data and Cognitive Computing, **8**, no. 4, art. 36 (2024). https://doi.org/10.3390/bdcc8040036

42. Lops, P., Silletti, A., Polignano, M., et al. "Reproducibility of LLM-Based Recommender Systems: The Case Study of the P5 Paradigm," Proc. ACM RecSys (2024). https://doi.org/10.1145/3640457.3688072

Biographies

Nafiseh Fareghzadeh is a faculty of Computer Science Department, Khodabandeh Branch, Islamic Azad University, Zanjan. She has PhD in Software Systems Science, from Science and Research Branch, Islamic Azad University, Tehran. She has published related research papers in international journals and scientific conferences. Her research interests are: Artificial Intelligence, Blockchain, Internet of Thing (IoT), optimization problems in Cloud and Fog Computing.

Mehdi Bazargani recieved PhD degree in software system and is a full-time faculty member at Islamic Azad University, Zanjan Branch, and a lecturer in Distributed Systems and Cloud Computing (CC). He is interested in research in the

fields of recommender systems, distributed systems, cloud computing, and the Internet of Things (IoT).

Appendix A: Figures and tables captions

Figure 1. Proposed hybrid recommender system

Figure2. Majority voting mechanism for generating preference lists

Figure 3. Performance Comparison (MAE)

Figure 4. Performance Comparison (RMSE)

Figure 5. User Cold Start Problem Resolution (MAE)

Figure 6. Item cold start problem resolution (MAE)

