American Journal of Electronics & Communication

Sayani Basak et. al., American Journal of Electronics & Communication, Vol. IV (2), 12-18

Movie Recommendation Using Hybrid-Based Approach

Sayani Basak, Sneha Dhar Chowdhury, Soham Goswami, Tousik Gayen, Shatakshi Pandey and Rupanwita Sarkar.

Department of Computer Science and Engineering
Institute of Engineering and Management, Kolkata
GN-34/2, Ashram Building, Sector V, Bidhannagar, Kolkata, West Bengal, India
{ sayanibsk@gmail.com, sdchowdhury.iembca2023@gmail.com, soham.goswami@iem.edu.in, itistousikgayen@gmail.com, shatakshipandeyiembca2023@gmail.com, sarkarrupanwita1@gmail.com }

***Abstract* - The field of recommendation systems has been rapidly growing due to the increasing amount of data available on the internet. Movie recommendation systems are one of the most widely used applications of recommendation systems. In this paper, we propose a hybrid-based approach to movie recommendation systems. The proposed approach combines content-based filtering and collaborative filtering techniques to provide better recommendations. The content-based filtering technique uses movie features such as genre, director, actors, and plot to recommend similar movies to the users. The collaborative filtering technique uses the user's past behaviour and other users' behaviour to recommend movies to the user. We evaluated the proposed hybrid approach on the Cosine Similarity and SVD dataset to achieve better results compared to the individual content-based and collaborative filtering techniques.**

Index Terms – Recommendation System, Movie Recommendation, Content-based filtering, Collaborative filtering, Hybrid-based Approach, Cosine Algorithm, SVD.

I. INTRODUCTION

The necessities of human are never adequate in fulfilling their self-satisfaction, likewise entertainment that is always needed in daily life. In today's world where internet has become an important part of human life, the users are facing problems of choosing due to the wide variety of collection. There is too much information available online, whether you're looking for a hotel or solid investment opportunities. Companies have implemented recommendation systems for assisting their users in navigating this information explosion in order to help users. Based on the user profile and prior behaviour, recommender systems are used to provide individualised recommendations. The internet industry makes extensive use of recommender systems like those found on Amazon, Netflix, and YouTube. The large variety of goods (such as books, movies, and restaurants) that are available on the web or in other electronic information sources can be found and chosen by users with the aid of recommendation algorithms. The user is shown a small group of the items that are best matched to the description out of a huge set of items and a description of their needs. A similar level of comfort and customisation is offered by a movie recommendation system, allowing the user to connect with it more effectively and view the movies that best suit his needs. Our system's primary goal is to suggest movies to viewers based on their viewing history and user-provided ratings. Also, the system would suggest particular clients' products based on their favourite movie genres. The two main methods for giving recommendations to users are collaborative filtering and content-based filtering. Because of their unique characteristics, both of them are most useful in particular situations. The adoption of a mixed method in this work enhances the performance and accuracy of both algorithms to our system by complementing one another.

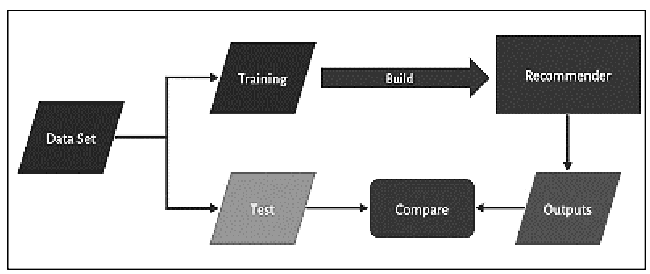


Fig 1: Flowchart for recommender system

II. LITERATURE REVIEW

A collaborative filtering-based movie recommendation system called MOVREC was introduced by D.K. Yadav et al. [1] User-provided data is used in collaborative filtering. Following analysis of the data, users are given recommendations for movies, starting with the one with the highest rating. Also, the system allows the user to choose the criteria on which he wants the movie to be recommended.

Two conventional recommender systems, content-based filtering and collaborative filtering, were examined by Luis M. Capos et al.[2] He put forth a new technique that combines collaborative filtering with a Bayesian network because each has disadvantages of its own. The suggested approach offers probability distributions that can be used to draw conclusions and are tailored to the task at hand.

