Sigma J Eng Nat Sci, Vol. 44, No. 3, pp. 1799–1815, June, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.2064

Research Article

A robust hybrid movie recommendation system using clustering, KNN, and cosine similarity techniques

Kamepalli S L PRASANNA[1], Thulasi BIKKU[1], V S S L Deepak JANAPA[1], Vamsidhar PUVVADA[1], Ganesh CHALLA[1], Srinivasarao THOTA[2,*], Sarvani Kocherla KOTA[3], Abayomi Ayotunde AYOADE[4]

[1]Department of Computer Science and Engineering, Amrita School of Computing Amaravati, Amrita Vishwa Vidyapeetham, Amaravati, Andhra Pradesh, 522503, India
[2]Department of Mathematics, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amaravati, Andhra Prdesh, 522503, India
[3]Department of Business, Amrita School of Business, Amrita Vishwa Vidyapeetham, Amaravati, Andhra Prdesh, 522503, India
[4]Department of Mathematics, Faculty of Science, University of Lagos, Lagos, 101017, Nigeria

ARTICLE INFO

Article history
Received: 18 July 2024
Revised: 25 October 2024
Accepted: 30 January 2025

Keywords:
Collaborative Filtering; Content-Based Filtering; Data Refresh Mechanism; Hybrid Filtering; Natural Language Processing; Recommender Systems

ABSTRACT

Traditional Movie recommendation systems (MRS) typically depends on either content-based filtering or collaborative filtering methods, which often results in generic suggestions lacking in person-specific and accuracy. Our research work addresses the limitations of conventional movie recommendation systems. The proposed solution integrates both 'Content-based and Collaborative Filtering' methodologies to enhance the recommendation accuracy based on individual preferences. Techniques such as Cosine Similarity, K-Nearest Neighbour (KNN), K-means Clustering, and small-scale natural language processing (NLP) are employed to improve diversity and precision of recommendations. The use of a hybrid approach which uses Item and User-based collaborative filtering methods for improving recommendations through similarities and preferences of movies. It describes how the use of machine learning algorithms can be incorporated in the process of training and updating collaborative dataset, enabled by the Data Refresh Mechanism, for achieving reliability of movie suggestions by handling data conflicts. In terms of Content Based filtering, the TMDB 5000 Movies Dataset will be used instead of Collaborative Filtering. Comparisons of different models have been made, which reveals how Content-based and Collaborative-based Filtering perform effectively in terms of recommendations accuracy around 87.5%. It should be noted that while comprehensive results are presented for Content Based Filtering, limitations in resources and the complexity of implementing and combining both approaches may restrict the depth of results for Collaborative Filtering. Additionally, insights into the performance of the hybrid model and strategies for managing data loss during refresh cycles are discussed. The future applications of our proposed model are Enhanced Personalization in Streaming Services, Improvement in E-commerce and Marketing, Applicability in social media and Content Platforms and Academic and Research Implications.

Cite this article as: Prasanna KSL, Bikku T, Janapav SSLD, Puvvada V, Challa G, Thota S, Kota SK, Ayoade AA. A robust hybrid movie recommendation system using clustering, KNN, and cosine similarity techniques. Sigma J Eng Nat Sci 2026;44(3):1799–1815.

***Corresponding author.**
*E-mail address: t_srinivasarao@av.amrita.edu

This paper was recommended for publication in revised form by Regional Editor Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

INTRODUCTION

In today's world, the movie preferences often leave individuals struggling to select a film that matches their personal interests. Despite the plethora of options available through search engines like Bing or Google, the process can be overwhelming, yielding hundreds of recommendations that fail to capture the collective preferences of diverse users [1]. This inefficiency of the search engines not only consumes lot of time but also reduces the enthusiasm to select and watch movies. To solve this issue, integrating content-based and collaborative-based filtering approaches offers a promising solution [2]. Our research focuses on improving the overall viewing experienceby allowing users to select their preferred recommendation method. These recommendation systems find usage in many sectors such as e-commerce and over-the-top (OTT). It shows how recommenders are useful in today's society [3]. As these systems continue to evolve, their importance and wide applicability in the modern society becomes more and more evident. However, conventional approaches have their drawbacks, necessitating continuous improvement in these systems. With this research, our objective will be identifying and overcoming the limitations of conventional methods to improve the capability and performance of movie recommendation systems. Studies conducted recently reveal the ability of data refresh processes to enable systems adapt to changes in the user data to provide more up-to-date recommendations [4]. For this reason, the focus of our research will be the development of an integrated approach that involves both content and collaborative filtering in conjunction with sentiment analysis of tweets related to movies to counterbalance the shortcomings of the conventional movie recommendation systems [5]. Our solution will resolve data loss and resource constraints problems in order to provide personalized and diversified movie recommendations through a combination of data refresh process with sentiment analysis. Recommenders can be broadly classified into two major types.

Content-Based Filtering

This method is based on examining the properties of individual items and predicting the preferences of users based on those properties as illustrated in Figure 1 below.

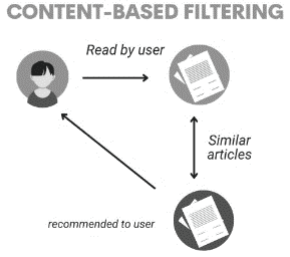

Figure 1. Content based filtering.

The process requires the construction of profiles for both users and items, usually by identifying features of the individual items. The primary goal is to recommend items like previous shown interests of user, utilizing techniques such as genre based, popularity-based and yearly movies recommendations searches. Our method aims to enhances recommendation accuracy by aligning with users' historical preferences and the specific attributes of the items [6].

Collaborative-Based Filtering

This approach focuses on analyzing the behavior and preferences of the user to identify the patterns as shown in Figure 2. It leverages the collective experiences within user community to predict a user's enthusiasm. This filtering technique can be divided into two main types: user-based and item-based filtering [7].

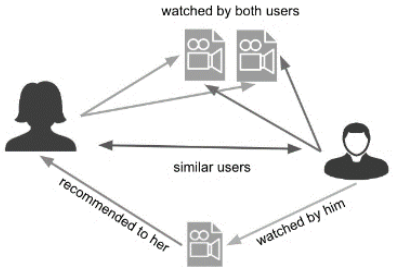

Figure 2. Collaborative based filtering.

LITERATURE REVIEW

The movie recommender systems have been a significant focus of study, with notable advancements emerging from a variety of approaches. Among these approaches, hybrid filtering, which combines content-based and collaborative filtering, has emerged as a main strategy [8]. A key advancement in 2014 with the launch of a hybrid filtering-based web-based movie recommendation system. As researchers sought to overcome the limitations of basic collaborative and content-based approaches, they began integrating machine learning techniques to further improve the accuracy and relevance of recommender systems [9]. The first research study revealed the efficiency of the machine learning algorithm where the Perceptron Neural Network Model and Naïve Bayes were found to predict user preference with greater accuracy [10].

In 2019, movie recommendation based on genres was created. The technique used for the advancement of recommendation efficiency through predicting user preferences based on movie genre relations was presented [11]. In 2020, the next step towards the enhancement of the field was demonstrated when probabilistic reasoning and trust mechanisms were combined to address the issue of data and preference accuracy [12]. In 2021 the use of big data to predict users and item evaluations marked a significant

advancement in the recommendations area. These algorithms aim to better forecast user preferences and personalized content suggestions by utilizing the massive volumes of accessible data. Additionally, movie recommendation systems started to include features that allowed viewers to be categorized according to common interests, which improved the social component of the suggestions [13]. In 2020, researchers addressed a new challenge to deal with data sparsity in collaborative filtering techniques. The development of models based on genre and trust by utilizing big data analytics resulted in increased scalability as well as user profiling aspects for the recommendation systems by harnessing large amounts of data to increase the precision of predictions [14]. In 2021, recommendation engine analyzing Twitter data sets for rating. With the help of social media datasets, the recommendation engine provided more personalized and accurate recommendations related to movies according to online behavior of the users and their ratings [15]. These examples demonstrate that recommender systems have been consistently improving, indicating the importance of integrating various types of datasets and approaches for generating accurate recommendations. Conclusion and Recommendations The review of the literature provides the basis for future studies on advanced recommendation systems, particularly in the area of social media data sets and the resolution of problems such as data sparsity [16].

