Research Article

A Deterministic Model for Determining Degree of Friendship Based on Mutual Likings and Recommendations on OTT Platforms

Aqeel Khalique [iD],[1] **Mohammad Khalid Imam Rahmani** [iD],[2] **Mohd Saquib** [iD],[1]
Imran Hussain [iD],[1] **Abdul Wahab Muzaffar** [iD],[2] **Mohd. Abdul Ahad** [iD],[1] **Md Tabrez Nafis** [iD],[1]
and Mohd Wazih Ahmad [iD][3]

[1]*Department of Computer Science & Engineering, Jamia Hamdard, New Delhi, India*
[2]*College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia*
[3]*ASTU, Adama, Ethiopia*

Correspondence should be addressed to Mohammad Khalid Imam Rahmani; m.rahmani@seu.edu.sa and
Mohd Wazih Ahmad; wazih.ahmad@astu.edu.et

In recent years, the application of various recommendation algorithms on over-the-top (OTT) platforms such as Amazon Prime and Netflix has been explored, but the existing recommendation systems are less effective because either they fail to take an advantage of exploiting the inherent user relationship or they are not capable of precisely defining the user relationship. On such platforms, users generally express their preferences for movies and TV shows and also give ratings to them. For a recommendation system to be effective, it is important to establish an accurate and precise relationship between the users. Hence, there is a scope of research for effective recommendation systems that can define a relationship between users and then use the relationship to enhance the user experiences. In this research article, we have presented a hybrid recommendation system that determines the degree of friendship among the viewers based on mutual liking and recommendations on OTT platforms. The proposed enhanced model is an effective recommendation model for determining the degree of friendship among viewers with improved user experience.

1. Introduction

During the last couple of years, Amazon Prime, Netflix, Disney + Hotstar, and several other OTT platforms have emerged. Viewers have changed their watching patterns. The majority of the viewers rely on recommendations or ratings of movies. These watching patterns have been widely exploited by OTT platforms to gain a significant rise in popularity among their customer base. Viewers are privileged to use recommendations that are made available in the recommendation system adapted by the OTT platforms. The trend of recommender systems has increased due to an avalanche of various Internet services such as e-commerce to suggest buyers some interesting commodities that they

might find useful to buy and online promotion to suggest the users some content that might be matching their preferences. Nowadays, recommender systems are almost unavoidable during our online activities. Naturally, users feel more confident to find recommendations accurate and reliable when they come from a known or likable person. Recommender systems have the flexibility to forecast whether users will prefer items based on their personal preferences and choices. A recommendation system is like a data shortlisting process to prioritize the items based on users' preferences discovered from their use patterns. A recommendation system is an algorithm that aims to recommend related items to users, where the items can be movies to watch, texts to read, products to buy, or any other

content that depends on the application field. The recommendation systems have much potential to increase revenue for business organizations dealing in almost every area of application. So, their important features need to be harnessed effectively [1]. The organization of the study is depicted as shown in Figure 1.

There exist numerous recommender systems covering widespread usage and application areas. Figure 2 shows different application areas of recommendation systems.

1.1. Evolution and Significance of Recommendation System. The humongous growth in the accessibility of digital content and an exponential increase in Internet usage have brought a massive data overload due to which timely access to accurate information is delayed. Information retrieval giants such as Google and Bing have nearly resolved this problem, but still lack prioritization and personalization of data (the system maps available content to users' interests and preferences).

This increases the demand for recommender systems over the years [2]. Recommender systems have evolved from item relationship-based to machine learning approaches, and hence, their prediction has been significantly improved. Figure 3 shows the evolutionary techniques of recommender systems as used over the past few years.

Recommendation systems' personalization features improve user involvement and retention. These systems capture users' experience and enhance their recommendations whether it can be product recommendations on an e-commerce site, movie, or music recommendations on streaming services such as OTT platforms [3]. The major reason behind the popularity and growing state of recommendation systems is that the users have multiple options available to choose from. Due to the exponential growth of social networking services, people tend to utilize these available services online. Similarly, on OTT platforms, users have plenty of choices, and hence, it is difficult for any user to decide what recommendation to follow [4–6].

1.2. Existing Approaches to Recommendation Systems. Recommendation systems need to know the personal preference of the user for effective suggestions. Information on user interactions is used to create a recommendation system. Therefore, data collection from different activities namely direct interactions and indirect interactions is very important. Direct interactions include information about the user's previous activities, ratings, comments, and information about the user's profile, such as gender, age, or interests. Indirect interactions include the user's device, clicked links, location, and date [7]. The working principle of the recommendation system is to find patterns in the user's previous practice. User characteristics, such as demographic data such as age and gender, and psychological characteristics, such as interests and dislikes, can help identify customers. Product features such as movie type and actors can help calculate the similarity between products [8]. There are three basic methods used to develop recommendation systems namely content-based, collaborative filtering, and hybrid methods. For example, in Netflix, instead of browsing

through thousands of packages and movie titles, Netflix provides the user with a specific item as per his likeness. This feature can save time and provide a better user experience further reducing the cancellation rate and hence saving the company about a billion dollar annually [9].

1.2.1. Collaborative Filtering. It is based on gathering and inspecting data on users' actions, habits, or likeness and forecasting based on their similarity with other users. Collaborative filtering does not depend on the content of the data, and it recommends without understanding the content of the item. As shown in Figure 4, collaborative filtering is based on historical likings of items resulting in similar types of items in the future.

1.2.2. Content-Based Filtering. It is based on the details of the item and the interaction and preferences of individual users. Recommendations are based on the user's history and interactions. It uses algorithms to recommend items that are similar to the user's earlier liking.