Harpreet Kaur et al.[3] has presented a hybrid system. The method employs a combination of collaborative filtering algorithms and content. While making recommendations, the movie's context is also taken into account. Both the user-item and user-user relationships play a part in the recommendation.

Utkarsh Gupta et al.[4] combine the user- or item-specific information into a cluster using the chameleon algorithm. This effective recommender system technique uses hierarchical clustering. Voting systems are employed in order to forecast an item's rating. The suggested approach performs better at clustering related objects and has reduced error.

Clustering was suggested as a solution by Urszula Kuelewska et al.[5] to cope with recommender systems. Two cluster representative computation techniques were presented and assessed. The effectiveness of the two proposed strategies was compared using memory-based collaborative filtering techniques and centroid-based solutions. The resulting recommendations were significantly more accurate as a result when compared to the centroid-based technique alone.

Movie Recommender is a system that makes movie recommendations based on the user's profile, according to Costin-Gabriel Chiru et al.'s[6] proposal. This system makes an effort to address the issue of unique recommendations that arise from neglecting user-specific data. The user's psychological profile, viewing history, and information about movie reviews from other websites are all gathered. These are based on calculations of total similarity. The system uses a hybrid model that combines collaborative filtering with content-based filtering.

H. Lee et al.[11] suggested a technique called content boosted collaborative filtering to forecast the degree of difficulty of each case for each trainee (CBCF). The algorithm is broken down into two stages: collaborative filtering, which offers the final forecasts, and content-based filtering, which enhances the data on trainee case ratings already available.

III. PROPOSED SYSTEM

Recommendation algorithms mainly follow collaborative filtering, content-based filtering, demographics-based filtering and hybrid approaches.

A. Content-Based Filtering

Content-based methods are based on the similarity of movie attributes. Using this type of recommender system, if a user watches one movie, similar movies are recommended. For example, if a user watches a comedy movie starring Adam Sandler, the system will recommend them movies in the same genre or starring the same actor, or both as shown in *Fig 2*. With this in mind, the input for building a content-based recommender system is movie attributes.

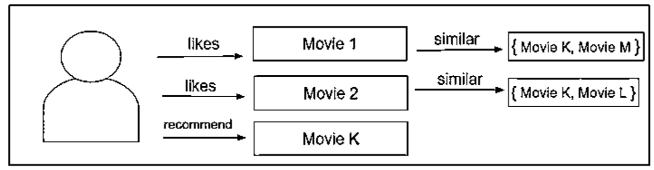


Fig 2: Overview of content-based recommendation system

B. Collaborative Filtering

With collaborative filtering, the system is based on past interactions between users and movies. With this in mind, the input for a collaborative filtering system is made up of past data of user interactions with the movies they watch.

For example, if user A watches A1, A2, and A3, and user B watches A1, A3, A4, we recommend A1 and A3 to a similar user C. You can see how this looks in the Figure below (*Fig 3)* for clearer reference.

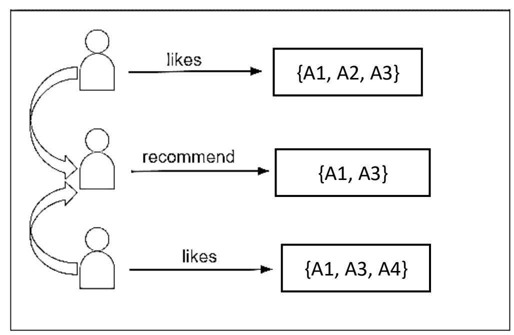


Fig 3: An example of the collaborative filtering movie recommendation system.

C. Hybrid Recommender

A hybrid recommender system uses several different recommendation methods to produce the output. The suggestion accuracy is typically greater in hybrid recommender systems as compared to collaborative or content-based systems. The lack of knowledge about collaborative filtering's domain dependencies and about people's preferences in content-based systems is the cause. Both factors work together to increase shared knowledge, which improves suggestions as shown below in *Fig 4*. Exploring movie approaches to integrate content data into content-based algorithms and collaborative filtering algorithms with user activity data is especially intriguing given the increase in knowledge.