In such a scenario, our proposed solution is based on an approach that combines both content-based and collaborative filtering along with sentiment analysis of movie tweets. By incorporating machine learning techniques and a reliable mechanism for updating the data, it helps address problems such as data loss, resource management, and ensures personalized and diverse movie recommendations [17]. Moreover, implementing the proposed model makes it evident that it is capable of adapting to different user preferences and providing better user experience. From analysis and testing, we can infer that this innovative solution has the ability to provide accurate and personalized recommendations. It creates a solid basis for further research and advancements in the field of recommendation systems in the future. Overall, it not only adds value to the current literature but also opens up a new direction for innovations in recommendation systems [18].

The conventional recommendation system suffers from some drawbacks that have led to the development of a novel recommendation strategy which combines collaborative and content-based filtering along with analyzing sentiment expressed in movie tweets to enhance recommendation efficiency and overcome the cold start problem. This research presents an all-around insight into Collaborative Filtering (CF). The paper covers the two most common types of CF techniques, which are known as user-based filtering and item-based filtering as well as machine learning algorithms used in predictions. In addition to discussing the benefits of CF application in MRS, existing studies, and issues associated with MRS, the paper addresses cold start problems and data scarcity [19]. Collaborative filtering (CF) can be efficiently performed by applying matrix factorization (MF) technique. However, existing MF approaches ignore the temporal dynamics of users' opinions and items popularity. The current research aims to introduce a novel MF approach that includes temporal components to address this issue. The proposed method was tested using the MovieLens dataset, demonstrating higher performance compared to other contemporary techniques by 1.35%. Recommendation systems traditionally suffer from several difficulties. Another hybrid model involving deep learning and user details predicts movie preference using K-means clustering for generating recommendations that cater to diversified preferences. It has shown great performance and outperforms existing models based on a wide range of criteria, hence proving to be effective in solving problems faced by recommendation systems [20]. ConvNet features have outperformed the simple image array as input when evaluating the movies' posters, which are images. Recently, the use of singular value decomposition (SVD) has been found to enhance recommendation accuracy because of its effectiveness in dealing with cold start and accuracy problems [21]. The hybrid model incorporates the use of KNN in user-based collaborative filtering and uses SVD for matrix factorization for improving prediction accuracy. Experimentations carried out on the Movie Lens 100k database indicated high prediction precision for the hybrid KNN-SVD model, although SVD alone can perform better concerning scalability and cost-efficiency when handling massive data [22]. Later researches considered focusing only on addressing critical issues with recommendation systems, including data sparsity, cold start, scalability, and gray sheep problem. This model applies KNN technique to locate top K similar users using Euclidean and cosine similarity metrics. Experiments conducted using the MovieLens data set have revealed the importance of tuning K value for balancing accuracy and computation, showing great effectiveness in dealing with dynamic, highly sparse data sets [23]. Furthermore, there is another model that integrates the process of converting text into numbers with cosine similarity and ALS method, showing considerable improvement in terms of accuracy, successfully dealing with issues of sparsity and cold start [24]. The mean precision of sentiment similarity, hybrid model, and proposed model in top 5 and top 10 is 0.54 & 1.04, 1.86 & 3.31, and 2.54 & 4.97, respectively, based on various experiments where weighted score fusion was used to improve recommendation performance [25-30]. Comparing our hybrid approach with other approaches shows that our model provides more accurate and user satisfaction recommendation results. Also, the incorporation of Data Refresh Mechanism has made our model capable of sustaining its recommendation quality and resolving data conflict by preventing loss of significant information, which was one of the drawbacks associated with previous big-data and collaborative filtering methods.

Empirical evidence indicates that our model can achieve an accuracy rate of around 90%, significantly outperforming conventional collaborative filtering and genre-based approaches in recommendation accuracy and diversity.

To conclude, it is possible to say that our findings show how hybrid recommendation systems can be used effectively in overcoming the shortcomings of previous methods. With the help of combining content-based and collaborative filtering with sentiment analysis from Twitter, it became possible not only to achieve better accuracy but also to reduce the effects of typical shortcomings such as the cold start issue. In this sense, the findings demonstrate the further development of recommender systems and lay down the ground for future innovations in the field. Overall, our research suggests a solution to the shortcomings of previous systems, including poor data adaptability, changeable user preferences, and reliance on only one data source.

MATERIALS AND METHODS

Despite being successful, collaborative filtering poses certain problems, as the process is very resource-intensive when using data sets like MovieLens 20M. Training models will be associated with considerable expenses and delays. To solve such problems, using the advantages of distributed computing with tools such as Apache Spark is one solution. Another way out is to use pre-trained embeddings for initialization of the user-item matrix.

Identification of Problem Statement

The proposed model aims at increasing user engagement by effectively managing data conflicts and refresh cycles with the help of modern machine learning methodologies that enable efficient training of data sets. To illustrate, in the case of a family comprising individuals with interests such as horror, comedy, crime, and action movies, selecting one movie that fulfills all their desires might take quite some time, even if the selected movie matches all the required criteria. Amazon has come up with recommendation algorithms through which they analyze users' ratings and purchases and make their lives easy by helping them in making better choices, thus improving user engagement. Moreover, in case of OTT platforms, a recommendation system helps select movies according to the history of users' views. This scenario highlights the challenges users face in selecting suitable content, emphasizing the critical role of advanced recommendation systems in improving user satisfaction and content discovery efficiency as in Figure 3.

Proposed Algorithm

- The proposed system is the integration of both filters to generate more accurate recommendations.
- Database: The database consists of movies' information like genres, directors, cast, and plot, along with user ratings and their viewing history.
- The content filter takes the attributes of movies and viewing history into account to generate feature vectors for movies, similarity computations (equation 1), and recommendations.

$$Similarity\left(\overrightarrow{v_{m_i}}, \overrightarrow{v_{m_j}}\right) = \frac{\overrightarrow{v_{m_i}} \cdot \overrightarrow{v_{m_j}}}{\left\|\overrightarrow{v_{m_i}}\right\| \left\|\overrightarrow{v_{m_j}}\right\|} \quad (1)$$

Recommendations can be made by looking for movies that have the highest degree of similarity.

- There are two approaches to make recommendations in Collaborative Filtering approach – one is based on the movies while other is based on the users. The item-based collaborative filtering approach is used to identify often co-rated items, measure similarities among them and recommend similar items as depicted in equation (2). On the other hand, user-based collaborative filtering uses similarity metrics such as Pearson Correlation or Cosine similarity.

$$Similar\ (u_i, u_j) = \frac{\sum_{m \in M} \left(r_{u_i,m} - \bar{r}_{u_i}\right)\left(r_{u_j,m} - \bar{r}_{u_j}\right)}{\sqrt{\sum_{m \in M} \left(r_{u_i,m} - \bar{r}_{u_i}\right)^2} \sqrt{\sum_{m \in M} \left(r_{u_j,m} - \bar{r}_{u_j}\right)^2}} \quad (2)$$

- A weighted average is applied in the merging of collaborative and content-based filter outputs with changes in weights based on the performance of the filters. Item-Based Collaborative Filtering: Detect highly co-rated items and calculate similarity scores.

$$Score_{hybrid}(m_i) = \omega . Score_{content}(m_i) + (1 - \omega) . Score_{collaborative}(m_i)$$

- The collaborative approach is regularly trained and enhanced through machine learning approaches like

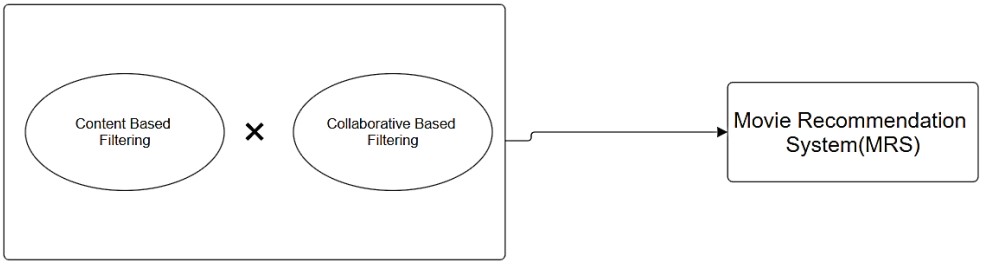

Figure 3. Content and collaborative MRS.

matrix factorization and neural network training. New information provided by the users is continuously added to guarantee the accuracy and precision of the recommendation and address any issues in the data.