Accuracy depends on the available information about the data. In some industries due to privacy and regulatory issues, users' data and transactional data are not available sometimes resulting in a cold start problem. As shown in Figure 5, in content-based filtering, if a user likes an item, he will also like a "similar" item.

1.2.3. Hybrid Recommendation Systems. The hybrid recommendation systems are an amalgamation of a collaborative filtering approach and a content-based filtering approach. In Netflix, hybrid recommender systems are used by comparing the viewing and search habits of similar users and providing movies that share characteristics with highly rated similar movies. Cold start and data scarcity problems are resolved in the hybrid technique. Figure 6 shows the model of hybrid recommendation systems.

1.3. Phases of Recommendation Systems. Recommendation systems can be implemented in several phases. These phases are shown in Figure 7.

2. Related Works

This section presents current state-of-the-art technology of recommendation systems with social connections between users. Trust-aware recommendation systems are more popular because social trust provides user preferences along with ratings of items. With the advent of social networking sites, personal information in social networks is used to predict user behavior. These methods assume that there are social networks between users and recommend users based on ratings provided by users who are connected with other users. Several research papers were studied and analyzed, and they have been summarized as shown in Table 1.

The contribution of these techniques to recommendation systems is shown in Figure 8. However, as we can observe, the majority of the recommendation systems were
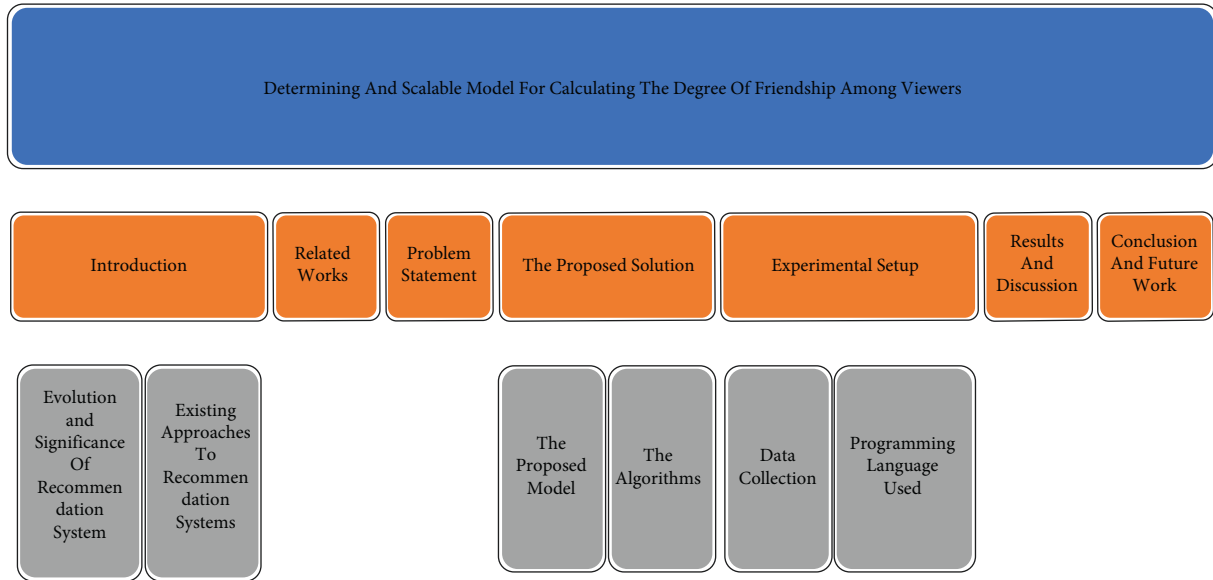
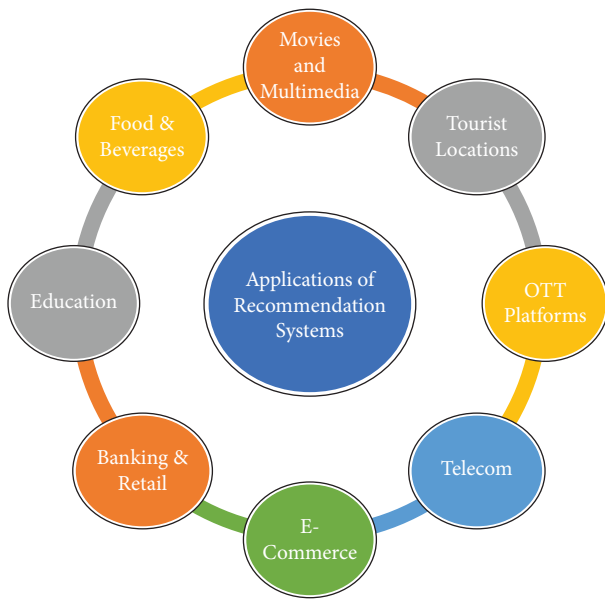Figure 1: Overall structure of the study.

Figure 2: Applications of recommendation systems.

based on collaborative filtering. There is a need of the hour to determine recommendations that shall be scalable and more resilient in terms of social interactions, social bonding, or likings. These social attributes are much required when recommendations are required in OTT platforms.