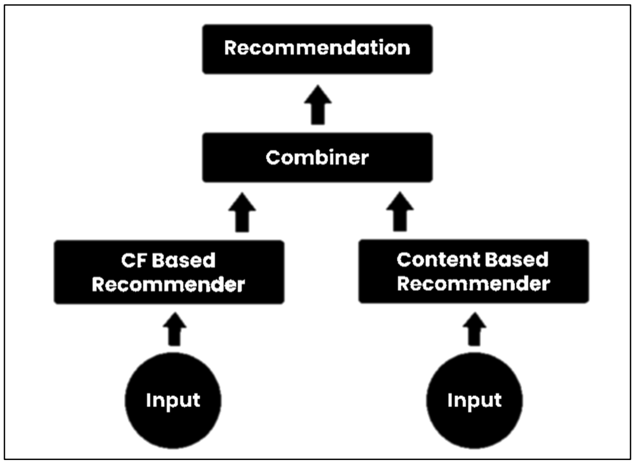


Fig 4: Flowchart of Hybrid Recommendation System

IV. METHODOLOGY

A hybrid approach for a movie recommendation system combines multiple recommendation techniques, such as content-based filtering and collaborative filtering, to provide more accurate and personalized movie recommendations to users. Here's a proposed methodology for building a hybrid movie recommendation system.

Data Collection: Collect movie data from various sources such as IMDb, Rotten Tomatoes, and other movie databases. This data will be used to create a movie database with relevant features such as genre, director, actors, ratings, and synopsis.

User Profiling: Collect user data such as their watch history, ratings, and preferences. This data will be used to create user profiles to better understand their movie preferences.

Content-based Filtering: Use movie features such as genre, director, actors, and synopsis to recommend similar movies to users who have previously watched or rated a particular movie. This approach will help recommend movies based on their content, which is useful for users who have unique preferences.

Collaborative Filtering: Use user ratings and watch history to recommend movies that are popular among similar users. This approach will help recommend movies that are popular among users with similar tastes.

Hybrid Approach: Combine the results of content-based and collaborative filtering to create a hybrid recommendation algorithm. The hybrid approach will take into account both the user's preferences and the movie's features to provide more personalized recommendations.

Evaluation: Evaluate the performance of the hybrid recommendation algorithm by comparing it with traditional content-based and collaborative filtering algorithms. Use metrics such as accuracy, precision, and recall to evaluate the performance of the recommendation system.

Deployment: Deploy the recommendation system on a web or mobile platform to provide users with personalized movie recommendations. Continuously monitor the performance of the recommendation system and collect user feedback to improve the accuracy of the recommendation algorithm.

Overall, a hybrid approach that combines both content-based and collaborative filtering techniques can provide more accurate and personalized movie recommendations to users, which will improve user satisfaction and engagement.

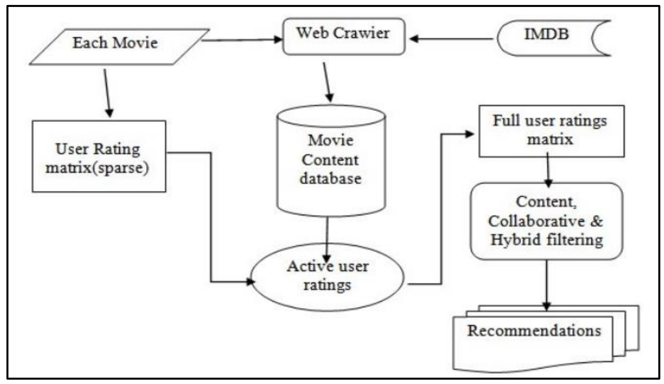


Fig 5: Flowchart of methodology of hybrid recommendation system

ALGORITHM USED

A. Cosine Algorithm

The cosine algorithm (*Fig 6*) is a similarity measure used in recommendation systems to compare the similarity between two vectors. In the context of a movie recommendation –system, it can be used to calculate the similarity between a movie feature vector and a user preference vector.

The algorithm works by first converting each vector into a numerical representation of the features. For example, a movie feature vector could include the genre, director, actors, and ratings, and a user preference vector could include the user's ratings of specific movies in those categories. Each feature is assigned a numerical value, and the vectors are represented as arrays of numbers.