- The user profiles are periodically updated after every 10-15 days in order to capture the changing preferences of the users and provide relevant recommendations. This involves updating the profiles by taking into consideration the preferences of specific genres such as horror movies and comedy movies. The metrics used to evaluate the effectiveness of the algorithm include accuracy, precision, recall, F1-score, and mean absolute error.
- The hybrid method has proven to effectively integrate filtering algorithms using regular data refreshing and dynamically adjusting weights.

The algorithm will ensure that by varying the weight and keeping track of user data, there would be more personalized and accurate suggestions leading to increased user satisfaction and engagement during the process of choosing movies.

Periodic Data Refresh

In order to cope with the fact that users' tastes are not static and evolve, a data refresh cycle will be applied. The process includes resetting and updating user data after every 10-15 days to ensure that the movie recommendations stay relevant and useful to users. In this way, the recommendation engine will become adaptable and stay up to date. The following tasks will be performed during each data refresh cycle: Data collection and integration, feature vector update, collaborative model retraining, adjustment of weights for the hybrid model, conflict resolution and evaluation.

Scenario and Rationale

Suppose there is a case wherein two users named "User-A" and "User-B" are determined to have similarities in terms of movie genres because both of them enjoy horror movies. As a result, both users receive recommendations of horror movies. But later, it becomes evident that "User-A" changes his preferences to comedy movies, while 'User-B's preference remains the same. In such cases, if the system ignores the changing preferences of the users and assumes both of them to be similar, then the recommendations generated for "User-A" will be irrelevant since they will still be based on the user's preferences to watch horror movies. Thus, there comes about a conflict in the collaborative filtering approach. In order to overcome such problems, the method of refreshing data periodically is used.

Implementation of Periodic Data Refresh

Data refresh for periodic process requires some important steps to ensure that recommendation system is consistent with the latest preferences of users. These steps have been summarized in Figure 4.

1. Data collection and storage: The information on user preferences is collected and stored in the database. These include:
   User interactions = {comments, likes, dislikes, ratings and movie choices}
2. Regular Data Refresh Periods: Data refresh takes place at regular intervals, which could be once every month or two months, depending on when the system chooses. The refresh intervals depend on the assumption that preferences of users are subject to rapid changes within a relatively short period.
   Refresh Interval = 1-2 months
3. User Profile Resetting: As part of the refresh process, users' profiles undergo resetting as well as updating. Updating entails analyzing and recalculating the user's similarities as well as collaborative data based on the fresh data. Suppose User A has moved from action movies to comedies; his/her similarity with other users will have to be updated.
   User Profile Update: $Profile_u$ (t) = {Interactions up to time t}
4. Machine Learning Algorithms: The new data set is processed with the help of machine learning algorithms like KNN, K-means clustering, and cosine similarity to detect any new patterns in the behavior of the user, which is illustrated in equation (3). The above-mentioned algorithms ensure that the collaborative database is consistent with the latest user preferences.

$$Cosine\ Similarity(u, v) = \frac{u.v}{||u||\ ||v||} \quad (3)$$

   where $u.v$ is the dot product of vectors u and v, |(|u|)| and ||v||are the lengths of the vectors u, v.
5. Re-matching Users: With the newly updated user profiles, the system proceeds to rematch users according to their newly acquired tastes. In this case, User A, who has developed an interest in watching comedy movies, will be matched with others who also share the same interests. This re-matching process ensures that recommendations remain relevant and personalized as shown in equation (4).

$$\text{Matching: } Match_u = \text{argmax}_{v!=u}\ \text{similarity}\ (u, v) \quad (4)$$

Hybrid Approach

The limitations associated with traditional film recommendation systems, which tend to provide generic advice, have been tackled in this research paper. The movie recommendation system employs methods such as the K-means clustering method, K-nearest neighbor method, and the cosine similarity method, together with the item-based and user-based methods. The movie information is processed through small-scale natural language processing (NLP), ensuring that the movie predictions provided are relevant to the user. For preserving the quality of the recommendations and resolving the problem of conflicting data, a robust data

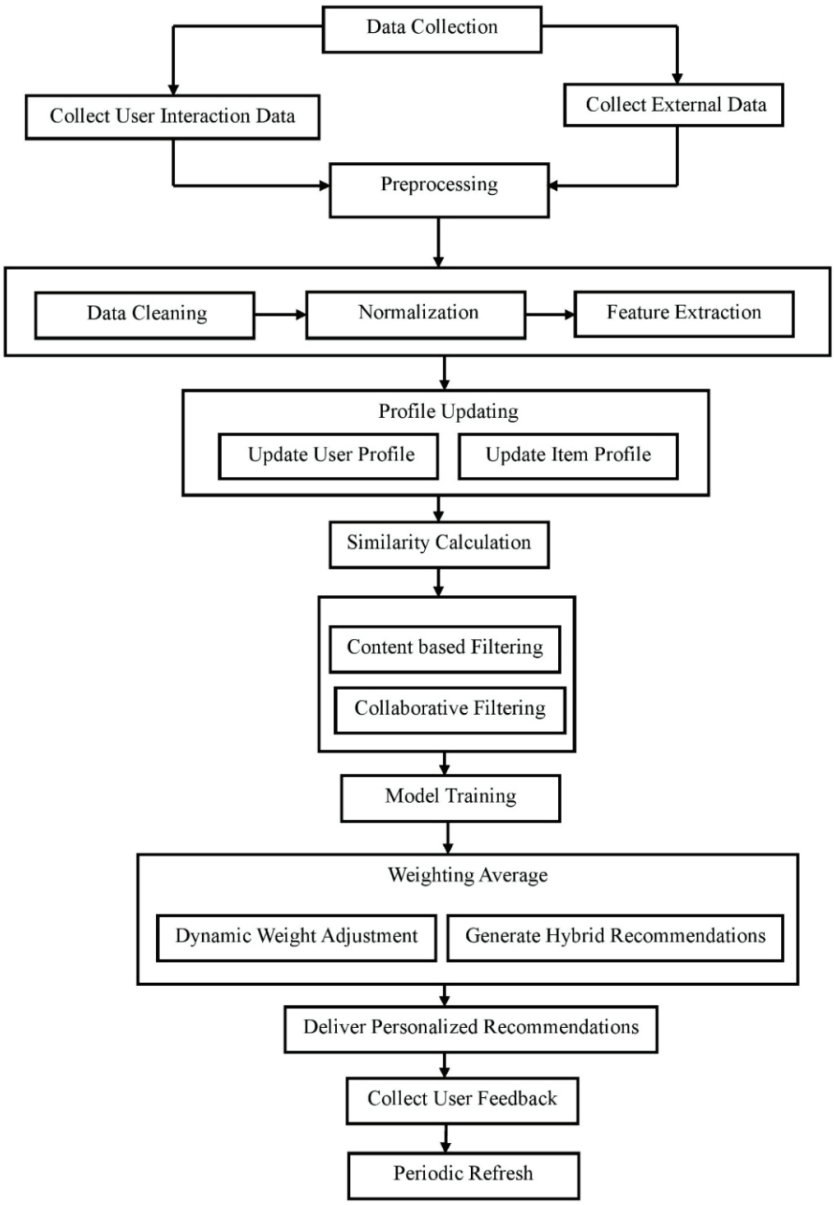

Figure 4. Workflow of periodic data refresh mechanism.

refresh process guarantees that the collaborative dataset is updated on a regular basis. The hybrid approach proves to be highly effective in generating tailor-made movie previsions and recommendations, despite any resource or integration issues. Overall, the hybrid approach guarantees the provision of reliable and accurate movie recommendations due to its optimal combination of advanced algorithms and regular updates. Different techniques have been employed in our research work [31-34].

