International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I10P119

Volume 72 Issue 10, 191-203, October 2024

Original Article

Enhancing Movie Recommendation using Ensemble based Machine Learning Approach

Sneha Bohra[1], Amit Gaikwad[2]

[1,2]*Department of Computer Science and Engineering, G H Raisoni University, Amravati, Maharashtra, India.*

[1]*Corresponding Author : snehab30@gmail.com*

Received: 21 May 2024 Revised: 27 September 2024 Accepted: 08 October 2024 Published: 25 October 2024

Abstract *- Traditional Recommendation Systems suffer from Concept Drift- a phenomenon that assumes that user preferences are static over time. To address this issue, there is a need for a recommendation algorithm that takes into account time-sensitive shifts in user preferences and offers relevant recommendations. This research work proposes an Ensemble-based Hybrid Recommendation System which incorporates temporal variations in user interests. The proposed system combines distinct algorithms such as Popularity, Clustering, Collaborative Matrix Factorization and Singular Value Decomposition (SVD). The movie recommendations obtained from these individual models are then combined and classified using Artificial Neural Network (ANN). User feedback on the presented batch of recommendations is then recorded, which contributes to the calculation of the relevancy factor for each batch of recommendations. Finally, the user is provided with relevant movie recommendations. In case of a lower relevancy factor, the recommendations are reclassified. The objective of the proposed system is to provide diverse recommendations to the user depending on his time-sensitive preferences. The novelty of the proposed research is the integration of the popular recommendation strategies and the incorporation of the user feedback mechanism on the presented recommendations. The proposed system is implemented on the standard movie dataset Movielens-25m and is evaluated using statistical performance metrics like RMSE and MAE. The experiments show that the integration of an Artificial Neural Network as a classifier to the Ensemble- Hybrid Recommendation Model demonstrates promising results in terms of providing relevant recommendations with 0.56 and 0.43 as RMSE and MAE values. The work illustrates the improvement in recommendation quality and increased adaptability to varying user preferences.*

Keywords *- Movie recommendation system, Collaborative filtering, Content-based filtering, Hybrid approach, Singular Value Decomposition (SVD), Neural Network (NN).*

1. Introduction

A recommendation system is a machine learning based information filtering personalized information. Recommender systems are vital for enhancing the user experience across a range of online platforms, such as social media, streaming services, and e-commerce websites. These systems collect and keep track of user interaction, user preferences, and user behaviour, which includes the user's recent purchases, browsing history, product ratings, and social connections. The user details are then processed to extract essential insights, and personalized recommendations for the user are generated. Recommendation algorithms are categorized as Content-based filtering (Con BF), Collaborative Filtering (Coll F) and Hybrid algorithms. Content filtering recommends items that are relevant to the user's preferences utilizing the item's attributes. The idea behind this approach is that it is based on the resemblance between the attributes of the user and the item. For example, if a user has watched a thriller movie, the recommender system would recommend the user with the movies having the same genre or cast. The collaborative filtering algorithms suggest items on the basis of user preferences. This recommendation strategy uses analogies of users' preferences and acquires knowledge for predicting potential interactions on the basis of prior interactions between users and items. This recommender algorithm creates a framework that records the user's history, like his previously purchased items, his ratings of such items and preferences of other users for similar items. The principle of the algorithm states that depending on whether the user has previously made similar decisions and purchases, the probability of choosing other similar items is higher. Hybrid recommendation algorithms are a blend of content-based and collaborative filtering to overcome the drawbacks of individual techniques and achieve the maximum benefits of the combined approach. A Movie Recommender System (MRS) is a machine learning approach that predicts the users' movie preferences based on their prior movie choices and interactions. MRS can be modelled using content-based, collaborative-based filtering or a combination of both. This study introduces the multimodel-based hybrid movie recommendation system. The proposed movie recommendation system assumes that the user preferences are subject to change over time and, therefore, considers the user's feedback over the recommendations generated in order to suggest novel and relevant movie recommendations. The paper is arranged into 5 sections. Section 1 presents an introduction. Section 2 provides a brief review of the latest hybrid movie recommendation systems. Section 3 provides the details about the dataset and proposed methodology. Section 4 discusses the obtained results and compares them with the existing literature. Section 5 proposes the conclusion and future scope.

2. Related Work

This section discusses related work, the research gap and the objective of this research work. Ravi Kumar et al. [1] introduced a hybrid recommendation system for movies that combined weighted average and min-max scalar to evaluate the movie ratings and popularity. It created vectors from the data using TF-IDF and used cosine similarity to determine the similarity among these vectors. The Movies dataset was used to create the recommender system, and the findings demonstrated that the suggested system could efficiently predict a movie's rating. The observed value for ranking RMSE is 1.117.

The combination of weighted average and min-max scalar helped in providing more accurate prediction by balancing different aspects of the data, such as movie ratings and popularity. The use of TF-IDF and cosine similarity ensured that the system identified similar movies based on their metadata or content. The proposed hybrid model addressed issues related to data sparsity by combining multiple techniques. The integration of multiple techniques added to system complexity and the requirement for more computational resources. The system focused on content-based similarity and did not fully capture the social and behavioral factors. The use of weighted average ratings could result in biased recommendations and cold-start problems.

C. Wu et al. [2] presented a work for movie recommendation on Movielens and HetRec2011 datasets in which the user embedding vector was dynamically adjusted first for various target items based on the content of the target item. The system assumed that user preferences changed with time, and in order to emphasize the user's most recent preferences, the work took into account two time-decay functions based on the attention model. The first decay function considered the most recent rating, and the second function took into account the recently released movies. The combination of latent factors of these two decay functions is used for movie rating prediction. This system was evaluated over RMSE and MAE, and the values obtained for these metrics were 0.83 and 0.64. This system offered significant strengths in adaptability and relevance and effectively captured the shifting user preferences, thereby enhancing the quality of recommendations. However, the model's complexity and computational overhead complicated the implementation with larger datasets. Additionally, it faced challenges such as the cold start problem for new users or items.

H Wu et al. [3] presented a research article that examined the next-item recommendation for a cold-start scenario. The system addressed the issue with the use of Zero-Shot Learning (ZSL) which categorized items not included during training of the model. It also presented a new model called User-Item Matching and Auto-encoders (UIMA). It utilized the user history and item attributes to learn the latent factors for both users and items. UIMA consisted of two auto-encoders for learning user and item latent features and a corresponding connection for investigating the connections between the learned user and item embedding.

The experimental results were presented for LastFM, Movielens and Delicious datasets and were evaluated on parameters like Precision, NDCG and hit Ratio with remarkable value for these metrics. This work offered notable advantages by addressing the cold-start problem through zero-shot learning. The utilization of user history and item attributes for learning the latent factors enabled the model to capture complex user-item relationships through a dual auto-encoder architecture, which led to improved recommendation accuracy across multiple datasets. However, the model was found to be complex for implementation.

The movie recommendation system proposed by L. Vuong Nguyen et al. [4] addressed the issue of data scarcity and the user ratings' cold start by combining word embedding-based content analysis and collaborative filtering techniques. It focused on several movie features like titles, genres, directors, actors, and plots. The work presented analyzed the word embedding in movie plots to find a similar plot among other movies in the dataset. The experiments were conducted on self-generated user embedding over the IMDB dataset and were evaluated over performance metrics like Precision, Recall and the obtained values for metrics were 0.834 and 0.773 respectively. The system tackled data scarcity and the cold-start problem by combining word embedding-based content analysis with collaborative filtering. This hybrid approach enhanced the model's ability to recommend movies with similar plots, even for users with limited rating histories. However, the complexity of combining content-based and collaborative methods increased computational costs.