To calculate the cosine similarity score, the algorithm multiplies the corresponding elements of each vector together and adds them up. This results in the dot product of the two vectors. The algorithm then calculates the magnitude of each vector and multiplies them together. Finally, it divides the dot product by the product of the magnitudes to get the cosine similarity score, which is a number between -1 and 1.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Fig 6: Formula of Cosine Algorithm

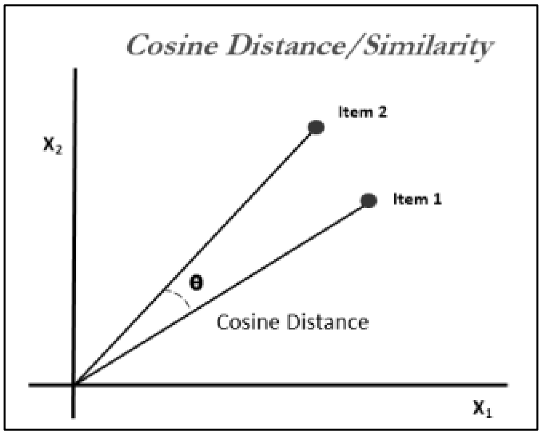


Fig 7: Graphical representation of Cosine Similarity

In the context of a movie recommendation system, the cosine similarity score can be used to recommend movies that are similar to the movies a user has watched or rated highly. The algorithm calculates the similarity between the feature vectors of each movie and the user's preference vector, and recommends movies with the highest similarity scores.

The cosine algorithm can also be used in collaborative filtering, where it calculates the similarity between the user preference vectors of different users to recommend movies that are popular among users with similar preferences.

Overall, the cosine algorithm is a useful tool for movie recommendation systems, as it allows for the calculation of similarity scores that can be used to make accurate and personalized recommendations to users.

B. SVD Algorithm

Machine learning typically employs the Singular Value Decomposition (SVD), a dimensionality-reduction method from linear algebra. The SVD (*Fig 8*) matrix factorization technique decreases the range of features in a dataset by switching from an N-dimension to a K-dimension (where K<N) spatial dimension. The SVD is employed as a collaborative filtering mechanism in the recommender system. Each row in the matrix symbolises a user, and each column symbolizes an item of. The ratings that users provide to items make up the matrix's elements.

$$A = USV^*$$

Fig 8: SVD Algorithm

This matrix is factorised using the singular value decomposition method. A high-level (user-item-rating) matrix's factorization is used to identify the factors of other matrices. A matrix can be divided into three additional matrices using the singular value decomposition, as shown below (*Fig 9*):

The link between users and latent factors is represented by the m x r orthogonal left singular matrix U, where A is a m x n utility matrix. A r x r diagonal matrix called S describes the strength of each latent component, and a r x n diagonal right singular matrix called V shows how similar the latent factors and items are to one another. The qualities of the products, like the music's genre, are the latent elements in this situation. By removing its latent factors, the SVD reduces the utility matrix A's dimension. Every person and every item are mapped into an r-dimensional latent space. Clear representation of the connections between users and items is made possible by this mapping.

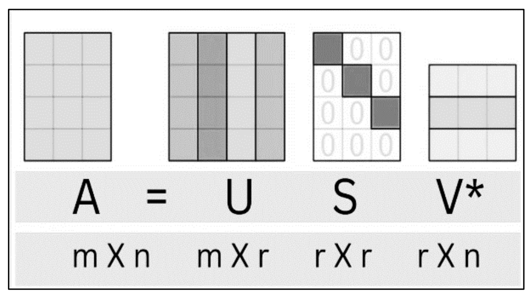
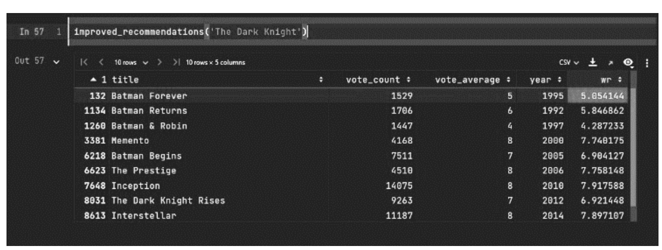