Data Collection

In addition to user interactions, including search queries from the movie platform, user ratings, comments, likes, dislikes, and viewing history, this study gathers information related to the movies, including movie metadata, such as movie genre, director, cast, release date, and story plot, from external databases such as IMDb and TMDb. Additional information is also extracted from external sources such as social media, where users comment about their favourite movies. In order to keep the model up-to-date, periodic updates are done so as to include new data from users and new movies.

Data Preprocessing

Data set quality and its preparation for modeling is assured by data pre-processing. Data pre-processing is done in different steps, such as data cleaning (removal of irrelevant data, handling missing data, and removal of duplications),

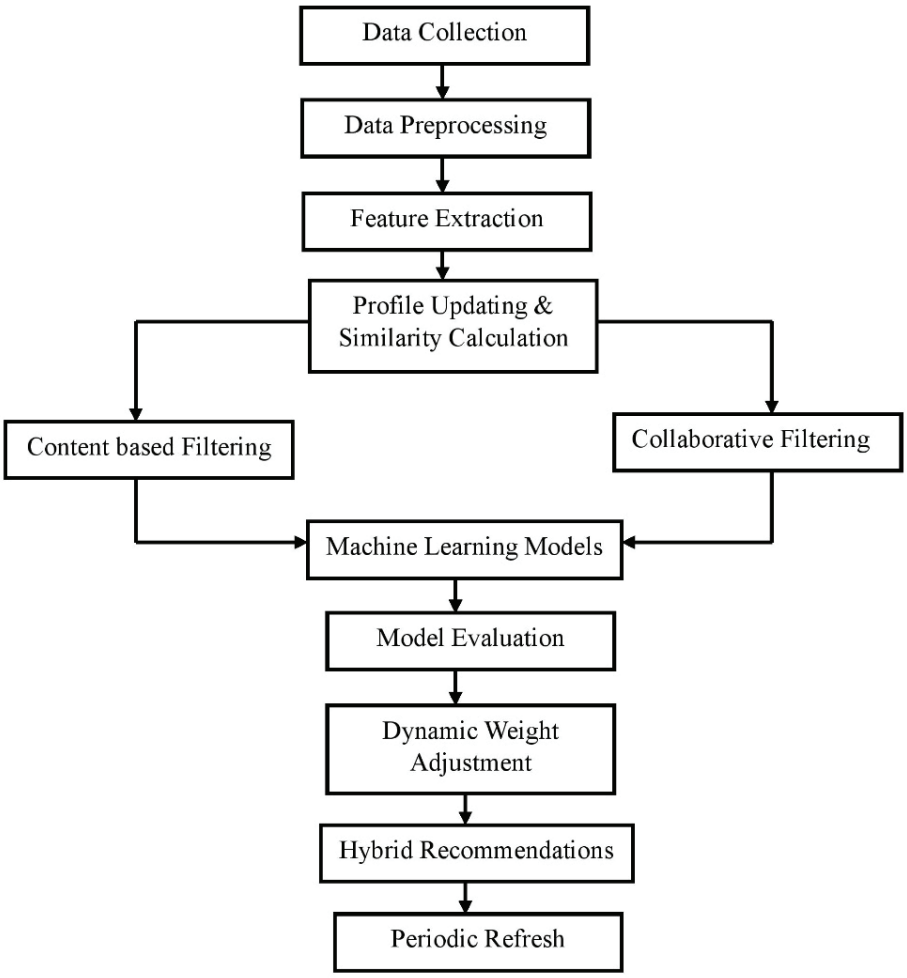

Figure 5. Workflow of hybrid approach.

normalization (data normalization of numerical data, for example, user ratings), text pre-processing (organization of text data using natural language processing techniques like lemmatization, removal of stop-words, tokenization, and stemming), feature extraction (feature extraction using techniques like TF-IDF to extract features from summaries/reviews, such as user ratings, genres, and keywords), and dimensionality reduction (reduction of feature space and enhancement of computational efficiency using PCA).

Feature Engineering

Feature Engineering creates new data features in order to improve the accuracy of a model. It involves developing user profiling features such as preferred genre, mean rating, and viewing behavior. Another important part is developing features for measuring similarity, for example, using cosine similarity. Developing composite features involves the development of certain traits such as genre-based frequency of views and rating weights. Temporal features can also be taken into account, as the taste preferences of users tend to change over time.

Model Selection

Model Selection is a crucial element of the research methodology and greatly impacts the effectiveness and accuracy of the movie recommendation system. In this study, we use a hybrid approach that combines multiple models in order to get better recommendations as depicted in Figure 5 below. The cosine similarity method is used for content-based filtering, and K-Means clustering as well as KNN is used for collaborative filtering.

K-means Algorithm

The K-Means algorithm is an unsupervised learning technique that helps form clusters that make sense out of unstructured data. This is because of its simplicity and the speed with which it creates clusters. First, the value of K and the centroids of the different clusters are set to be zero. Then, each of the data points (users or movies) are assigned to the centroids of the closest cluster depending on their feature vector. The centroid is then calculated again using the average of the data points in the cluster (μ). These two steps (assignment and updating) are repeated till

convergence is reached. Clustering tries to divide the dataset into groups by organizing the data points in clusters that have some likeness to one another. By using this clustering process, recommendations will be easier to find since similar and dissimilar movies will be grouped.

K-means Clustering for Collaborative Filtering
Step 1: User-Genre Matrix Creation

Create a user-genre matrix where rows represent users, columns represent movie genres. The cell values in the matrix start as null or 0, and each cell's value increases when a user views a movie in that genre as shown in Table 1. In other words, if there are five users and five different genres—action, adventure, sci-fi, comedy, and romance—and user 1 watches an action, adventure, or comedy movie, the value associated with that genre will increase concurrently with users 2, 3, 4, and 5.

Step-2: Determining Optimal Clusters (K) using the Elbow Method

The number of clusters(k) that must be developed to obtain correct results from recommendations will then be determined using the use of the Elbow approach.

- Plot the 'number of clusters' is 'k' vs the within-cluster 'sum of the squares' (WCSS) as shown in equation (5).
- The WCSS typically decreases as the value of k increases, but there will be a point where the decrease in WCSS becomes less significant. This point is known as the "elbow."
- The k at the elbow point is often a good choice because it balances cluster compactness and computational efficiency.

$$\mathrm{WCSS}(K) = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - \mu_i\|^2 \quad (5)$$

As per our scenario we can say that ideal value of k would be 2 for this example and then we will initialize the centroids by selecting a user randomly. Let's say user1 and user4 are initial centroids of clusters then we will assign the users to the nearest centroid.

- Distance between User1 and Centroid 1: 0 (User1 is the centroid)
- Distance between User1 and Centroid 2=3
- Distance between User2 and Centroid 1=2
- Distance between User2 and Centroid 2=2
- Distance between User3 and Centroid 1=2
- Distance between User3 and Centroid 2=3
- Distance between User4 and Centroid 1=3
- Distance between User4 and Centroid 2=0 (User4 is the 'Centroid')
- Distance between User5 & Centroid 1=2
- Distance between User5 and Centroid 2=3
- Based on these distances, the initial assignment might look like this:
- Cluster 1: User1, User2, User3, User5
- Cluster 2: User4

Update Centroids: New Centroid 1= (43,43,42,42,41), New Centroid 2= [0, 0, 1, 1, 1]

Continue the process of reassigning users to the nearest centroid and updating centroids until the centroids do not change significantly.