The work by S Hwang et al. in [5] suggested a solution for the traditional recommendation system in place in South Korea that implemented a collaborative filtering technique utilizing the genres for suggesting movies. The system in place failed to suggest when the user demanded the recommendation on the basis of either actors or directors. This work implemented a content-based algorithm on the actor's details and genres over the South Korean Movie dataset. The system was evaluated on the correlation between actor-genre and movie-genre using Pearson co-efficient. The work gave results for limited features and datasets. The work was presented on the South Korean dataset and provided improved recommendation accuracy for specific user demands. However, the system was limited by a small dataset and feature set, which constrained its broader applicability.

Another work in [6] suggested a hybrid method for movie recommendations that made use of Collaborative Filtering models and graph features. It used an image-user-item model. To learn the interactions between the users and items, this model used single layer neural networks and matrix factorization. It also considered various graph architecture types for the identification of the graph structure, which conveyed the relationship between users and items. The system was tested on two benchmark datasets, Movielens 1M and Movielens 100k, and observed RMSE as 0.923 and MAE as 0.644 respectively.

This method combined collaborative filtering with graph-based features, incorporating single-layer neural networks and matrix factorization to capture user-item interactions and poster image characteristics. Tested on Movielens datasets, the hybrid model achieved solid performance with RMSE of 0.923 and MAE of 0.644, but its reliance on graph architecture and neural networks increased the system's complexity and resource requirements.

The hybrid system implemented by A Tanwar et al. [7] was a combination of a user network and a deep neural network implemented on the Movielens dataset. This system used the user-item vectors from the high dimensional and non-linear data. The system utilized the similarity among users belonging to the same user network, assuming that the users would share similar tastes. In order to achieve this, the system considered the shortest distance between the two users rather than considering the distance between movie features.The experiments were evaluated over metrics like RMSE and MAE and concluded remarkable values of 0.767 and 0.612 respectively. This approach efficiently captured the user preferences in high-dimensional non-linear data.

However, its reliance on user networks limited recommendations for users without strong connections, and the use of deep neural networks increased the computational complexity. The system presented by P. Bahrani et al. [8] was a hybrid combination of content and collaborative filtering implemented on the Movielens dataset. It presented an improved algorithm for KNN clustering, wordnet and ontology based schemes for recommendation. The system was a blend of memory-based and model-based recommender systems. It used an original ontological analogy. It used the previously mentioned ontological similarity to create a profile for each user and defined a partition for users using it. To generate the final predictions, it also made use of a unique type of KNN.

The system first created a profile for each user and item using ontology. Then, a similarity matrix between all users was made using an improved gray distance-based similarity measure. Users were then clustered by feeding a hierarchical clustering algorithm with the similarity matrix. Then, a temporary set and the target user's profile were updated with the indices of some of the clusters of users that the target user had as the most common. The ontological similarity between the profiles of all the users and the item's profile was then calculated for any item that the target user had not yet rated. This hybrid system was evaluated on various other datasets using evaluation metrics like RMSE and MAE.

The hybrid structure allowed for better handling of diverse data, but the complexity of ontological modeling and clustering relatively increased the computational demands. Z. Wang et al. [9] presented a probabilistic matrix factorization-based recommendation scheme called Visual Recurrent Convolutional Matrix Factorization (VRConvMF), which made use of textual and multi-level visual features extracted from the posters and descriptive texts. It suggested VRConvMF into practice and had extensive tests on three widely used real-world datasets to confirm its efficiency.

The experimental results show that the proposed VRConvMF demonstrates a lower RMSE value. The system effectively integrated textual and multi-level visual features from movie posters and descriptions, resulting in enhanced recommendation accuracy. However, the complexity of incorporating visual and textual data increased computational costs and required substantial resources for feature extraction and processing.

To summarize the literature review, there is scope for the development of a technique that can efficiently address the issue of cold-start recommendations. Also, the traditional approach of content-based recommendations can be enhanced to include more contextual data for capturing the trends of user-preferences. In order to explore varied user interests, a multi-modal approach can be implemented to generate diverse and efficient recommendations. Despite advancements in hybrid movie recommendation systems that integrate various techniques to enhance accuracy and address issues like data sparsity and cold-start problems, significant challenges remain. Existing models often exhibit biases towards popular movies, lack effective incorporation of social and behavioral factors, and face high computational complexity. Additionally, they struggle to balance dynamic user preferences and multi-modal data effectively, leading to less diverse and personalized recommendations for users. Therefore, there is a need to develop more adaptive and efficient recommendation models that can enhance user satisfaction while mitigating these limitations. The objective of this research is to develop a hybrid recommendation framework that integrates multi-modal data to consider dynamic user preferences for the generation of diverse recommendations.

3. Methodology

The hybrid framework suggested in this paper is a module that combines and filters the recommendation results provided by the four individual recommendation modules implemented on the notion of popularity, clustering and dimension reduction methods like Collaborative Matrix Factorization and Single Value Decomposition. The recommendation classifier presented here classifies the recommendations provided by these modules on the basis of the estimated movie ratings. The Recommendation Evaluation Function takes these classified recommendations and user feedback as input to make required updations in the re-generation of recommendations. The proposed methodology works on multi-modal data. It uses Movielens 25M (ml-small) and The movie dataset (tmdb) dataset for validation. The ml-small dataset contains approximately 25 million ratings based on 5 5-star ratings provided by more than a lakh users for about 60 thousand movie titles. The other dataset tmdb-5000 dataset, contains features like movie genre, movie-id, movie title and movie popularity. Table 1 shows the dataset characteristics. The distribution of movie genres across the dataset can be seen in Figure 1.

The first step is the computation of the input vector, and to achieve this, the pre-processing step is applied to the dataset. This step uses data-wrangling methods like the removal of irrelevant dataset features and encoding. It also includes the deletion of movie entries with zero ratings and user entries that rated less than five movies. These pre-processing steps are applied for enhancement of the predication accuracy for the proposed model.

Table 1. Dataset characteristics

Number of movies	62423
Number of users	162541
Number of ratings	25000095
Rating Scale	0.5-5.0
Average rating of the dataset	3.53
Minimum movies rated by the user	27
Maximum movies rated by the user	2456
Average count of movies rated by the user	153
Minimum count of users who rated the movie	1
Maximum count of users who rated the movie	3452
Average count of the users who rated the movie	270
Dataset sparsity	97.6%

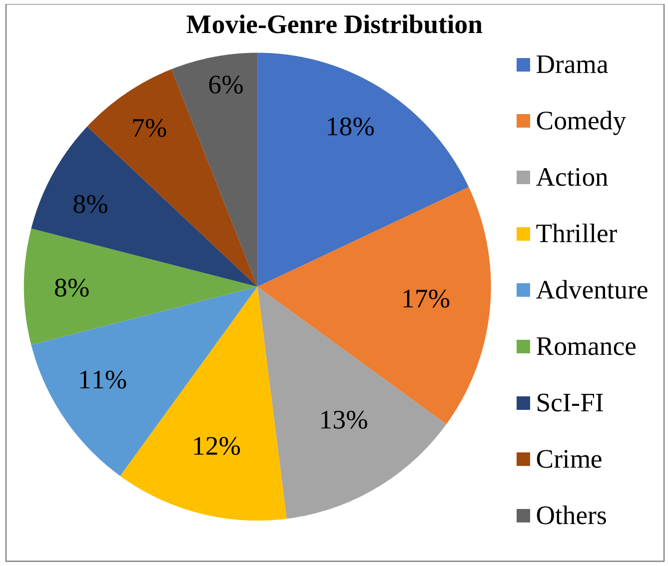

Fig. 1 Movie genre distribution

Figure 2 describes the proposed hybrid framework for recommendation. The input dataset is first computed into a user-movie vector. This vector is then processed by the feature engineering module so that the model can utilize it. To generate the recommendations, various modules use various strategies. The first and simple popularity based module calculates the weighted ratings and suggests the top recommendations. The clustering module works on the movie-rating matrix. It generates clusters of the movies and provides recommendations relevant to the cluster.