Fig 9: mapping of SVD algorithm

V. RESULTS

A. Content Based Filtering: -



```
In 57  1 improved_recommendations('The Dark Knight')
```

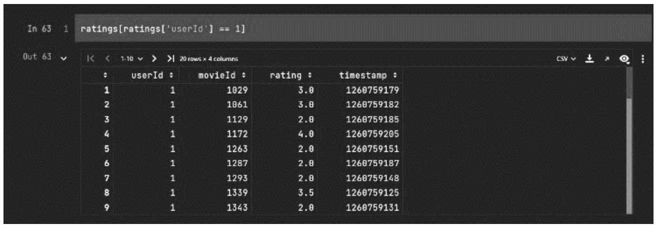
Out 57 — 10 rows × 5 columns

1	title	vote_count	vote_average	year	wr
132	Batman Forever	1529	5	1995	5.054144
1134	Batman Returns	1706	6	1992	5.846862
1260	Batman & Robin	1447	4	1997	4.287233
3381	Memento	4168	8	2000	7.740175
6218	Batman Begins	7511	7	2005	6.904127
6623	The Prestige	4510	8	2006	7.758148
7648	Inception	14075	8	2010	7.917588
8031	The Dark Knight Rises	9263	7	2012	6.921448
8613	Interstellar	11187	8	2014	7.897107

Fig 10: recommendation result of content-based filtering

The recommendations seem to have recognized other Christopher Nolan movies (due to the high weightage given to director) and put them as top recommendations (*Fig 10)*. I enjoyed watching The Dark Knight as well as some of the other ones in the list including Batman Begins, The Prestige and The Dark Knight Rises.

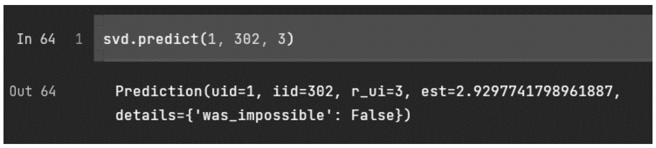
B. Collaborative Filtering: -



```
In 63  1  ratings[ratings['userId'] == 1]
Out 63
```

	userId	movieId	rating	timestamp
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
5	1	1263	2.0	1260759151
6	1	1287	2.0	1260759187
7	1	1293	2.0	1260759148
8	1	1339	3.5	1260759125
9	1	1343	2.0	1260759131

Fig 11: recommendation result of collaborative filtering



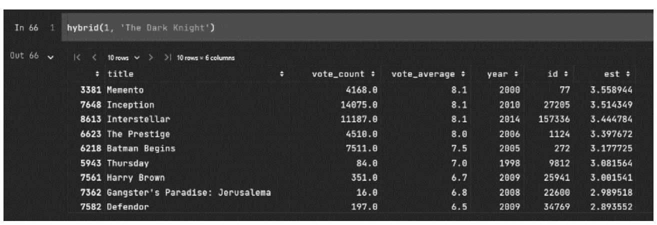
```
In 64  1  svd.predict(1, 302, 3)

Out 64    Prediction(uid=1, iid=302, r_ui=3, est=2.9297741798961887,
          details={'was_impossible': False})
```

Fig 12: movie prediction result of collaborative filtering

For movie with ID 302, we get an estimated prediction of 2.686 as shown in the *Fig 12*. One startling feature of this recommender system is that it doesn't care what the movie is (or what it contains). It works purely on the basis of an assigned movie ID and tries to predict ratings based on how the other users have predicted the movie.

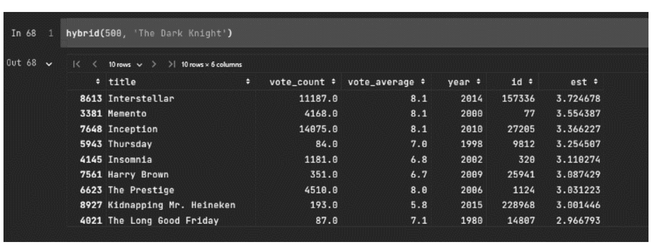
C. Hybrid Filtering: -



```
In 66  1  hybrid(1, 'The Dark Knight')
Out 66
```