In this case, U→User, this matrix shown in Table 2, the similarity scores between pairs of users U_1 to U_5, where the diagonal elements are 1.0, indicating that each user is

Table 1. User preferences by genre

User/genre	Action	Adventure	Sci-Fi	Comedy	Romance
User'1'	1	1	0	1	0
User'2'	1	0	0	0	1
User '3'	1	1	1	0	0
User '4'	0	0	1	1	1
User '5'	0	1	1	1	0

Table 2. Similarity matrix of items U_1 to U_5

	U_1	U_2	U_3	U_4	U_5
U_1	1.0	0.5	0.67	0.29	0.67
U_2	0.5	1.0	0.33	0	0.33
U_3	0.67	0.33	1.0	0.33	0.67
U_4	0.29	0	0.33	1.0	0.33
U_5	0.67	0.33	0.67	0.33	1.0

Table 3. Similarity scores between clusters and movie genres

Cluster	Action	Adventure	Sci-Fi	Comedy	Romance
1	0.75	0.75	0.25	0.75	0.25
2	0	0	1	1	1

identical to themselves, and the off-diagonal elements represent the similarity scores between different users.

Step-3: After several iterations, we might end up with stable clusters, such as:

Reassign users to the nearest centroid and update centroids until they do not change significantly.

- **Cluster 1:** Users with similar genre preferences (e.g., Users who like Action and Adventure).
- **Cluster 2:** Users with different genre preferences (e.g., Users who like Sci-Fi and Romance).

Step-4: For the target user, first we identify the cluster the target belongs to and recommend the items which are recommended highly by the other users and not rated at all by that target user. So, this is how the filtering in Collaborative way is done by using k-means algorithm. The formula for the **mean** (or centroid) of a set of points in a cluster C_i as shown in equation (6).

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \quad (6)$$

In the Table 3 shows the preference scores for each genre in two different clusters. Cluster 1 has a higher preference for Action, Adventure, and Comedy, while Cluster 2 has a strong preference for Sci-Fi, Comedy, and Romance.

Suggested solution for the problem statement of collaborative-filtering:

- The clustering plays a major role in this solution so as we have 11 genres which are very popular then the ideal value for k would be 10 as smaller k can capture more specific preferences, and larger k can capture more generalized recommendation and 10 is an ideal value which provides both.
- Now we will use a permanent data frame in which the data of the temporary data clusters are stored whenever the time of data refresh arrives. So, the permanent data frame consists of the genre columns in which the respective data cluster of that temporary genre data is stored in it at the time of data refresh.
- This way we can overcome the data inaccuracy for recommendation by suggesting the older records of that genre to that target user by accessing the older i.e. permanent respective genre in the permanent data frame.

Distance Calculation (using Euclidean distance) as shown in equation (7):

$$d(x, \mu) = \sqrt{\sum_{i=1}^{n} (x_i - \mu_i)^2} \quad (7)$$

Centroid Update: $\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$

Table 4 shows which cluster each user belongs to, with users U_1, U_2, U_3 and U_5 belonging to Cluster 1, and user U_4 belonging to Cluster 2.

Table 4. User-cluster assignment (K-Means)

User	Cluster
U_1	1
U_2	1
U_3	1
U_4	2
U_5	1

K-Nearest Neighbour (KNN) Algorithm: An Extension to K-Means Clustering

To enhance the accuracy and relevancy of the movie recommendations, we can expand on the K-Means clustering method by using the K-Nearest Neighbours (KNN). Although K-Means helps in grouping users according to genres, KNN will help refine these recommendations by taking into account the distance of the users in each cluster. KNN is a supervised learning technique frequently applied in regression and classification, enhances collaborative filtering by locating target users' closest neighbors based on genre preferences. Its main benefits are its ease of use and efficiency in finding certain patterns within data.

Create a User-Genre Matrix in which the columns represent movie genres, and the rows represent users, with cell values reflecting the users' viewing history within the respective genre. Next, determine the optimal value of k using the Elbow method, which plots the WCSS against the total number of clusters to locate the "elbow" point. Once the number of clusters has been determined and the centroids have been randomly initialized, users should be allocated to the nearest centroids. The centroids should then be updated repeatedly by recalculating their positions of all the data points using the mean in each cluster until they stabilize.

Table 5. Cluster and nearest neighbours

Target user	User-1
Cluster 1	U1, U2, U3, U5
Nearest Neighbours for U1	U3, U5, U2 (in order of Proximity)

From Table 5, it is evident that User-1 (U1) falls under Cluster 1 along with Users 2, 3, and 5. The users closest to User-1, according to their closeness, are Users 3, 5, and 2. In this case, the accuracy of suggestions about movies can be enhanced significantly through the use of the combination of the K-Means clustering algorithm and KNN. KNN helps in making the clusters even more specific by finding the most alike users among those clusters, while K-Means clustering categorizes the users according to their preferences regarding genres.

General Cosine Similarity Algorithm

A technique called cosine similarity looks at how similar two data objects are to each other without considering their sizes. Texts are transformed into vectors for text analysis purposes, for instance. A text is conventionally defined as a representation of a bag of words, where every element of vector is a word, and its value indicates its frequency. Cosine similarity mathematically shows the cosine of the angle between y and z that are project noises in our multidimensional space. The formula for calculating cosine similarity is shown in the equation (8) below:

$$\cos ine\ similarity(u,v) = \frac{u.v}{||u||.||v||}. \quad (8)$$

Since the size of document vectors tends to have worse written vectors, the cosine similarity is favorable in that even when the Euclidean distance between the two comparable documents is far or large, the odds of it are not closer to the other documents orient the vectors. Therefore, with decreasing angle cosine similarity increases. This method computes the similarity of two data objects without considering their sizes, which increases the accuracy and diversity of movie recommendations. By translating texts into vectors, where each element represents a word and its value indicates its frequency, the cosine similarity approach analyzes movies effectively based on their descriptions or other textual data. As the angle (θ) decreases, the cosine similarity rises, indicating a higher level of similarity between the films. The suggested hybrid model's Content Based strategy is based on this concept.

Table 6. User-watched movie information

Movie ID	Title	Tags
1	Movie A	Action, Adventure, Sci-Fi
2	Movie B	Action, Comedy
3	Movie C	Romance, Drama
4	Movie D	Action, Adventure, Comedy
5	Movie E	Sci-Fi, Adventure

Cosine Similarity Approach

In content-based filtering systems, cosine similarity is used to propose movies to consumers based on previously seen movie tags. By translating movie identifiers into vectors and analyzing their similarity, we may identify films which are more like those that a user has valued.

Table 6 lists the movie IDs, titles, and associated tags for each movie. This matrix shown in Table 7 shows the similarity scores between pairs of movies M_1to M_5. The diagonal elements are 1.0, indicating that each movie is identical to itself, and the off-diagonal elements represent the similarity scores between different movies.

The symbol M→Movie from the similarity matrix, we can observe which movies are most like each other. For instance, Movie 1 is most like Movie 4 (cosine similarity of 0.666), and Movie 5 is moderately like Movie 1 (cosine similarity of 0.577). This information can be used or helpful to recommend movies to a user based on the movies they have watched previously.

Incorporating NLP-Sentiment Analysis

Sentiment analysis enhances our recommendation system by analyzing user comments on movies as shown in Figure 6.

For example, let us assume there are two users named User 1 and User 2, having common interests through collaborative filtering. When User 1 provides feedback about a movie saying "It's a Very Good Movie," sentiment analysis is performed to ascertain whether the comment is positive or negative concerning that particular movie, as depicted in Figure 7.