The module based on dimension reduction methods like collaborative matrix factorization utilizes the similarity index between two movies to generate recommendations. Another module based on SVD processes a user-movie matrix for generating the recommendations. The recommendations from all these modules are then combined, sorted and re-ordered by the recommendation classifier before presenting them to the user. The user is then asked to provide feedback on the provided recommendations which are subsequently evaluated by the recommendation evaluation function. This function generates a value that is compared to the evaluation threshold value, and depending upon the value, either the generated recommendations are provided to the user or are fed to a hybrid model for the regeneration of recommendations.

3.1. Popularity Module

This module takes as an input the user-movie vector computed by arranging the data according to movies rated by each unique user in the dataset. The popularity module considers data features like the movie votes, total rating for each movie and movie average rating. This unit of the proposed model utilizes the mentioned features of the input vector for the computation of movie popularity. The Popularity algorithm takes U, M, R, and V as input and provides R as output.

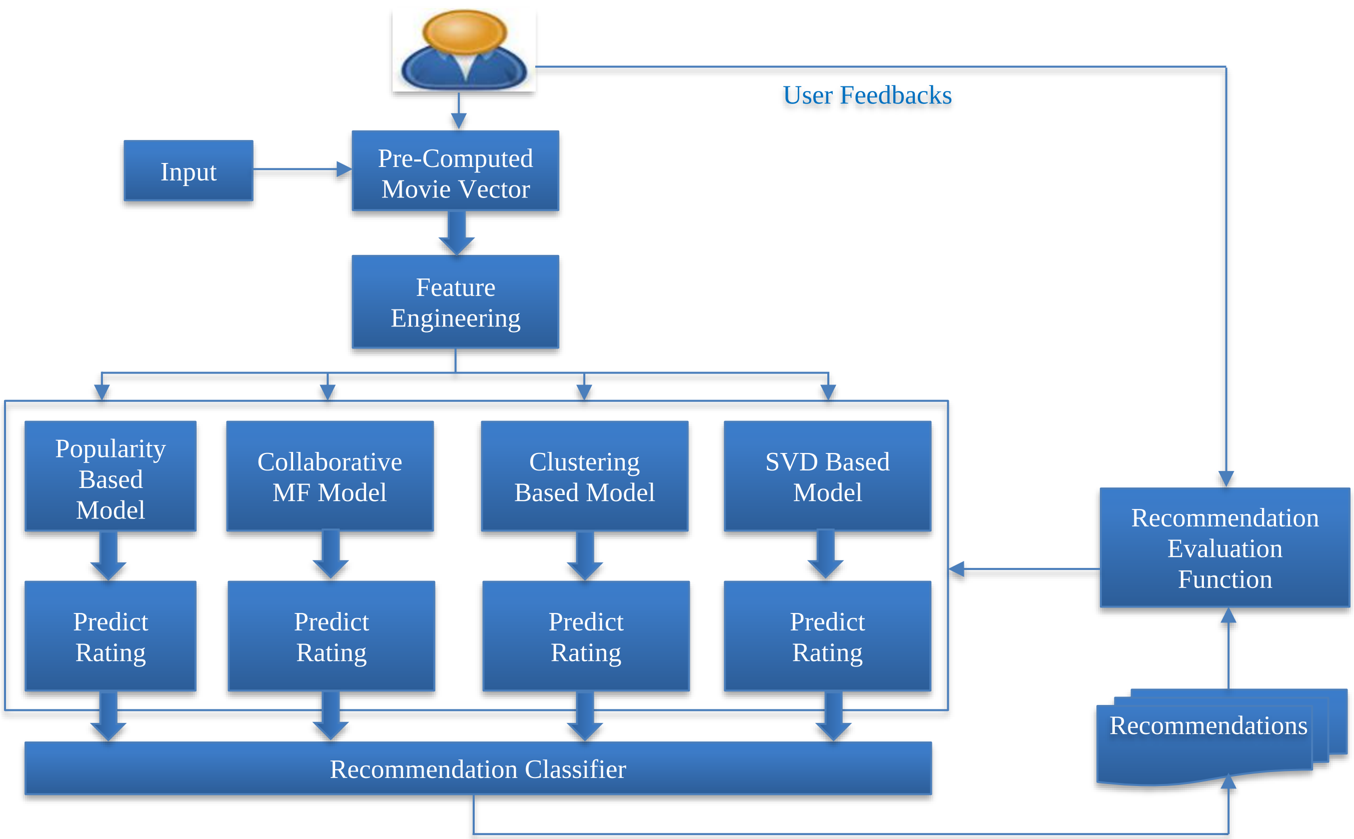

Fig. 2 Proposed framework

Algorithm 1: Popularity Module

Input: - U-set of Users, M-set of movies, R=Set of Rating, where R_{im} denote user i rated movie m, V =Votes, where V_{im} denote user i voted for movie m

Output: - S= Set of Scores for all movies, Recommendation=Set of Recommendations

Step 1: For all movies m in M,

a) Find a set of Users ($Users_m$) who have rated movie m for all i in M:

if $R_{im} > 0$, then add User i to $Users_m$

b) Find total vote

Set tv_m to 0

for all i in $Users_m$

$tv_m = tv_m + V_{im}$

c) Find average vote $av_m = tv_m/size(Users_m)$

d) Find average rating

Set tr_m to 0

for all i in $Users_m$

$tr_m = tr_m + R_{im}$

e) Find average rating $avr_m = tr_m/size(Users_m)$

f) Calculate the popularity score for movie m

$$pop(m) = \frac{(ar(m)*tv(m))+(av(m)*mv(m))}{tv(m)*mv(m)}$$

Store pop (m) to Score (m)

Step 2. Arrange Scores in descending order.

Step3: Add first 10 movies from Scores to Recommendations R

movie_id	original_title	popularity
211672	Minions	876
157336	Interstellar	724
293660	Dead pool	515
118340	Guardians of the	481
76341	Mad Max	434
135397	Jurassic World	419
22	Pirates of the	272
119450	Dawn of the Planet	244
131631	The Hunger Games:	206

Fig. 3 Popular movies recommended by popularity model

userId	movieId	rating	genres
1	1	4.0	Adventure|Animation|Children|Comedy|Fantasy
1	3	4.0	Comedy|Romance
1	6	4.0	Action|Crime|Thriller
1	47	5.0	Mystery|Thriller
1	50	5.0	Crime|Mystery|Thriller
2	4	3.0	Mystery|Thriller|Action
2	45	3.0	Comedy|Romance

Fig. 4 User-movie input vector

The popularity module calculates the popularity of the movies within the genre. Figure 3 represents the popular movie computed and suggested by the popularity module.

3.2. Clustering based Module

The clustering module takes into consideration the vector build using the movie titles and movie ratings. The movie rating considered by this module is the average rating of the movie. It uses the movie-rating vector, as shown in Figure 4, for the formation of the user cluster who rated similarly for the same movie. The performance of several similarity metrics was evaluated and compared, as shown in Figure 5. From Figure 5, it can be concluded that the Pearson-baseline provides optimised results and hence, it is used to find the distance between the target user and similar users. The module partitions the vector into clusters using the mean rating of the movie and adds movies with similar genres and ratings into the same cluster. The top-rated movies from the cluster so formed are then offered as recommendations from this module. The algorithm of this module is shown below.

Algorithm 2 Clustering Module:

Step 1: Given a target user T, find a set of unrated movies (Z)
Step 2: For all the movies in Z, find a set of users (U) who have rated the movie
Step 3: Find the sum of the product of movie rating and distance between the target user and users in U
Step 4: Compute the average rating using the sum computed in Step 3
Step 5: Arrange the average rating in predicted in descending order.
Step 6: Recommend a list of top 10 movies

3.3. Matrix Factorization based Model

The Collaborative Matrix Factorization method factorizes the user-movie interaction matrix into two lower dimension matrices where one represents user features and the other represents movie features. The Singular Value Decomposition (SVD) algorithm is used to achieve this dimension reductionality.