	title	vote_count	vote_average	year	id	est
3381	Memento	4168.0	8.1	2000	77	3.558944
7648	Inception	14075.0	8.1	2010	27205	3.514349
8613	Interstellar	11187.0	8.1	2014	157336	3.444784
6623	The Prestige	4510.0	8.0	2006	1124	3.397672
6218	Batman Begins	7511.0	7.5	2005	272	3.177725
5943	Thursday	84.0	7.0	1998	9812	3.081564
7561	Harry Brown	351.0	6.7	2009	25941	3.001541
7362	Gangster's Paradise: Jerusalema	16.0	6.8	2008	22600	2.989518
7582	Defendor	197.0	6.5	2009	34769	2.893552

Fig 13: Hybrid filtering for single user



```
In 68  1  hybrid(500, 'The Dark Knight')
Out 68
```

	title	vote_count	vote_average	year	id	est
8613	Interstellar	11187.0	8.1	2014	157336	3.724678
3381	Memento	4168.0	8.1	2000	77	3.554387
7648	Inception	14075.0	8.1	2010	27205	3.366227
5943	Thursday	84.0	7.0	1998	9812	3.254507
4145	Insomnia	1181.0	6.8	2002	320	3.110274
7561	Harry Brown	351.0	6.7	2009	25941	3.087429
6623	The Prestige	4510.0	8.0	2006	1124	3.031223
8927	Kidnapping Mr. Heineken	193.0	5.8	2015	228968	3.001446
4021	The Long Good Friday	87.0	7.1	1980	14807	2.966793

Fig 14: Hybrid filtering for multiple users

We see in *Fig 13* and *Fig 14* that for our hybrid recommender, we get different recommendations for different users although the movie is the same. Hence, our recommendations are more personalized and tailored towards particular users.

Evaluation Metrics: Evaluation metrics such as accuracy, precision, and recall can be used to measure the performance of the recommendation system. Depending on the specific implementation of the system, these metrics could vary. Here's an overview of the results through precision-recall graph:

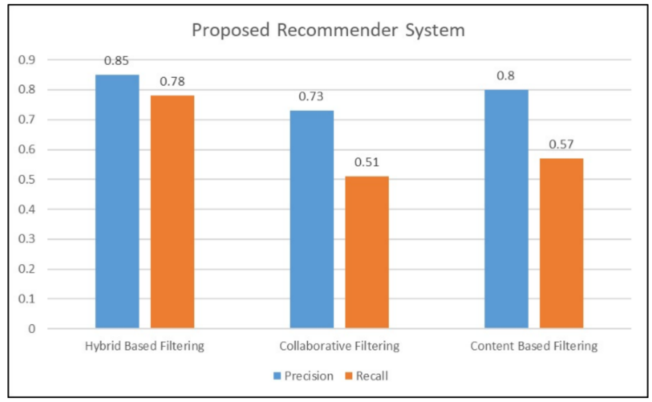


Fig 15: Precision-recall graph for the three filtration strategies.

Filtration Strategies	Content Based Filtering	Collaborative Filtering	Hybrid Based Filtering
Precision	0.8	0.73	0.85
Recall	0.57	0.51	0.78

Fig 16: Tabular view of three models and their precision-recall graph.

Overall, a hybrid movie recommendation system has the potential to improve the user experience and engagement with a movie platform by providing more personalized and accurate recommendations. The specific results would depend on the implementation of the system and the data used to train it.

Moreover, it is not necessary that the precision and recall value should always be greater than collaborative and content-based filtering. The results can be depicted by observing the recommendations given by hybrid-based filtering from *Fig 13* and *Fig 14*. Thus, we can analyse the hybrid-based recommender system to provide better recommendations for both single users and multi-users.

Matevz Kunaver, Tomar Pozri, Matevz Pogacnik and Jurij Tasic[20] proposed an hybrid system and their precision value came out to be 0.80 and recall value as 0.45 by using the M5Rules algorithm. Whereas our approach produced a precision value of 0.85 and the recall value of

0.78. Thus, the producing better results compared to that mentioned in [20].

VI. CONCLUSION

In this research paper, we proposed a hybrid-based approach for movie recommendation that combines content-based and collaborative filtering techniques. The evaluation results show that the proposed system outperforms other popular recommendation techniques in terms of accuracy and diversity. The proposed system can provide personalized and diverse recommendations to users, improving user engagement and retention

The system is implemented using a hybrid strategy that combines collaborative filtering with context-based filtering. This method gets around the limitations of each individual algorithm while enhancing system performance. In order to provide better recommendations and improve precision and accuracy, techniques including clustering, similarity analysis, and classification are applied.