3. Problem Statement

Nowadays, recommendations are mostly used in every aspect. Recommending users with the most suitable product is important to enhance user satisfaction. Generally, on OTT platforms, users do not know the importance of mutual liking and its impact on their degree of friendship. So, it is very important to establish an effective relationship between

users of recommendation systems to exploit the advantages of their mutual likings for better user experience and satisfaction. For instance, if a user "a" watches a movie on an OTT platform and assumes that three of his other friends also watch movies on the same OTT platform, then it is important to determine the similarity of recommendations among these three users to the user "a"; besides, it will be helpful to determine the mutual likings among the three users. The similarity of likings is not available on OTT platforms for users. However, based on this mutual liking, a set of recommendations can be improved because, generally, people accept the recommendations of persons who have more similar likings in common. Hence, a deterministic and scalable model is much needed to offer more precise and correct recommendations to the users. Furthermore, based on the similarities of mutual likings, there will be more precise and correct recommendations. Therefore, there is a need to determine the degree of mutual friendship among the users. This study proposes a novel and unique model to address these problems.

4. The Proposed Solution

4.1. The Proposed Model. As discussed in the problem statement, we have proposed a deterministic model for computing the degree of friendship among viewers based on mutual liking and recommendations. Our work is an extension of [25]. In this research work, we have implemented an elaborated model for OTT platforms. However, the system can be implemented for every type of recommendation system. We have focused on the social relationships among connected users over any social networking services and further determine the mutual liking and degree of friendship among these socially connected users. We have formulated the objectives below and aimed our system to gather results based on these objectives during the implementation of the research work.
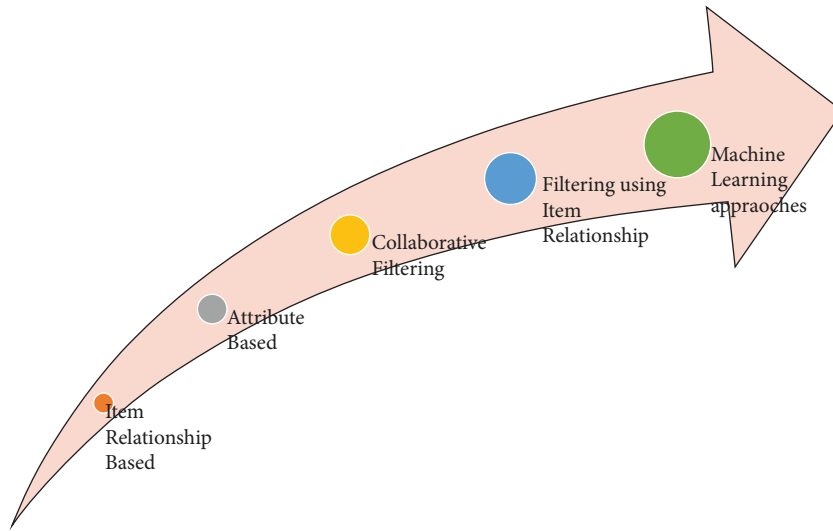
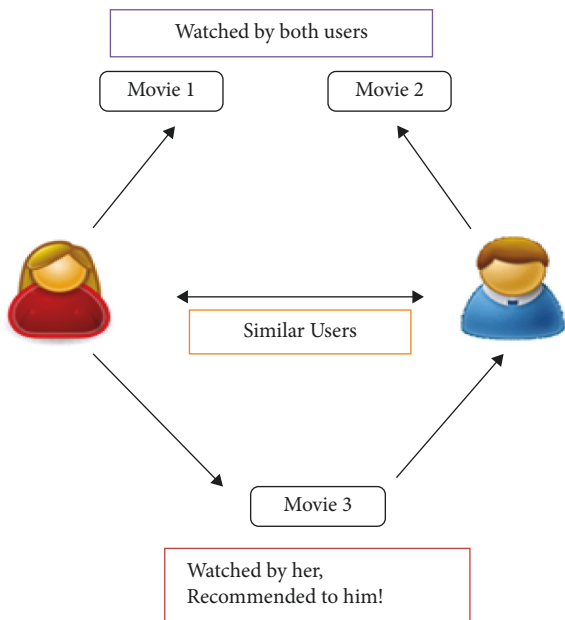FIGURE 3: Evolution of recommendation systems based on different techniques.

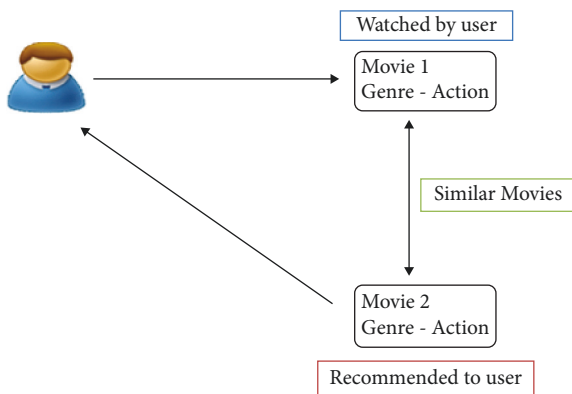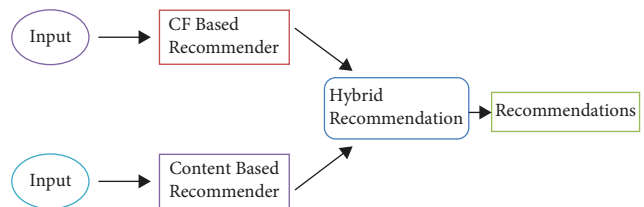FIGURE 4: Collaborative filtering.

FIGURE 5: Content-based filtering.

FIGURE 6: Hybrid recommendation systems.

(i) Objective 1 [O_1]: similarity identification between user "a" and user "b" from the movie ratings given by them

(ii) Objective 2 [O_2]: recommendation of movies to user "a" from other users socially connected to the user "a"

(iii) Objective 3 [O_3]: determining mutual liking between two users

(iv) Objective 4 [O_4]: determining the degree of friendship between two users

Based on our objectives, we have presented our work in [25] and further extended the work to achieve objectives 3 and 4.