The following formula is used to perform dimension reduction.

$$A = U\Sigma V^T \tag{1}$$

Where,
U - User latent matrix, where each row represents a user, and each column represents a latent factor.
V- Movie latent matrix, where each row represents a movie, and each column represents a latent factor.
Σ - Diagonal matrix containing the singular values, which represent the movie features.

In order to predict ratings, the original matrix A must be reconstructed using the reduced matrices. Recommendations can be made by selecting the highest rated movies. An extended version of SVD, i.e. SVD++ is also used in this module to inherit the implicit input of the user.

3.4. Hybrid Model

All models operate simultaneously, and their results are aggregated to offer an improved recommendation, as illustrated in Figure 1. This process is called Ensemble Learning, which merges multiple models into a unified, robust model. The outcomes of all models are stored in a list and are provided to the Recommendation Classifier. This hybrid model is implemented using the Surprise Library and is optimized using the GridSearchCV algorithm.

The value of hyper-parameters that gives optimized value for performance metrics are number_of_epoch=12; number_latent_factor=56; regularization_parameter=0.03; learning_rate=0.02.

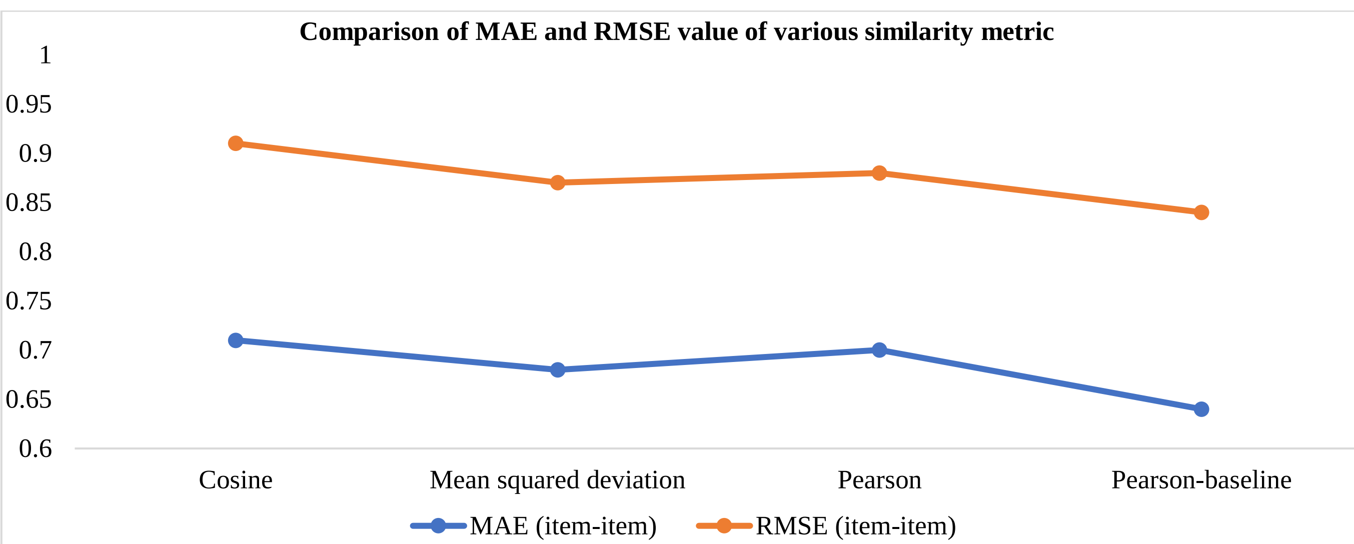

Fig. 5 RMSE and MAE comparison for various similarity metric

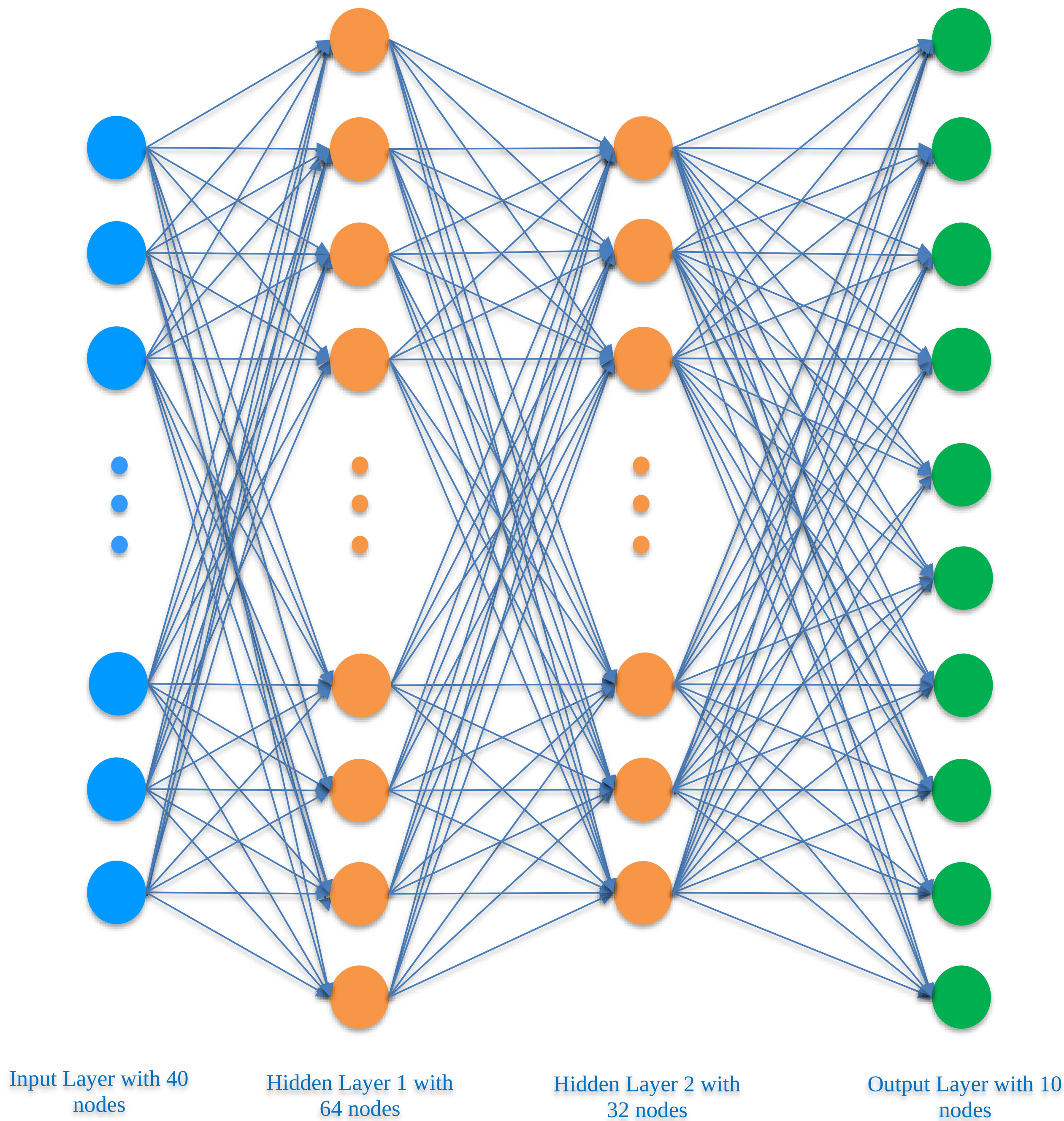

Fig. 6 Structure of artificial neural network

The novel contribution of the proposed system is the ranking and classification of the combination of recommendations provided by the individual modules. The ranking of the recommendation set is done on the basis of the weighted average of the recommendation set and rearranging them in decreasing order of the predicted rating. The classification is achieved by the implementation of the Artificial Neural Network (ANN). This ANN receives the output of the four modules and its objective is to learn the latent factors of the input and to classify them so as to provide a batch of ten movie recommendations to the users. The ANN is implemented using the Keras Library of Python. The general structure of ANN is shown in Figure 6. The input layer takes the user-movie vector as input. The input layer of the Neural Network has 40 input nodes, and each node is represented using 5 features, namely movieId, title, genre, and est_rating. Here, est_rating represented the predicted rating calculated by the model. The hidden layer consists of user and movie embeddings that resemble the latent factors of matrix factorization. There are two hidden layers consisting of 64 and 32 neurons, respectively. The output layer provides 10 output nodes each representing the movie recommendation to be made to the user. The network is trained and tested over 10 epochs. The classified recommendations are then presented to the user, and user feedback is demanded. The user feedback about the relevancy of the presented recommendations is then evaluated, and depending on the value of the relevancy factor, the recommendations are offered to the users. The Recommendation relevancy factor (e) is calculated using the evaluation function shown in the Equation below.