Table1. Comparison with related work

Table 2. Hardware configuration

Table 3. Evaluation Results

Table 4. Evaluating the Impact of Link Prediction

Table 5. System Scalability

Table 6. Comparison between the proposed approach and related work

Table 7. Quantitative comparison with related work

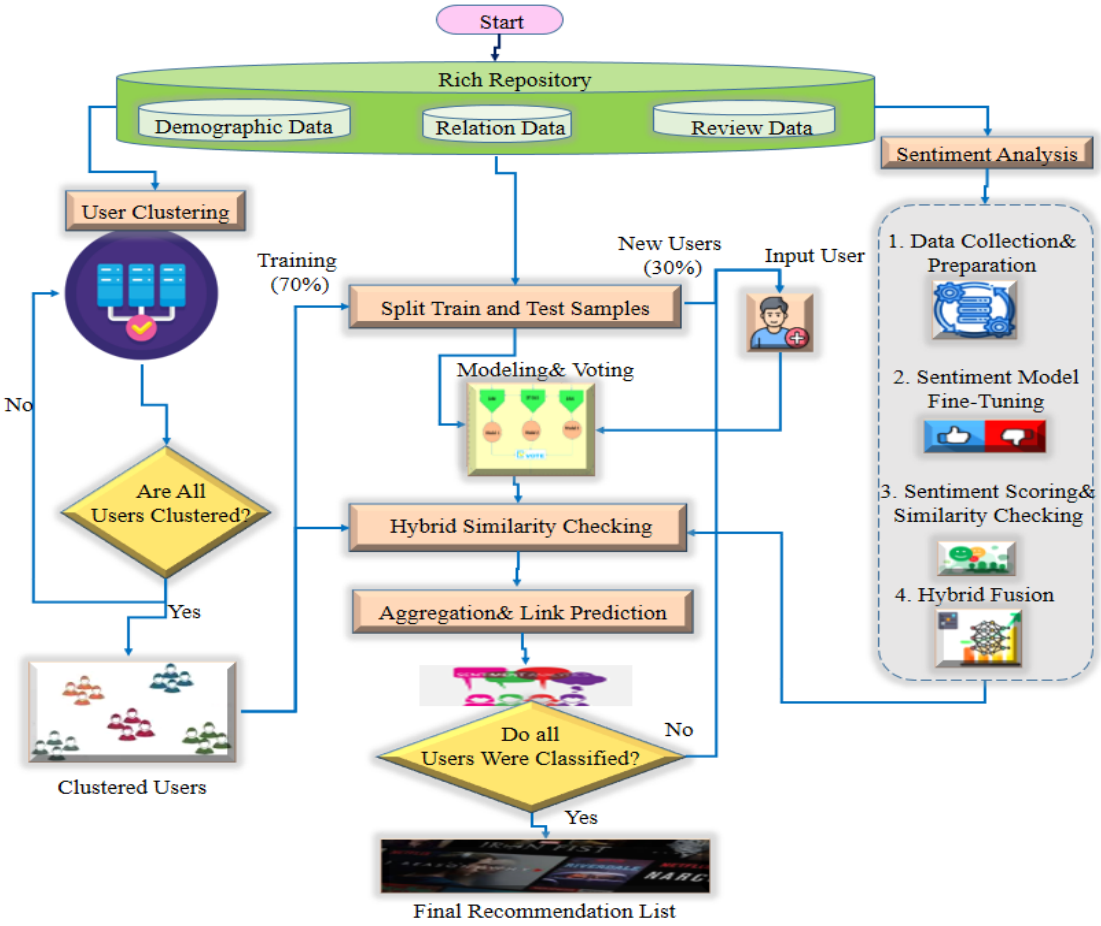

Figure 1. Proposed hybrid recommender system

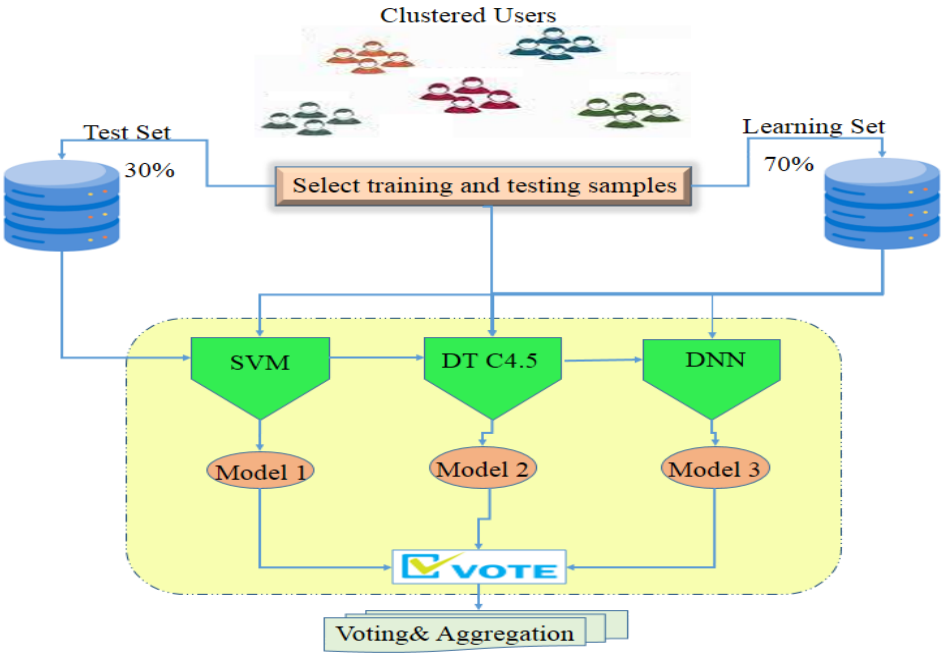