The proposed model is represented diagrammatically in Figure 9.

Based on the objectives defined in an earlier section, we have designed the algorithms to achieve these objectives. Taking an exemplary scenario, the proposed model is explained below in detail.

(i) Firstly, we have user details in our system that have watched some movies and give them ratings after watching, and the system will find the common movies that they all have watched before.

(ii) Secondly, the system will determine the Euclidean distance using the ratings of movies that the users gave to commonly watched movies. Using the Euclidean distance, the system will calculate the
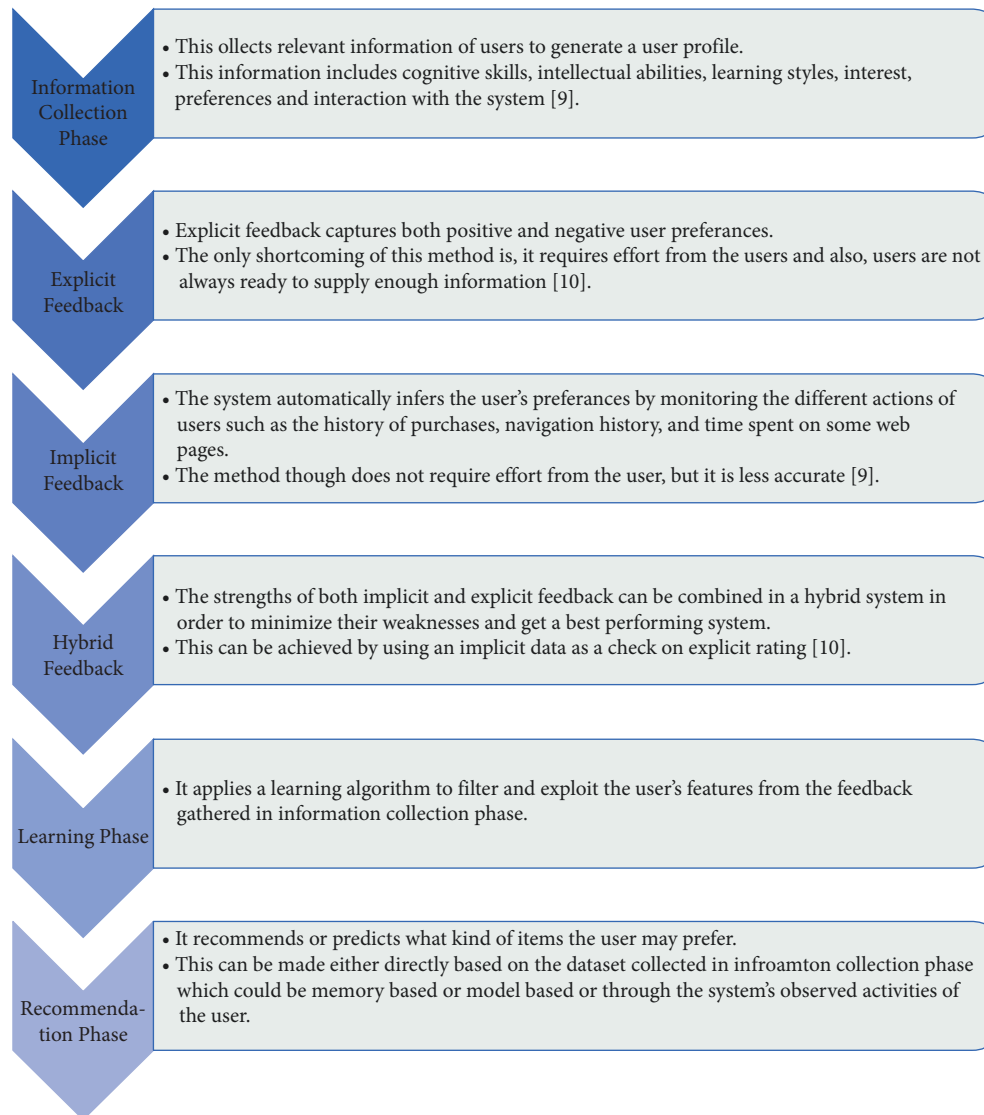
Figure 7: Phases of recommendation.

similarity between users. The reason behind using the Euclidean distance metric for determining the similarity identification between two users is because it is simple to implement and can give the difference in ratings given by different users for commonly watched movies. Other mechanisms for calculating similarity are cosine similarity, Pearson's correlation coefficient, K-means clustering, the Jaccard method, etc.

(iii) Thirdly, the system will recommend movies to the concerned user; for instance, if the user "a" has the highest similarity with user "b" then the system will recommend movies from the watching history of user "b."

(iv) Fourthly, user "a" watches the recommended movies and gives them ratings, resulting in a slight change in the similarity between users. There are two types of similarities defined in our proposed model, S_1 is the similarity among users before watching the recommended movies and S_2 is the similarity among users after watching the recommended movies. The difference between the two similarities S_1 and S_2 is defined as mutual liking.

(v) Lastly, the degree of friendship is the product of mutual liking and an intersection between recommended movie sets of two concerned users.

Our proposed recommendation system uses a modular approach, as we have devised several algorithms and used these algorithms to fulfill the research objective mentioned earlier. Firstly, we determine the similarity between users regarding common movies in Algorithm Obj_1, fulfilling O_1. Secondly, in Algorithm Obj_2, we recommend movies to the specific user based on the similarity. After the recommendations, the user might have finished watching the recommended movies and usually did the rating of the watched movies also. Hence, we again determine a new similarity among those sets of users

TABLE 1: Related work.