Relevancy Factor (e) =Relevant Rec/Total Rec (2)

If the value of relevancy factor (e) is greater than or equal to 0.7, then recommendations are presented to the user else the hybrid model recomputes them. The relevancy factor determines the ratio of recommendations found relevant by the user from amongst the presented batch of 10 recommendations. The advantage of user feedback over the presented batch of recommendations enhances the degree of personalization and diversity in recommendations.

4. Results and Discussion

The proposed methodology is implemented using Surprise Library and dataset processed through Pandas data-frame. The selection of evaluation parameters depends on the dataset's attributes and the specific task performed by the recommendation algorithm. Several metrics, like Loss Metrics and Decision Support Metrics, are used to assess the performance of the recommendation system.

The ratio of predicted ratings to verified ratings is evaluated using statistical metrics like Mean Average Error (MAE) and Root Mean Square Error (RMSE). Precision is the metric that focuses on the quality of the positive predictions, whereas Recall focuses on the quantity of the positive predictions.In order to decide the number of recommendations to be presented to the user, the proposed system was tested for Precision@k, Recall@k, Accuracy@k and F1-score@k for all the genres in the training set of the dataset. Figures 7-10 show values for decision support metrics for different numbers of recommendations for the training set.

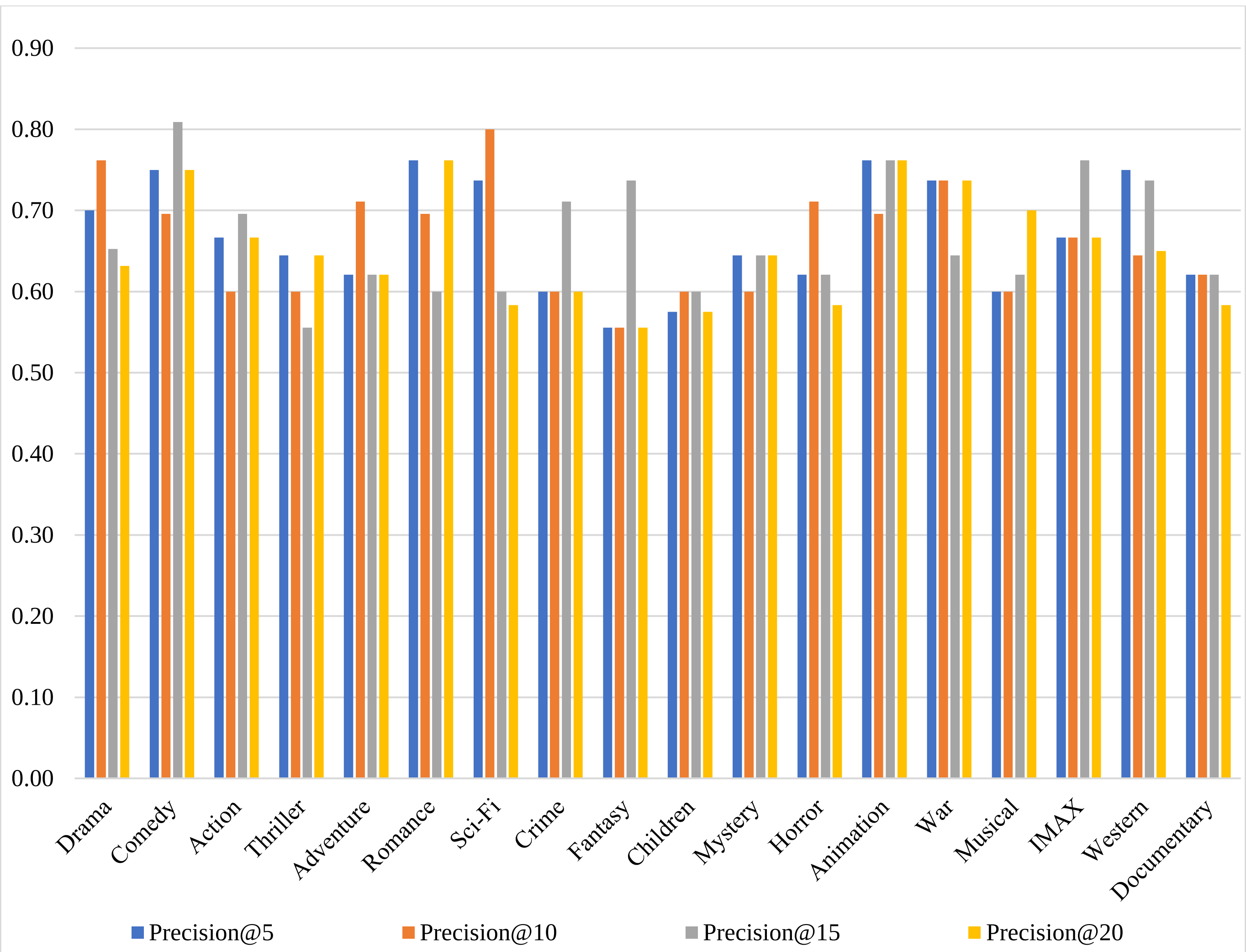

Fig. 7 Precision@k for training set

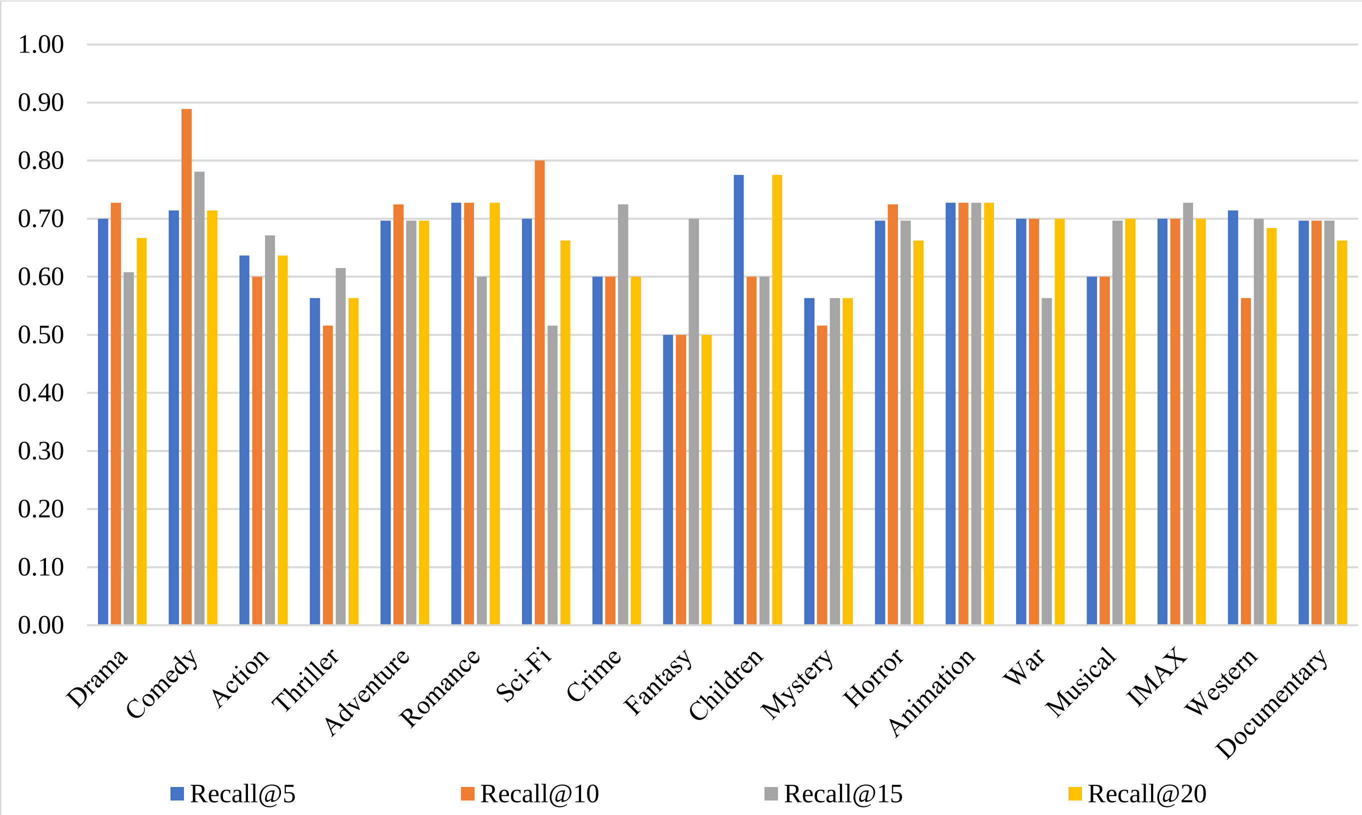

Fig. 8 Recall@k for training set

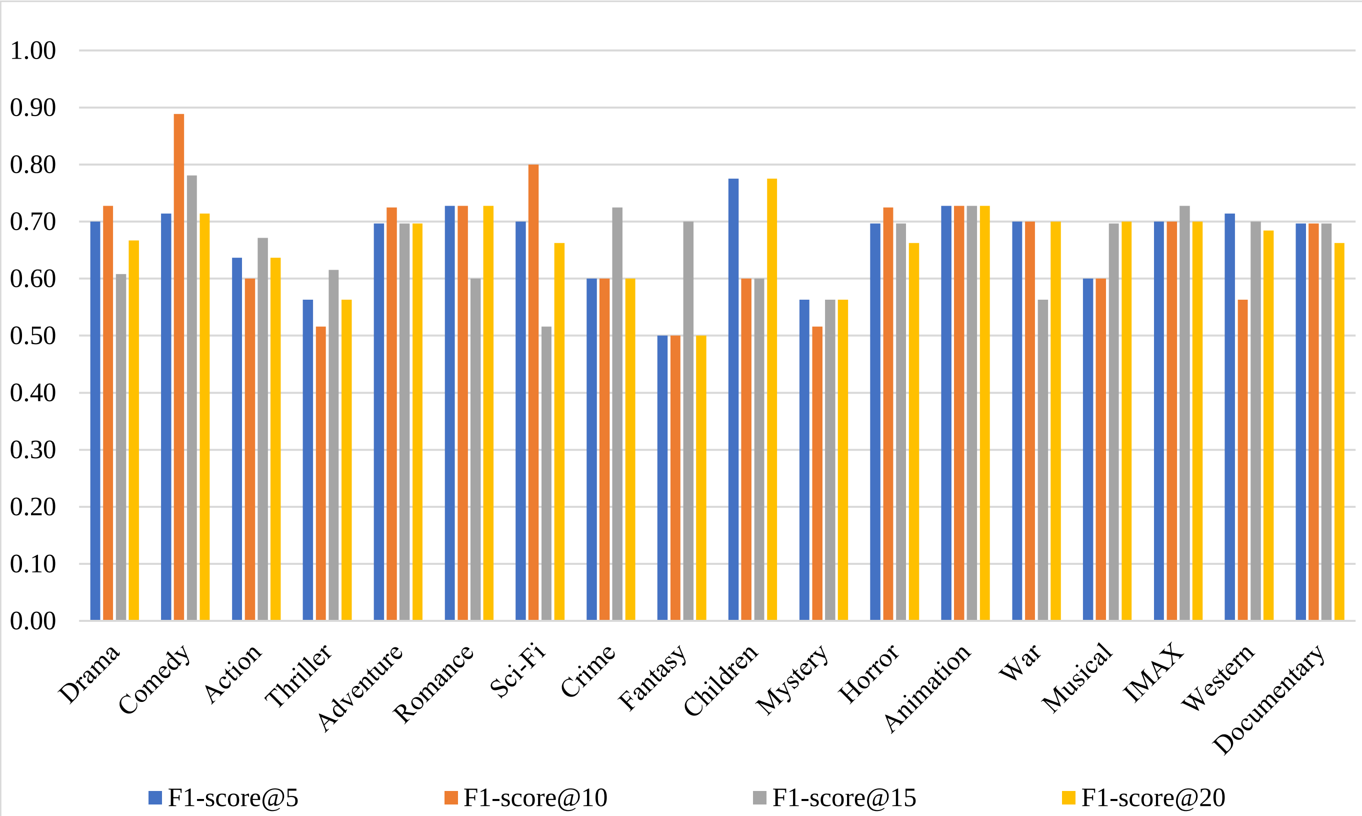

Fig. 9 F1-score@k for training set

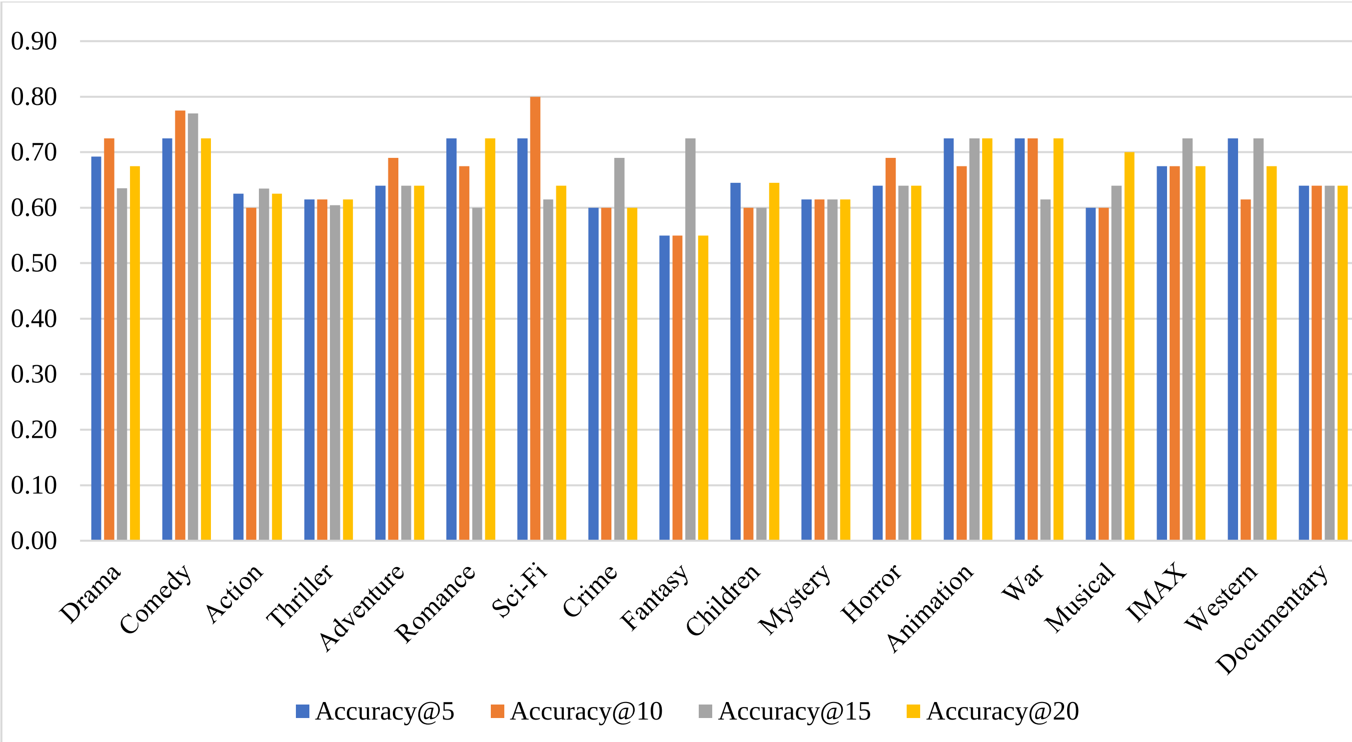

Fig. 10 Accuracy@k for training set