Figure2. Majority voting mechanism for generating preference lists

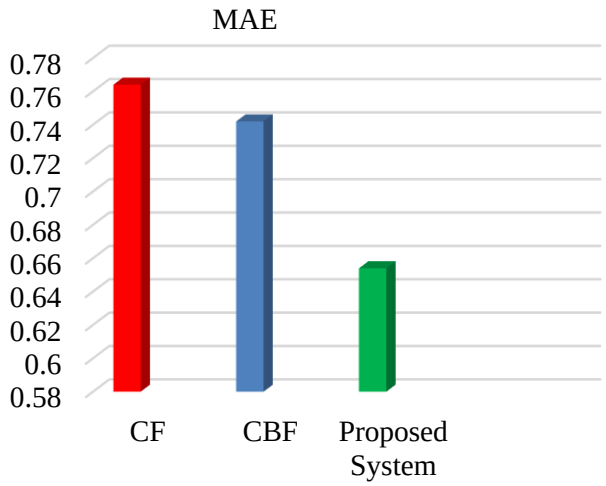

Figure 3. Performance comparison (MAE)

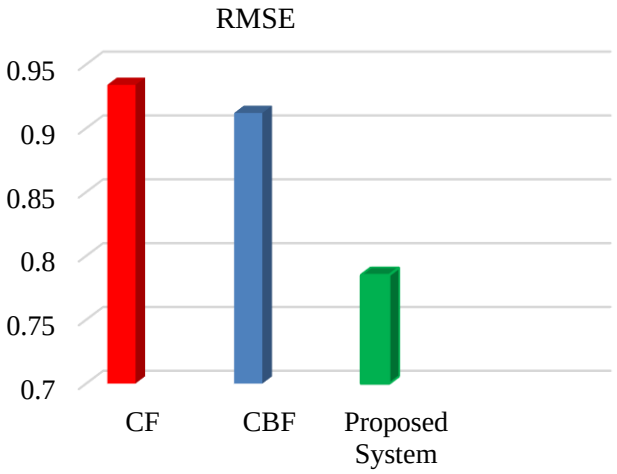

Figure 4. Performance comparison (RMSE)

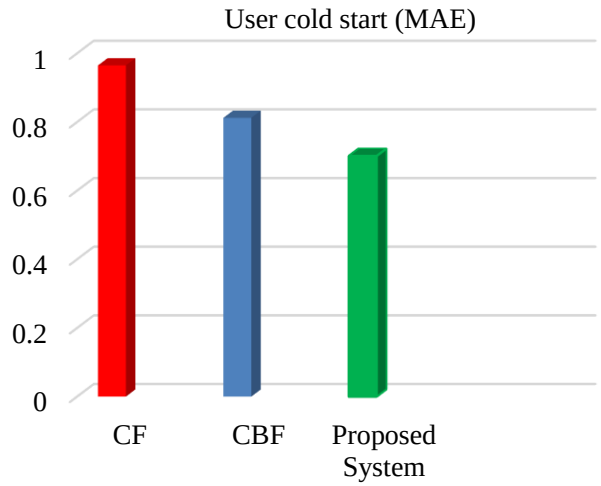

Figure 5. User cold start problem resolution (MAE)

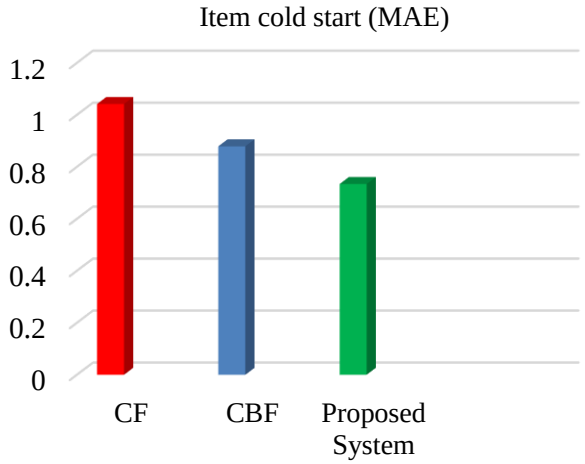

Figure 6. Item cold start problem resolution (MAE)