S. no.	Paper reference	Techniques used	Key findings	Observations
1	[10]	Probabilistic factor analysis	Determines authenticity and the realistic nature of news	Calculates the probability of news from news data sets collected by different websites
2	[11]	Collaborative filtering	Describes profile similarity and trust-based recommendation systems	A correlation between trust and the largest single difference in score exists along with overall similarity
3	[12]	Collaborative filtering	Describes social networks based on semantic networks and trust for movie recommendations	Creates predictive rating recommendations for movies
4	[13]	KNN	Includes multigraph ranking model	Results represent different user relationships in multiple graphs and recommend the nearest neighbors of specific users
5	[14]	Probabilistic factor analysis	Includes trust-based recommendations in a social network	It has lower recommendation probabilities
6	[15]	KNN	Includes a hybrid approach	Combines user ratings and social trust. Compared with other trust-aware recommendation work, their method uses untrusted links and investigates their dissemination effect
7	[16]	Collaborative filtering	Includes user relevance and evolutionary clustering	Correlation is calculated by combining user satisfaction and potential score information
8	[17]	Collaborative filtering	Includes a new similarity calculation method called JacRA	Calculates selection of the items and the ratings. It has complex calculations to determine ratings
9	[18]	Collaborative filtering	It solves data scarcities, cold start, recommendation accuracy, and timelines as an improved collaborative recommendation algorithm	It involves the traditional similarity of the collaborative recommendation algorithm, for an improved recommendation
10	[19]	Matrix factorization	Includes a hybrid approach combining social behavior, movie genre, and existing collaborative filtering algorithms	Calculates similarity of movies to predict user ratings
11	[20]	Collaborative filtering	Includes collaborative filtering based on the ratings of the movies implemented by Apache mahout	Considers user ratings to recommend movies
12	[21]	Matrix factorization	Includes model-based approach using matrix factorization techniques in social networks	It resolves the cold start problem to some extent using social MF
13	[22]	Probabilistic factor analysis	Includes probabilistic factor analysis method that calculates multifaceted trust relationships and user profiles by sharing the user's potential feature space	It cannot generate better results using trust relations for predictions
14	[23]	Content-based, collaborative filtering	It includes a collaborative filtering approach and uses the information provided by users	It provides suggestions to the users using the two renowned algorithms
15	[24]	Matrix factorization	Includes dual role preferences (trustee/trustee specific preferences), and trust-aware recommendations are achieved by modeling explicit interactions	Using explicit interactions makes it difficult to compute due to privacy issues

using Algorithm Obj_1, fulfilling O_2. Thirdly, to determine the mutual liking we use Algorithm Obj_3 by subtracting old similarity from new similarity, fulfilling O_3. Fourthly, we use Algorithm Obj_4 to determine the degree of friendship by multiplying the value of mutual liking with many commonly recommended movies between users, fulfilling O_4. The section below presents the algorithms used in the proposed system.

4.2. The Algorithms. In our model, Algorithm Obj_1, fulfilling O_1, and Algorithm Obj_2, fulfilling O_2, were presented in [25]. As shown in algorithms 1 and 2, we present Algorithm Obj_3, fulfilling O_3, and Algorithm Obj_4, fulfilling O_4.

Global Assumptions: we have assumed the following three sets:

$U = \{u_1, \ldots, u_n\}$, the set of users

$R = \{r_1, \ldots, r_n\}$, the set of ratings

$M = \{m_1, \ldots, m_n\}$, the set of movies

5. Experimental Setup

We have implemented our model with Python on the Google Colab platform and a cloud-based Jupyter Notebook environment.
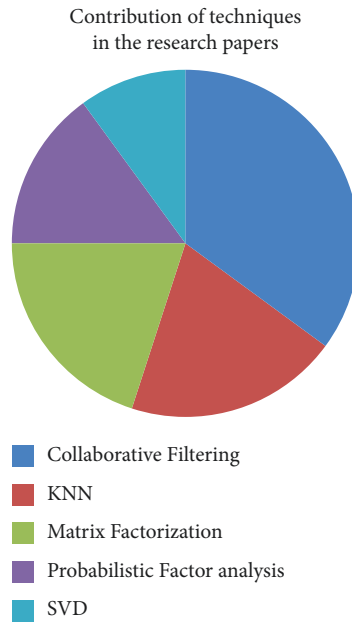
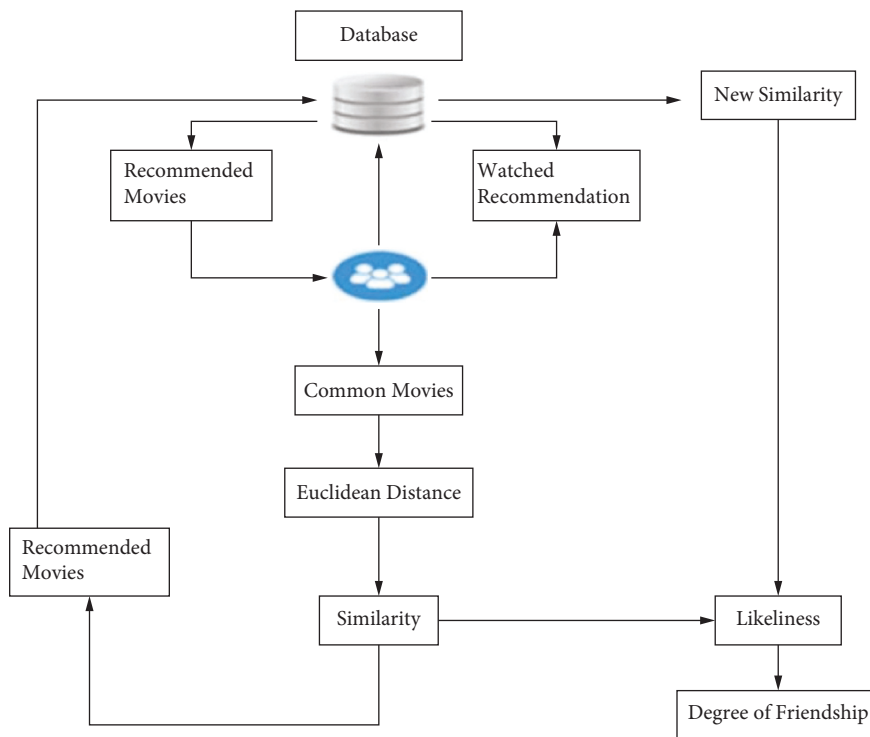FIGURE 8: Contribution of recommendation techniques in related papers.