From the above figures, it is observed that the metrics give optimized values for k=10. Therefore, the proposed system provides a batch of 10 recommendations.The performance of the proposed hybrid scheme is analyzed for the common scenario of movie recommendations. When the user-movie interaction has enough data, the user-user correlation, user-movie correlation and movie-movie correlation are utilized for the generation of recommendations. So, in this scenario, the recommendations presented to the user are a combination of recommendations generated by all modules of the hybrid scheme. Figure 11 depicts the user-genre interaction vector, and Figure 12 shows the corresponding recommendations generated for the user.On the other hand, when the user-movie interaction is a sparse matrix, not enough data about user preferences is available.

Adventure	Animation	Children	Comedy	Fantasy
4.5	4.9	4.5	4.3	4.3
Romance	Drama	Action	Crime	Thriller
4.5	4.56	4.23	4.4	4.1
Horror	Mystery	Sci-Fi	War	Musical
3.89	4.6	4.24	4.67	4.65
Documentary	IMax	Western	Film-Noir	Other
0	0	4.3	2	0

Fig. 11 User genre vector for scenario 1

movie Id	title	genre	model	est_ rating
2	Jumanji	Adventure| Children| Fantasy	CMF	4
46972	Night at the Museum	Action| Comedy| Fantasy| IMAX	CMF+KNN	3
40851	Zathura	Action| Adventure| Children| Fantasy	CMF+KNN	3
65685	Inkheart	Adventure| Fantasy	Popularity+KNN	3
144620	Goosebumps	Adventure| Comedy|Horror	CMF+KNN	3
158	Casper	Adventure| Children	CMF	2
74255	Rampage	Thriller	CMF+KNN	2
120488	The Call of the Wild	Adventure| Children	CMF	2
2153	Avengers	Action| Adventure	CMF+KNN	2
480	Jurassic Park	Action| Adventure| Sci-Fi| Thriller	Popularity	2

Fig. 12 Hybrid model recommendations for scenario 1

Adventure	Animation	Children	Comedy	Fantasy
4.6	4.5	4.5	3	3
Romance	Drama	Action	Crime	Thriller
2	3	4.5	3	4.3
Horror	Mystery	Sci-Fi	War	Musical
4.2	4.1	4.3	2.2	1
Documentary	IMax	Western	Film-Noir	Other