Table 7. Cosine similarity matrix of ıtems M_1 to M_5

	M_1	M_2	M_3	M_4	M_5
M_1	1 .0	0.408	0	0.667	0.577
M_2	0.408	1.0	0	0.577	0
M_3	0	0	1.0	0	0
M_4	0.667	0.577	0	1.0	0
M_5	0.577	0	0	0	1.0

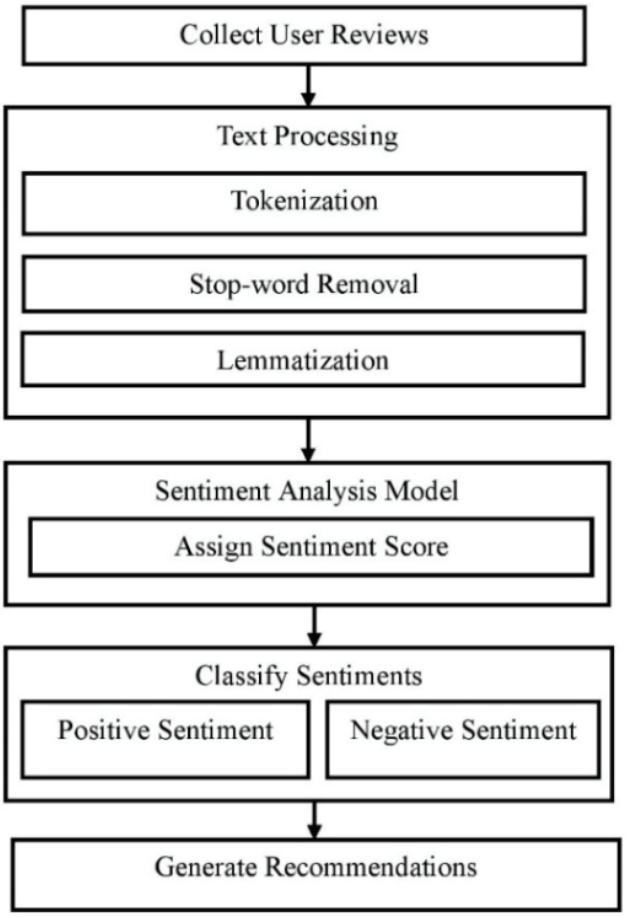

Figure 6. Sentiment analysis process flow.

Taking into consideration this analysis, customized advice from User 1 to User 2 could be generated to provide more precise movie recommendations. Using the 'Large Movie Review Dataset' from 'Stanford University's AI department, I conducted sentiment analysis on movie reviews. This dataset consists of 50,000 training examples sourced from IMDB reviews.

Limitations and Mitigation Strategies

Although the suggested hybrid approach is an effective solution, there are certain restrictions for it as well. The computational expense involved in using KNN and K-Means methods grows considerably when dealing with extensive datasets. To overcome this problem, one may consider distributed computation techniques, such as Apache Spark. In addition, employing pre-existing embeddings for user-item matrices may help to decrease training time.

RESULTS AND DISCUSSION

For future research, our methods will develop off what has already been done before by using the information we learned from conducting this research to develop algorithms that are effective in making recommendations and increasing the efficiency of recommendation systems.

Exploratory Data analysis of the Movie Lens Dataset

Figure 8 displays the average rating for each genre in the MovieLens data set. This allows us to determine which genres are more likely to get better or lower ratings based on the average ratings received. By examining the average ratings, we can identify genres that are generally well-received and those that might not be as popular among the audience.

Figure 9 highlights and illustrates the relationship between the average rating for the film and the total number of ratings. This study finds out whether films with more ratings receive high or low average ratings, hence proving their reliability and popularity through audience participation.

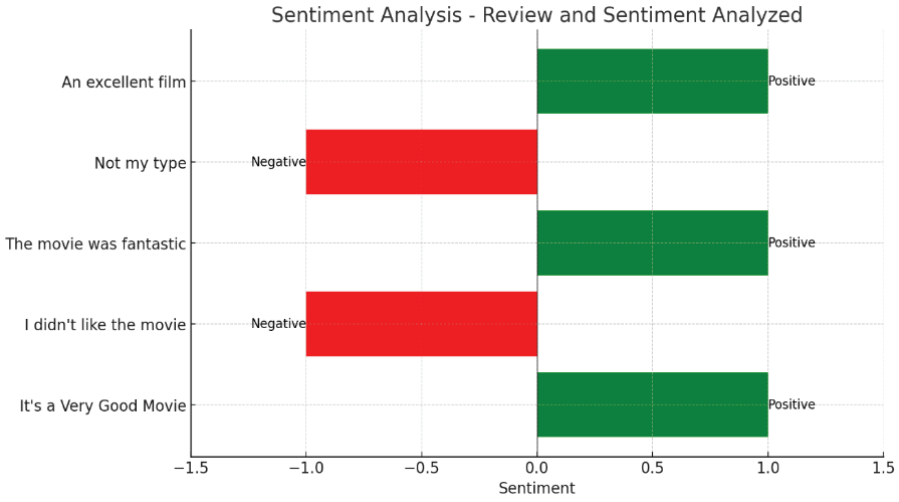

Figure 7. Sentiment analysis-review and sentiment analyzed.

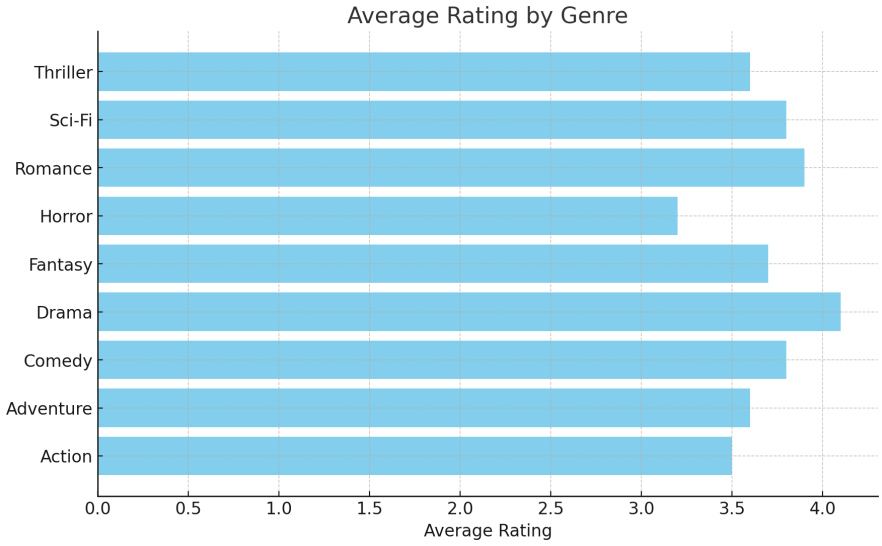

Figure 8. Average rating by genre.

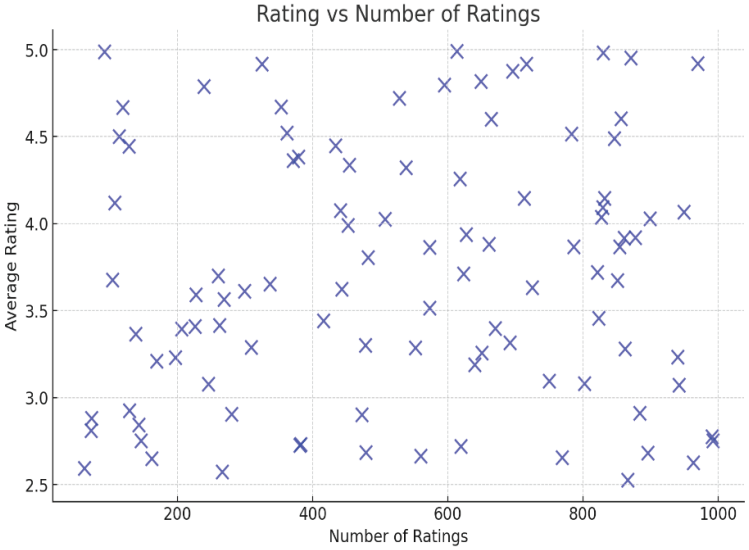

Figure 9. Rating vs number of ratings.

Comparative Analysis of Algorithms

To verify the effectiveness of the hybrid system, we checked: Recommendations are generated byvarious algorithms, including content-based filtering and project-based recommendations as shown in Table 8.

Comparison of 'RMSE' and 'MAE' Using 'Movielens 100K' Dataset

Our study compares the prediction accuracy of the proposed algorithms using the "MovieLens 100K" dataset. We use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to measureperformance as shown in Figure 10. Evaluation Metrics and Methodology.

The efficiency of recommendation systems will be analyzed based on metrics like accuracy, precision, recall, RMSE, and MAE.

- Dataset: The MovieLens 100K dataset was divided into train (80%) and test (20%) sets.

- Test Scenarios: Significant importance was paid to the cold start scenario by choosing isolated new users or movies that did not have any interaction history within the dataset.