FIGURE 9: Proposed model.

5.1. Data Collection. Data collection is a very important task for our implementation. We created our dataset by questionnaire method asking for the users' ratings through Google forms. Those users were connected through the social network also. Our dataset of 4000 records has a username, a list of movies watched, and ratings (1 to 5) for every watched movie given by the user. We cleaned it and used it for our research. From the dataset, we have created a data dictionary. The data dictionary has a user name, a list of movies watched, and ratings for every watched movie.

5.2. Programming Language Used. We used Python programming language to implement our approach. Python

Algorithm Obj$_3$: Determines mutual liking between two users.
Input: Name of user u_a
Output: Mutual liking between user u_a and the rest of the users in the user set $U = \{u_1, \ldots, u_n\}$
Begin
(1) Compute the Euclidean distance between movie ratings using the following equation [25, 26]:
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p}_1 - \mathbf{q}_1)^2 + (\mathbf{p}_2 - \mathbf{q}_2)^2}.$$
Where p_i and q_i are movie ratings and $d(p, q)$ is the Euclidean distance between ratings of the common movies.
(2) Compute the similarity, s_1 of user u_a with user u_b before user u_a has watched the recommended movies using the following equation [25]:
$$\mathbf{s}_1 = 1/(2 + \mathbf{d}(\mathbf{p}, \mathbf{q})).$$
Where $d(p, q)$ is the Euclidean distance between ratings of the common movies.
(3) Computer the similarity, s_2 of user u_a with user u_b after user u_a has finished watching the recommended movies using the above equation.
(4) Compute mutual liking between users' u_a and u_b using the following equation:
$$\mathbf{mutual\,Liking}(\mathbf{u_a}, \mathbf{u_b}) = |\mathbf{s}_1 - \mathbf{s}_2|.$$
(5) Recommend the selected movie M_i to user u_a
End

ALGORITHM 1: Algorithm Obj$_3$.

Algorithm Obj$_4$: Determines the degree of friendship between two users.
Input: Name of any two users from the user set $U = \{u_1, \ldots, u_n\}$
Output: Degree of friendship
Begin
(1) Determine recommended movie list $\mathbf{M_a} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \ldots \mathbf{m_n}\}$ for user u_a using algorithm Obj$_2$.
(2) Determine recommended movie list $\mathbf{M_b} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \ldots \mathbf{m_n}\}$ for user u_b using algorithm Obj$_2$.
(3) Compute the friendship weight using the following equation:
$$\mathbf{F_w} = \mathbf{the\,count\,of\,elements\,in\,the\,set}(\mathbf{M_a} \cap \mathbf{M_b})$$
(4) Compute the degree of friendship using the following equation which is the product of the mutual liking computed from using final equation of Algorithm 1 of algorithm Obj$_3$ and the friendship weight computed from the mutual Liking equation given in Algorithm 1:
$$\mathbf{degree\,Of\,Friendship} = \mathbf{mutual\,Liking}(\mathbf{u_a}, \mathbf{u_b}) * \mathbf{F_w}$$
End

ALGORITHM 2: Algorithm Obj$_4$.

focuses on the core functions of the application by handling functional tasks. In Python, it is easier to maintain the code and the application [27].

6. Results and Discussion

The implementation shows that we have achieved our objectives successfully, and as the mutual liking among the recommendations will increase, the degree of friendship between the users will also enhance. Therefore, for an instance and instinctively, if there is a greater degree of friendship among two users, it will be highly likely to follow the recommendations suggested by the latter to the former. Also, the friendship between two users can be determined based on their mutual liking. These results are presented below from the executing environment of Google Colab. Figure 10 shows the similarity S1 of user "aqeelkhalique" with other users in the database before user "aqeelkhalique" watched any recommended movie. Figures 11 and 12 show the recommendation of movies and similarities after the user has finished watching the recommended movies.
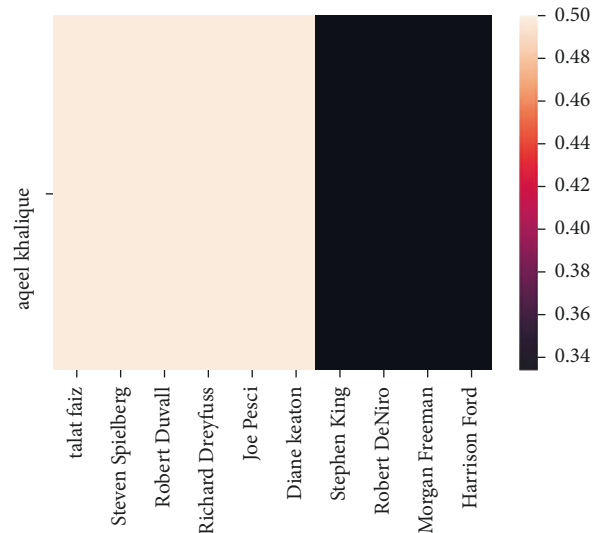

FIGURE 10: Similarity, S_1 between the specified user and the rest of the users before watching the recommended movie.