Table1. Comparison with related work

Reference	Approach and achievements	Challenges and gaps
Ahuja et al. [17]	Hybrid recommender system using K-Nearest algorithm neighbor and K-means clustering	Limited data set, without sentimental analysis, and feature significance.
Ali et al. [13]	Hybrid model that leverages genomic tags of movies with CBF	Feature limitation problem
Banjac et al. [18]	Movie recommender system to maximize the collective satisfaction of a group of users.	Limited implementation
Cami et al. [19]	A CBF based recommender system based on temporal user preferences	Low user serendipity and low accuracy with limited diversity
Dang et al. [9]	Integrates sentiment analysis into CF	CF challenges, limited domain
Hassan et al. [20]	Hybrid recommender with evolutionary Apriority and K-means clustering algorithms	Specific application domain
Lin et al. [21]	Movie recommendation system based on CF and neural networks	High MAE , RMSE, without sentiment analysis capability
Madani et al. [12]	Hybrid CF with sentiment analysis	Cold start, gray sheep problems persist
Walek et al. [22]	A hybrid recommender system for recommending relevant movies, based on the CF, CBF, and fuzzy expert system	Complex model without sentiment analysis capability
Wang et al. [23]	Latent-feature representation model for cold start link prediction	Needs learning on the latent-feature representation model for dynamicity
Yanto et al. [10]	Hybrid collaborative filtering method for movie recommendations	Cold start and CF challenges remain
Wang et al. [24]	Sentiment-enhanced hybrid recommender system the on Spark platform	Limited implementation

Table 2. Hardware configuration

Specification	Details
CPU	Intel® Xeon® Silver 4310 (2.1 GHz × 12 cores)
GPU	NVIDIA RTX A5000 (24 GB)
RAM	128 GB DDR4
OS	Ubuntu 22.04 LTS
Frameworks	Python 3.10, PyTorch 2.2, Scikit-learn 1.5

Table 3. Evaluation Results

Methods	MAE	RMSE
CF	0.764	0.934
CBF	0.742	0.912
Proposed system	0.654	0.785

Table 4. Evaluating the impact of link prediction

Scenario	Precision	Recall
Without link prediction	0.821	0.743
With improved link prediction	0.895	0.849

Table 5. System Scalability

Dataset	MAE	RMSE	Computation time (Sec)
Small dataset (100K ratings)	0.674	0.792	15
Medium dataset (500K ratings)	0.662	0.789	38
Large dataset (1M+ ratings)	0.654	0.785	66

Table 6. Comparison between the proposed approach and related work

Method	Prior work	Differences in our proposed system
CBF	Çano and Morisio [35] Cami et al. [19] Shi et al. [36] Prottasha et al. [37]	We cluster users and integrate CBF, CF, and sentiment analysis.
CF	Dang et al. [9] Lin et al. [21] Liu et al. [38] Li et al. [39]	BERT-based scores fused via a tunable combined metric.
Link prediction	Wang et al. [23] Markchom et al. [40]	Improved link prediction to assign new users to clusters.
Hybrid approach	Yanto et al. [10] Walek et al. [22] Al-Hasan et al. [41] Silletti et al. [42]	Multi-dimentional integration of CF, CBF, ML techniques, sentiment analysis, and improved link prediction.
Movie datasets analysis	Banjac et al. [18]	Employ both MovieLens-1M and an IMDb reviews subset for evaluation to support scalability.

Table 7. Quantitative comparison with related work

Method	Key features	MAE	RMSE	Precision	Recall	Improvement over aseline CF (%)
Zare et al. [15]	CF + Demographic Data	0.732	0.911	0.842	0.795	–
Wang et al. [24]	CF + Sentiment Fusion (Spark)	0.704	0.884	0.854	0.809	+7.6
Hassan et al. [20]	Hybrid CF+CBF + K-Means	0.681	0.823	0.876	0.824	+10.8
Proposed FriendLink+ Hybrid Approach	CF+CBF+Sentiment+BERT + Clustering + Link Prediction	0.654	0.785	0.895	0.849	+14.4