0	0	0	0	0

Fig. 13 User genre vector for scenario 2

In this case, the user is considered as the new user, and the scheme uses popularity module and movie-movie correlations for the generation of the recommendation. The user-genre vector and its recommendations are shown in Figures 13 and 14, respectively. Ensemble based hybrid recommendation systems combine multiple recommendation approaches.

The integration of multiple recommendation techniques provides more accurate recommendations as compared to individual methods. Figure 15 shows the values of evaluation metrics for various genres for the testing set. From the figure, it is observed that the average values for Precision, Recall, F1-score and Accuracy for the testing set are 0.7, 0.73, 0.71 and 0.66, respectively.

movie Id	title	genre	model	est_ rating
8361	Day After Tomorrow	Action | Adventure | Drama | Sci-Fi | Thriller	Popularity	4
78499	Toy Story 3	Adventure | Animation | Children | Comedy | Fantasy | IMAX	Popularity	4
7347	Secret Window	Mystery | Thriller	Popularity	4
150536	The Aviators	Animation	Clustering	4
187541	Incredibles 2	Action | Adventure | Animation | Children	Popularity	4
203222	The Lion King	Adventure | Animation | Children | Drama	CMF+SVD	3
4896	Harry Potter and the Sorcerer's Stone	Adventure | Children | Fantasy	Popularity	3
85268	Scooby-Doo! The Mystery Begins	Comedy | Fantasy | Mystery	Clustering	2
89983	Robot	Action | Comedy | Musical | Sci-Fi	Popularity	2
8810	AVP: Alien vs. Predator	Action | Horror | Sci-Fi | Thriller	CMF+SVD	2

Fig. 14 Hybrid model recommendations for scenario 2

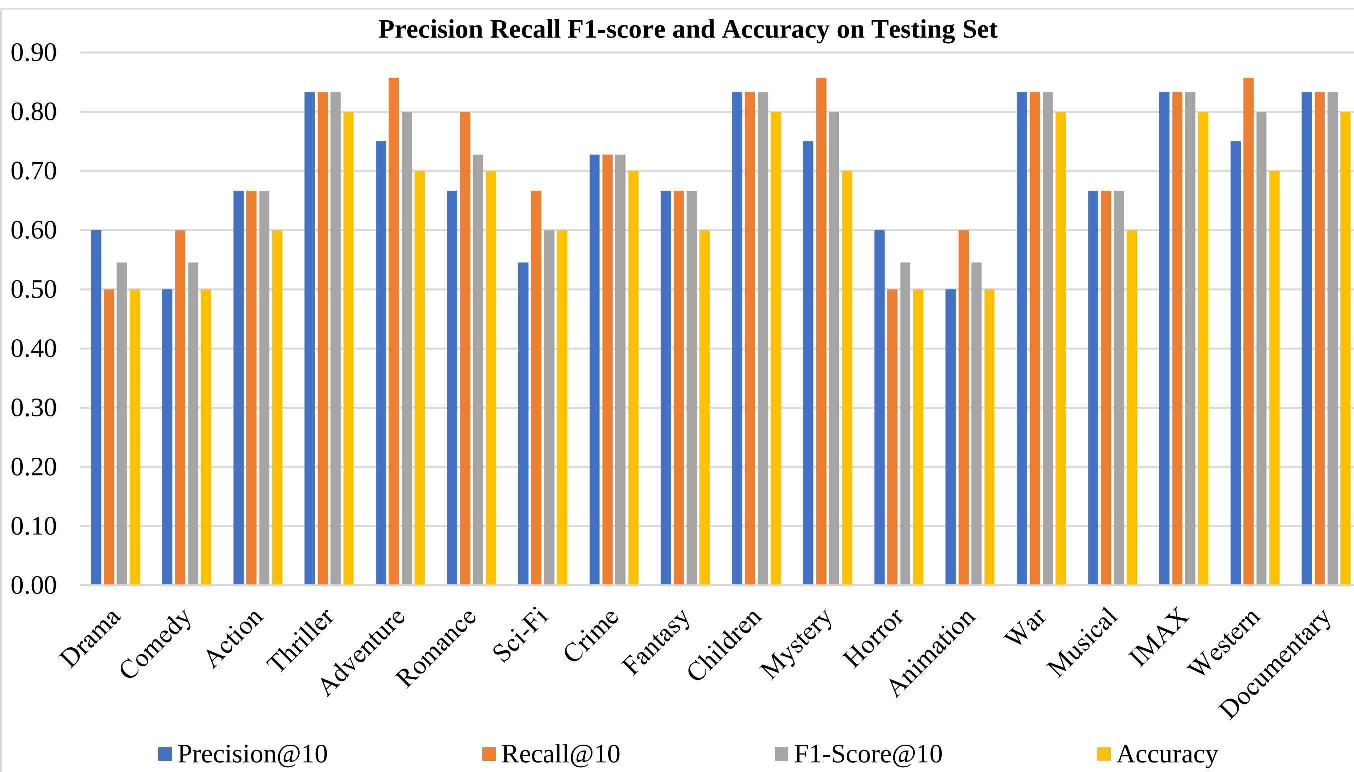

Fig. 15 Evaluation metrics on testing set

Table 2. Comparison of the proposed hybrid algorithm to the baseline model

Baseline Model	RMSE	MAE
Popularity algorithm	0.8	0.78
Clustering algorithm	0.78	0.67
SVD	0.64	0.75
Proposed Hybrid Algorithm	0.56	0.43