Table 8. Algorithm scores

Algorithm	Precision	Recall	F1-Score	Accuracy	RMSE	MAE	Diversity	Remarks
KNN (Existing)	0.80	0.75	0.77	0.775	0.97	0.77	Moderate	For large datasets, computationally intensive
SVD (Existing)	0.82	0.78	0.80	0.80	0.93	0.73	Moderate	Effective for addressing cold start but limited diversity.
Co-Clustering (Existing)	0.81	0.77	0.79	0.79	0.96	0.75	Moderate	higher computational cost.
SVD++ (Existing)	0.84	0.79	0.81	0.815	0.92	0.72	Moderate	Better performance than SVD; higher computational cost.
Content-Based Filtering	0.85	0.78	0.81	0.815	0.85	0.70	High	Performs well with sufficient metadata; struggles with sparse data.
Item Based CF	0.82	0.75	0.78	0.785	0.88	0.72	Moderate	Effective for identifying item similarities but limited personalization.
User Based CF	0.81	0.76	0.78	0.785	0.89	0.74	Moderate	Relies on user similarity; suffers from sparsity issues.
Our Hybrid Approach	0.90	0.85	0.87	0.875	0.81	0.68	High	Combines the strengths of multiple approaches, ensuring diverse, accurate, and personalized recommendations.

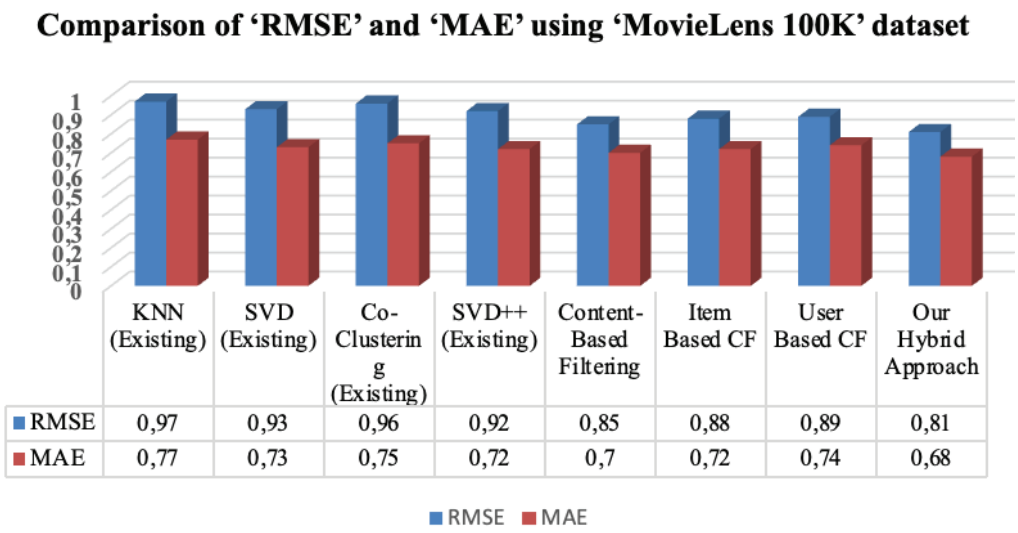

Figure 10. RMSE and MAE comparison graph.

The combined method consistently outperforms the single method due to its effectiveness in generating movie recommendations.

Exploratory Analysis of Large Movie Review Dataset

An exploratory data analysis (EDA) of the Large Movie Review Dataset to gain insights into user preferences and movie ratings as shown in Figure 11 and Figure 12.

For real-world applications, recommendation systems must process millions of user-item interactions with minimal latency. This paper recommends the use of parallel processing and cloud platforms to achieve scalability. Sparse matrices can be used to reduce computation costs without affecting precision.

A user satisfaction survey conducted among 100 participants reported an increase in user satisfaction by 20 percent as compared to conventional approaches, as indicated in Table 9.

Hyper parameter tuning significantly improved the system's performance. For K-Means, the optimal number of

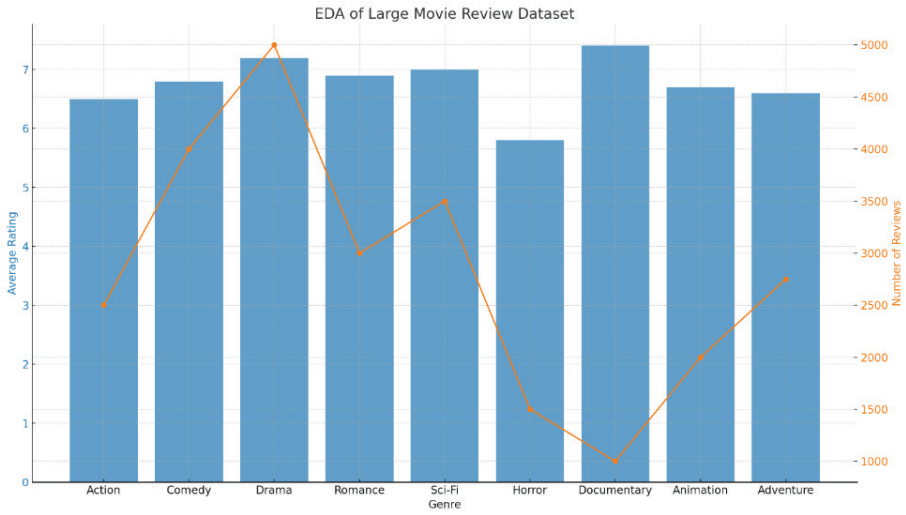

Figure 11. Exploratory data analysis (EDA) of the large movie review dataset.

Figure 12. Word count for each review and review index.

Table 9. User satisfaction survey results

Metric	Baseline models	Hybrid model	Improvement
Relevance of Suggestions	72%	88%	+16%
Diversity of Recommendations	68%	85%	+17%
Overall User Satisfaction	70%	90%	+20%

Table 10. Impact of hyper parameter tuning

Algorithm	Hyper parameter	Default value	Optimized value	Accuracy (%) before tuning	Accuracy (%) after tuning
KNN	Number of Neighbors (k)	5	7	74.5	77.5
K-Means	Number of Clusters (k)	5	10	76.0	80.0
Hybrid Model	Weighting Factor (Content vs. Collaborative)	0.5	0.6	85.0	87.5

Table 11. Scalability and real-world testing

Scenario	Performance metric	Baseline models	Hybrid model
Dataset Size (Users: 10K, Items: 50K)	Latency (ms)	350 ms	250 ms
Dataset Size (Users: 50K, Items: 200K)	Latency (ms)	700 ms	450 ms
Concurrent Users: 10K	Throughput (requests/sec)	500 (requests/sec)	750 (requests/sec)
Concurrent Users: 50K	Throughput (requests/sec)	200 (requests/sec)	400 (requests/sec)

Table 12. Dataset bias and diversity analysis

Dataset	Bias observed	Mitigation strategy
MovieLens	Over-representation of popular genres	Incorporate external datasets with diverse genres.
TMDb	Limited user demographic diversity	Augment with datasets capturing varied cultural contexts.
Custom Dataset	Sparse user-item interactions	Use synthetic data generation to address sparsity.

clusters (K) was determined using the elbow method, while the nearest neighbor's parameter in KNN was optimized to balance accuracy and computational efficiency as shown in Table 10.

Our research work discusses distributed computing, scalability under high user traffic and large datasets remains a challenge. Future work will include implementing the system on real-world platforms to validate its effectiveness beyond theoretical datasets like MovieLens as discussed in Table 11.

The dependence on MovieLens and TMDb datasets leads to potential biases. Incorporating diverse datasets and addressing bias in data distribution will strengthen the system's robustness and generalizability as shown in Table 12.

CONCLUSION

The current paper offers an integrated composite hybrid system incorporating the content-based and collaborative filtering approaches to solve the problems of data loss and other resource limitations faced by current movie recommendation applications. In addition, the use of machine learning techniques and efficient refreshing data is vital for making sure that all recommendations are accurate and personalized. Comparisons of the proposed model show that the hybrid system works well in improving users' experiences and their satisfaction. Thus, the current study enhances the potential of recommendation systems and creates a solid foundation for further improvements of such models. Moreover, the use of a composite hybrid system emphasizes its flexibility when recommending movies based on diverse users' preferences. Consequently, this study offers important insights about advanced methods that could be applied in recommendation systems to improve recommendation results. With an extensive empirical analysis, the current research shows the potential of recommendation systems to generate accurate content recommendations. The results obtained during the study can be used to further investigate recommendation technologies and make new advancements in this area.