FIGURE 11: Recommendation suggested by the proposed model for the specified user based on similarity.
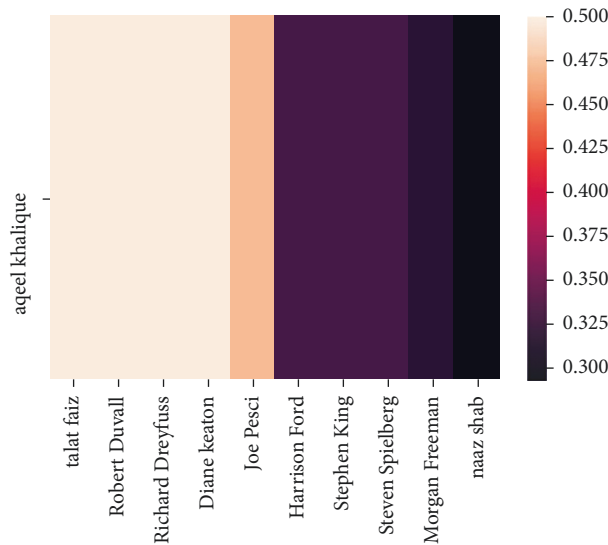

FIGURE 12: Similarity S_2 among users after they have finished watching the recommended movies.
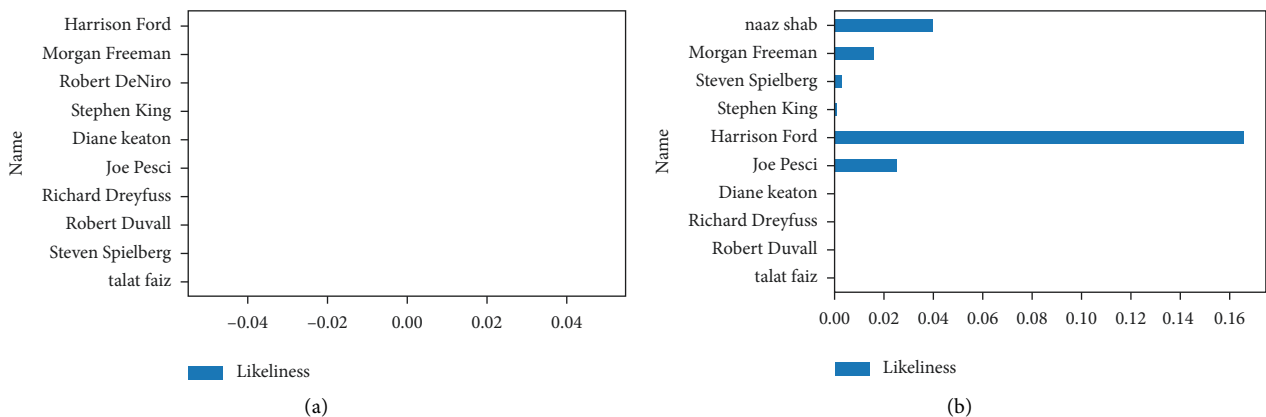

FIGURE 13: (a) Mutual likings between the specific users and the rest of the users before the user started watching the recommended movies. (b) Mutual likings between the specific user "aqeelkhalique" and the rest of the users after the user "aqeelkhalique" finished watching the recommended movies.
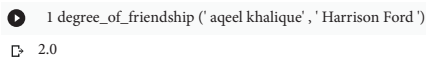

FIGURE 14: Degree of friendship.

Figures 13(a) and 13(b) show mutual liking among user "aqeelkhalique" and the rest of the users before watching recommended movies and submitting ratings after watching recommended movies, respectively.

Figure 14 finally shows the degree of friendship between "aqeelkhalique and Harrison Ford" after determining mutual liking among the concerned users.

The degree of friendship as determined in the results clearly indicates the relationship between mutual liking and similar recommendations. There exist several other mechanisms for recommending movies. However, determining the degree of friendship makes our proposed methodology unique. Hence, based on mutual liking, an effective recommendation system can determine the degree of friendship or social bond among users. Also, once mutual liking will increase, subsequent recommendations can be more effective as it depends on mutual liking and the personal preferences of users. The model can be scalable with the growing dataset as the number of users and ratings for watched movies will increase substantially. Prominent giants such as Netflix also used cold start and survey/questionnaires initially to generate recommendations [28, 29].

7. Conclusion and Future Work

In OTT platforms, it has been observed that the mutual liking between users can enhance the recommendation system's performance. In this study, we have proposed a hybrid approach to adapt the factors to user preferences and social connections to make accurate recommendations. Generally, when a user watches a movie suggested by the system, the mutual liking of the users for the movies may vary accordingly. Using mutual likings, we have defined a degree of friendship between users. With the help of our proposed recommendation system, the user will be able to know that among his/her friends who are closer or likable to him/her determined by the degree of friendship. Our model is novel and unique in its algorithmic approach and has never been implemented on any OTT platforms so far.

The results indicate that mutual likings and degree of friendship information are beneficial in recommending movies on an OTT platform. Our proposed model is novel and has been implemented using our algorithms. The experiments performed have a limitation on the available dataset as the proposed method will require scalable information about realistic user ratings.