Table 3. Comparison of the proposed hybrid scheme to the existing model

Algorithm	RMSE	MAE
Deep Recommendation System [2]	0.838	0.641
DNN based System [6]	0.767	0.612
iMovieRec [8]	0.923	0.644
Proposed Hybrid Scheme	0.561	0.432

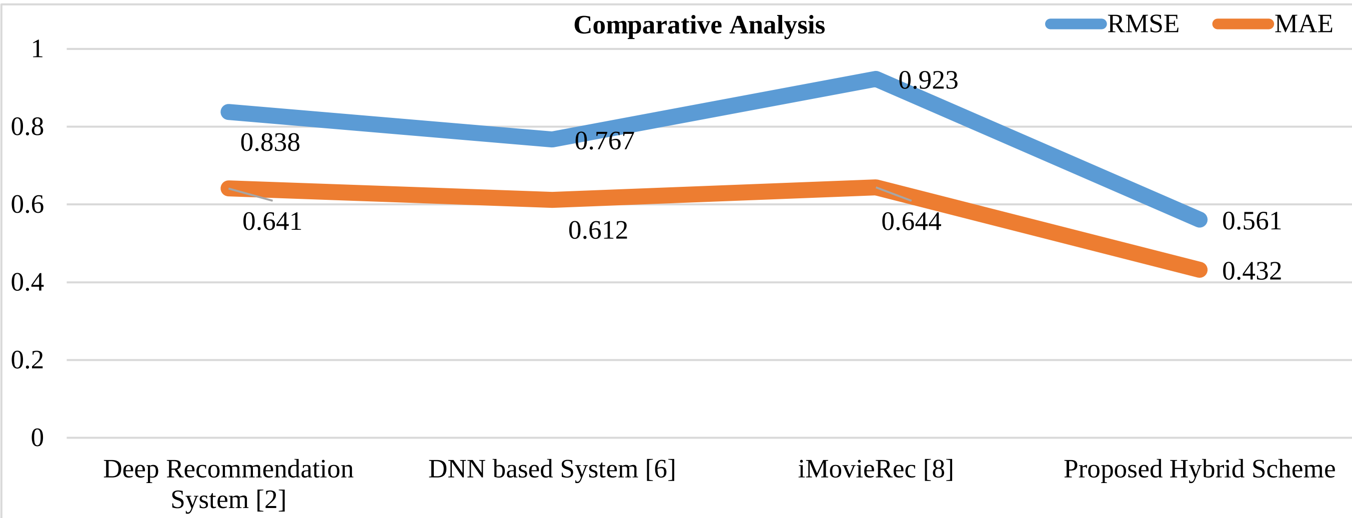

Fig. 16 Comparison of hybrid model

The performance of the hybrid model is compared to the baseline models on the basis of loss functions like RMSE and MAE. Table 2 gives a comparison of the proposed hybrid model to the individual algorithms. It can be seen that the hybrid model gives better performance as compared to individual algorithms. The proposed scheme is also compared to the existing hybrid recommendation system from the rich literature. Table 3 provides a comparative analysis of the performance of proposed methods to the algorithms studied in the literature. The Neural Network based classifier effectively addresses the cold start problem, where new users have not interacted with the movies in the dataset. The use of a classifier enhances the quality and diversity of the recommendations.

The experimental results demonstrated substantial improvement in the value of performance metrics. Figure 16 shows the comparison of performance metrics of the proposed hybrid scheme to various existing algorithms.This improvement suggests that the integration of the classifier into the hybrid model provides improved prediction accuracy, thereby enhancing user satisfaction. Hybrid recommendation systems effectively overcome the limitations of traditional single-method approaches by integrating various techniques such as popularity-based methods, clustering, dimensionality reduction, and neural networks. This combination enhances the system's ability to provide personalized, scalable, and diverse recommendations while addressing challenges like cold-start, concept drift, and complex user behavior. The use of neural networks further strengthens the system's adaptability and performance, making it well-suited for modern recommendation tasks.

5. Conclusion and Future Scope

The paper presents a novel ensemble-based hybrid recommendation system that combines the standard approaches of movie recommendation. This integration of diverse recommendation techniques leads to improved recommendation quality, broader coverage, and increased adaptability to varying user preferences. The use of Neural Networks as classifiers has an advantage in that the system adapts to learn the complex user-movie interaction an provides diverse recommendations. Furthermore, the proposed system efficiently addresses the problems like cold start and concept drift. The experimental evaluation demonstrates the effectiveness of the proposed approach with a substantial improvement of 3.5% and 2.3% in the values of RMSE and MAE. By utilizing the unique strengths of each model, the ensemble system effectively mitigates the limitations of individual approaches, resulting in more reliable and insightful recommendations. The future scope of this work can be an incorporation of varied information like user demographics and contextual details like location, device and user mood. It can also work on various fusion techniques like stacking or cascading of individual models. Unlike Neural Network, Neural Collaborative Filtering or Graph Neural Network can offer improved recommendations. By pursuing the suggested avenues for future work, the ensemble-based hybrid recommendation system can be further refined and extended to deliver more accurate, personalized, and context-aware recommendations for users across diverse application domains.

Acknowledgments

The authors are thankful to the G H Raisoni University Amravati for providing research support for completing this work smoothly.

References

[1] R.N. Ravi Kumar, Sanjay Jain, and Manash Sarkar, “Efficient Hybrid Movie Recommendation System Framework Based on a Sequential Model,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 9, pp. 145-155, 2023. [Google Scholar] [Publisher Link]

[2] Chia-Chi Wu, Yen-Liang Chen, and Yi-Hsin Yeh, “A Deep Recommendation Model Considering the Impact of Time and Individual Diversity,” *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 2558-2569, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Hanrui Wu et al., “Cold-Start Next-Item Recommendation by User-Item Matching and Auto-Encoders,” *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2477-2489, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Luong Vuong Nguyen et al., “Extending Collaborative Filtering Recommendation Using Word Embedding: A Hybrid Approach,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 16, pp. 6232-6243, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Syjung Hwang, and Eunil Park, “Movie Recommendation Systems Using Actor-Based Matrix Computations in South Korea,” *IEEE Transactions on Computational Social Systems*, vol. 9, no. 5, pp. 1387-1393, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Syjung Hwang, Hyeongjin Ahn, and Eunil Park, “iMovieRec: A Hybrid Movie Recommendation Method Based on a User-Image-Item Model,” *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 9, pp. 3205-3216, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ayush Tanwar, and Dinesh Kumar Vishwakarma, “A Deep Neural Network-Based Hybrid Recommender System with User-User Networks,” *Multimedia Tools and Application*, vol. 82, no. 10, pp. 15613-15633, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Payam Bahrani et al., “A Hybrid Semantic Recommender System Enriched with an Imputation Method,” *Multimedia Tools and Applications*, vol. 83, pp. 15985-16018, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Zhu Wang et al., “VRConvMF: Visual Recurrent Convolutional Matrix Factorization for Movie Recommendation,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 3, pp. 519-529, 2022. [CrossRef] [Google Scholar] [Publisher Link]