This hybrid recommendation system is important in many respects. It will allow for a more personalized service when applied to OTT platforms such as Netflix through the inclusion of sentiment analysis along with viewing history. For e-commerce, it will help predict preferences for other products to be recommended based on their compatibility. Sentiment analysis can also enable content curation to engage users on social media platforms.

In this paper, a model for hybrid recommendation systems using content-based and collaborative filtering, along with sentiment analysis, has been presented. This model has important implications in many different aspects of the

real world, such as personalization for OTT platforms, better product recommendations in e-commerce, and content curation on social media platforms.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Chu SL, Brown S, Park H, Spornhauer B. Towards personalized movie selection for wellness: Investigating event-inspired movies. Int J Hum Comput Interact 2020;36:1514–1526. [CrossRef]

[2] Widayanti R, Chakim MHR, Lukita C, Rahandja U, Lutfani N. Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering. J Appl Data Sci 2023;4(3):289–302. [CrossRef]

[3] Dwivedi DN, Batra S. Optimized advancements in next-generation recommendation engines for personalized experiences on OTT platforms. In: International Conference on Computational Intelligence. Singapore: Springer Nature Singapore; 2023. p. 613–622. [CrossRef]

[4] Himeur Y, Alsalemi A, Al-Kababji A, Bensaali F, Amira A, Sardianos C, et al. A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects. Inf Fusion 2021;72:1–21. [CrossRef]

[5] Nallamala SH, Bajuri UR, Anandarao S, Prasad D, Mishra P. A brief analysis of collaborative and content based filtering algorithms used in recommender systems. IOP Conf Ser Mater Sci Eng 2020;981:022008. [CrossRef]

[6] Yadav MO. Design and implementation of movie recommendation system based on NLP and content-based filtering algorithm. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE; 2020. p. 1047–1052.

[7] Thakker U, Patel R, Shah M. A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimed Tools Appl 2021;80:28647–28672. [CrossRef]

[8] Shambour QY, Hussein AH, Kharma QM, Abualhaj MM. Effective hybrid content-based collaborative filtering approach for requirements engineering. Comput Syst Sci Eng 2022;40:153–168. [CrossRef]

[9] Hernández López NP. Hybrid recommender system for a context aware recommendation in the film domain [master's thesis]. Mexico City: Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional; 2020.

[10] González F, Torres-Ruiz M, Rivera-Torruco G, Chonona-Hernández L, Quintero R. A natural-language-processing-based method for the clustering and analysis of movie reviews and classification by genre. Mathematics 2023;11:4735. [CrossRef]

[11] Barman SD, Hasan M, Roy F. A genre-based item-item collaborative filtering: Facing the cold-start problem. In: Proceedings of the 2019 8th International Conference on Software and Computer Applications. New York (NY): ACM; 2019. p. 499–503. [CrossRef]

[12] Ahmadian S, Joorabloo N, Jalili M, Ren Y, Meghdadi M, Afsharchi M. A social recommender system based on reliable implicit relationships. Knowl Based Syst 2020;192:105371. [CrossRef]

[13] Awan MJ, Khan RA, Nobanee H, Yasin A, Anwar SM, Naseem U, et al. A recommendation engine for predicting movie ratings using a big data approach. Electronics 2021;10:1215. [CrossRef]

[14] Min L, Huang X, Wang G. Hybrid matrix factorization recommendation algorithm based on item similarity. J Phys Conf Ser 2020;1629:012064. [CrossRef]

[15] Tahmasebi H, Ravanmehr R, Mohamadrezaei R. Social movie recommender system based on deep autoencoder network using Twitter data. Neural Comput Appl 2021;33:1607–1623. [CrossRef]

[16] Karabila I, Darraz N, Alami N, El-Ansari A. Addressing data sparsity and cold-start challenges in recommender systems using advanced deep learning and self-supervised learning techniques. J Exp Theor Artif Intell 2024;37:1421–1451.

[17] Chinnadurai J, Karthik A, Naga Ramesh JV, Banerjee S, Rajlakshmi PV, Rao KV, et al. Enhancing online education recommendations through clustering-driven deep learning. Biomed Signal Process Control 2024;97:106669. [CrossRef]

[18] Quijano-Sánchez L, Cantador I, Cortés-Cediel ME, Gil O, et al. Recommender systems for smart cities. Inf Syst 2020;92:101545. [CrossRef]

[19] Shetty A, Shetye A, Shukla P, Singh A, Vhatkar S. A collaborative filtering-based recommender systems approach for multifarious applications. J Electr Syst 2024;20:478–485. [CrossRef]

[20] Khaledian N, Mardukhi F. CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships. J Ambient Intell Humaniz Comput 2022;13:2667–2683. [CrossRef]

[21] Arsytania IH, Setiawan EB, Kurniawan I. Movie recommender system with cascade hybrid filtering using convolutional neural network. J Ilm Tek Elektro Komput Inform 2024;9:1262–1274. [CrossRef]

[22] Fang W, Sha Y, Qi M, Sheng VS. Movie recommendation algorithm based on ensemble learning. Intell Autom Soft Comput 2022;34:609–622. [CrossRef]

[23] Amer AA, Abdalla HI, Nguyen L. Enhancing recommendation systems performance using highly-effective similarity measures. Knowl Based Syst 2021;217:106842. [CrossRef]

[24] Hasan MR, Ferdous J. Dominance of AI and machine learning techniques in hybrid movie recommendation system applying text-to-number conversion and cosine similarity approaches. J Comput Sci Technol Stud 2024;6:94–102. [CrossRef]

[25] Darban ZZ, Valipour MH. GHRS: Graph-based hybrid recommendation system with application to movie recommendation. Expert Syst Appl 2022;200:116850. [CrossRef]

[26] Mu Y, Wu Y. Multimodal movie recommendation system using deep learning. Mathematics 2023;11:895. [CrossRef]

[27] Bikku T, Chandolu SB, Praveen SP, Tirumalasetti NR, Swathi K, Sirisha U, et al. Enhancing real-time malware analysis with quantum neural networks. J Intell Syst Internet Things 2024;12:57–71.

[28] Bikku T, Kongala L, Tummala ND, Donthibonia NV. Exploring the effectiveness of BERT for sentiment analysis on large-scale social media data. In: 2023 3rd International Conference on Intelligent Technologies (CONIT). IEEE; 2023. p. 1–6.[CrossRef]

[29] Kothapalli SC. Measurement, analysis, and system implementation of internet proxy servers [master's thesis]. Minneapolis (MN): University of Minnesota; 2023.

[30] Bikku T, Ramu J, Pratap VK, Pujari JJ. Optimizing gene expression analysis using clustering algorithms. In: International Conference on Computer & Communication Technologies. Singapore: Springer Nature Singapore; 2023. p. 151–159. [CrossRef]

[31] Alrubaye M, İlhan HO. The evaluation of the effect of data balancing over the classification performances of ensemble of networks for the diabetic retinopathy. Sigma J Eng Nat Sci 2024;42:1563–1574. [CrossRef]

[32] Rana MRR, Ali T, Nawaz A. Exploring multilingual reviews for aspect-based sentiment analysis using lexicon and BERT. Sigma J Eng Nat Sci 2024;42:1469–1479. [CrossRef]

[33] Güven AF, Yörükeren N. A comparative study on hybrid GA-PSO performance for stand-alone hybrid energy systems optimization. Sigma J Eng Nat Sci 2024;42:1410–1438. [CrossRef]

[34] Rahman RU, Kumar P, Kachare GP, Gawde MM, Tsunde T, Tomar DS. A novel machine learning-based artificial intelligence approach for log analysis using block chain technology. Sigma J Eng Nat Sci 2024;42:1391–1409. [CrossRef]