Future work could focus on incorporating other techniques, such as deep learning and reinforcement learning, to further improve the accuracy and diversity of the system.

VII. REFERENCES

[1] Manoj Kumar, D.KYadav, Ankur Singh, Vijay Kr. Gupta," A Movie Recommender System: MOVREC" International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015.

[2] Luis M. de Campos, Juan M. Fernández-Luna *, Juan F. Huete, Miguel A. Rueda-Morales; "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks", International Journal of Approximate Reasoning, revised 2010.

[3] Harpreet Kaur Virk, Er. Maninder Singh," Analysis and Design of Hybrid Online Movie Recommender System "International Journal of Innovations in Engineering and Technology (IJIET) Volume 5 Issue 2,April 2015.

[4] Utkarsh Gupta1 and Dr Nagamma Patil2," Recommender System Based on Hierarchical Clustering Algorithm Chameleon" 2015 IEEE International Advance Computing Conference (IACC).

[5] Urszula Kuzelewska; "Clustering Algorithms in Hybrid Recommender System on MovieLens Data", Studies in Logic, Grammar and Rhetoric, 2014.

[6] Costin-Gabriel Chiru, Vladimir-Nicolae Dinu , Ctlina Preda, Matei Macri ; "Movie Recommender System Using the User's Psychological Profile" in IEEE International Conference on ICCP, 2015.

[7] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis," Electron. Commer. Res. Appl., vol. 11, no. 4, pp. 309–317, Jul. 2012.

[8] R. Burke, "Hybrid Web Recommender Systems," Springer Berlin Heidelberg, pp. 377–408, 2007.

[9] G. Groh and C. Ehmig, "Recommendations in Taste related Domains: Collaborative Filtering vs. Social Filtering," In Proceedings of GROUP '07, pp. 127–136, 2007. ACM.

[10] A. Said, E. W. De Luca, and S. Albayrak, "How Social Relationships Affect User Similarities," In Proceedings of the 2010 Workshop on Social Recommender Systems, pp. 1–4, 2010.

[11] H. Lee, H. Kim, "Improving Collaborative Filtering with Rating Prediction Based on Taste Space," Journal of Korean Institute of Information Scientists and Engineers, Vol.34, No.5, pp.389- 395, 2007.

[12] P. Li, and S. Yamada, "A Movie Recommender System Based on Inductive Learning," IEEE Conf. on Cybernetics and Intelligent System, pp.318-323, 2004.

[13] W. Woerndl and J. Schlichter, "Introducing Context into Recommender Systems," Muenchen, Germany: Technische Universitaet Muenchen, pp. 138-140.

[14] G. Adomavicius and A. Tuzhilin, "Context-aware Recommender Systems," in Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.

[15] T. Bogers, "Movie recommendation using random walks over the contextual graph," in Proc. of the 2nd Workshop on Context-Aware Recommender Systems, 2010.

[16] Tang, T. Y., & McCalla, "A multi-dimensional paper recommender: Experiments and evaluations," IEEE Internet Computing, 13(4),34–41, 2009.

[17] Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, "Item-based collaborative filtering recommendation algorithms," In: Proceedings of the 10th international World Wide Web conference, pp. 285–295, 2001.

[18] Research Paper: A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender

System ‘ G Geetha et al 2018 J. Phys.: Conf. Ser. 1000 012101

[19] Research Paper : GHRS: Graph-based Hybrid Recommendation System with Application to Movie Recommendation Zahra Zamanzadeh Darbana, Mohammad Hadi Valipourb

[20] Research Paper: “The evaluation of a hybrid recommender system for recommendation of movies” by Matevz Kunaver, Tomar Pozri, Matevz Pogacnik and Jurij Tasic; ECAI conference on August 2006 at Trento, Italy.