Future work may not be limited to including intelligent insights based on analytics, optimization methods, and scalable real-time dataset to integrate different types of information and other factors to improve the performance of the recommendation systems. The influence of social networking may result in more accurate recommendations due to similar interests, tastes, mutual likings, etc.

Data Availability

The data are available on request from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the research article with any parties.

References

[1] Medium, "Introduction to recommender systems," 2021, https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada.

[2] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[3] H. Fang, Y. Bao, and J. Zhang, "Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 30–36, AAAI, Québec, Canada, July 2014.

[4] G. Guo, J. Jennifer Zhang, and N. Yorke-Smith, "TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 123–129, AAAI, Austin, TX, USA, January 2015.

[5] H. Bhardwaj, P. Tomar, A. Sakalle, and A. Bhardwaj, "Classification of extraversion and introversion personality trait using electroencephalogram signals," in *Proceedings of the International Conference on Artificial Intelligence and Sustainable Computing*, pp. 31–39, Springer, Gwalior, India, November 2021.

[6] H. Bhardwaj, P. Tomar, A. Sakalle et al., "A DeepLSTM model for personality traits classification using EEG signals," *IETE Journal of Research*, vol. 2021, pp. 1–9, 2022.

[7] B. Cui, "Design and implementation of movie recommendation system based on knn collaborative filtering algorithm," in *Proceedings of the ITM Web of Conferences, Information Theory and Information Systems*, pp. 1–5, Shanghai, China, March 2017.

[8] What is a recommendation engine and how does it work, "Express Analytics," 2021, https://expressanalytics.com/blog/recommendation-engine/.

[9] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2/3, pp. 103–130, 1997.

[10] C. Porcel and E. Herrera-Viedma, "Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries," *Knowledge-Based Systems*, vol. 23, no. 1, pp. 32–39, 2010.

[11] Z. Xia, S. Xu, N. Liu, and Z. Zhao, "Hot news recommendation system from heterogeneous websites based on bayesian model," *The Scientific World Journal*, vol. 2014, Article ID 734351, 8 pages, 2014.

[12] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *ACM Transactions on the Web*, vol. 3, no. 4, pp. 1–33, 2009.

[13] J. Golbeck and J. Hendler, "FilmTrust: movie recommendations using trust in web-based social networks," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference*, pp. 282–286, IEEE, Las Vegas, NV, USA, January 2006.

[14] M. Mao, J. Lu, G. Zhang, and J. Zhang, "Multirelational social recommendations via multigraph ranking," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4049–4061, 2017.

[15] T. Li, A. Liu, and C. Huang, "A similarity scenario-based recommendation model with small disturbances for unknown items in social networks," *IEEE Access*, vol. 4, pp. 9251–9272, 2016.

[16] W. P. Lee and C. Y. Ma, "Enhancing collaborative recommendation performance by combining user preference and trust-distrust propagation in social networks," *Knowledge-Based Systems*, vol. 106, pp. 125–134, 2016.

[17] J. Chen, C. Zhao Uliji, and L. Chen, "Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering," *Complex & Intelligent Systems*, vol. 6, no. 1, pp. 147–156, 2019.

[18] X. Wu, Y. Huang, and S. Wang, "A new similarity computation method in collaborative filtering based recommendation system," in *Proceedings of the IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, IEEE, Toronto, ON, Canada, September 2017.

[19] H. Chen, Z. Li, and W. Hu, "An improved collaborative recommendation algorithm based on optimized user similarity," *The Journal of Supercomputing*, vol. 72, no. 7, pp. 2565–2578, 2015.

[20] C. Yang, X. Chen, L. Liu, T. Liu, and S. Geng, "A hybrid movie recommendation method based on social similarity and item attributes," *in Advances in Swarm Intelligence*, Springer, Cham, Switzerland, pp. 275–285, 2018.

[21] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social

networks," in *Proceedings of the Fourth ACM Conference on Recommender Systems-RecSys'10*, pp. 135–142, Barcelona, Spain, September 2010.

[22] L. Guo, J. Ma, and Z. Chen, "Learning to recommend with multi-faceted trust in social networks," in *Proceedings of the 22nd International Conference on World Wide Web-WWW'13 Companion*, pp. 205-206, Rio de Janeiro, Brazil, May 2013.

[23] A. Pawlicka, M. Pawlicki, R. Kozik, and R. S. Choraś, "A systematic review of recommender systems and their applications in cybersecurity," *Sensors*, vol. 21, no. 15, p. 5248, 2021.

[24] W. Yao, J. He, G. Huang, and Y. Zhang, "Modeling dual role preferences for trust-aware recommendation," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 975–978, Gold Coast, Australia, July 2014.

[25] M. Saquib, A. Khalique, and I. Hussain, "A study of recommendation system on OTT platform and determining similarity and likeliness among users for recommendation of movies," *Proceedings of Data Analytics and Management*, vol. 2021, pp. 537–546, 2021.

[26] M. K. I. Rahmani and M. A. Ansari, "A color based fuzzy algorithm for CBIR," in *Proceedings of the Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, pp. 363–370, IET, Noida, India, September 2013.

[27] "Python 7 important reasons why you should use python solutions," Mindfire Solutions, 2017, https://medium.com/mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b.

[28] D. Chong, "Deep dive into Netflix's recommender system," *Towards Data Science*, Canada, 2020, https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48.

[29] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system," *ACM Transactions on Management Information Systems*, vol. 6, no. 4, pp. 1–19, 2016.