INTERNATIONAL CONFERENCE ON ADVANCES IN TECHNOLOGY AND COMPUTING (ICATC-2021)
Faculty of Computing and Technology (FCT), University of Kelaniya, Sri Lanka
18th December 2021

Index N0: CT-03-37

Hybrid Movie Recommendation System

Sinthujan Punitharasa (1st Author)
Faculty of Information Technology
University of Moratuwa
Moratuwa, Sri Lanka
sinthujanspp@gmail.com

Kirisanthi Selvanajagam (2nd Author)
Faculty of Information Technology
University of Moratuwa
Moratuwa, Sri Lanka
kirisanthiselvanajagam@gmail.com

Thamilini Ramakrishnan (3rd Author)
Faculty of Information Technology
University of Moratuwa
Moratuwa, Sri Lanka
thamiliniram96@gmail.com

Amalraj Chelvarajah (4th Author)
Faculty of Information Technology
University of Moratuwa
Moratuwa, Sri Lanka
amalraj@uom.lk

W.M.R.M. Weerasinghe (5th Author)
Faculty of Information Technology
University of Moratuwa
Moratuwa, Sri Lanka
ruviniw@uom.lk

***Abstract*- Movie recommendations play a great part in the aspects of our social life. Such a system allows users to recommend a group of films based on their interests or the popularity of the movies. This research was conducted to study different approaches to movie recommendation and discusses a hybrid approach that combines a content-based filter, a collaborative-memory-based filter, and a collaborative-model-based filter. The proposed system aims to reduce the issues with existing movie recommendation systems by enhancing performance. The content-based filter is based on a TF-IDF classifier with cosine similarity. The collaborative-memory-based filter is based on truncated SVD with Pearson correlation. A collaborative model-based filter is based on improved SVD matrix factorization.**

Keywords- Movie Recommendation System; Content-based filtering; Collaborative- memory-based filtering; Collaborative- model-based filtering; Hybrid approach.

I. INTRODUCTION

Due to the growth of the Internet and Information Technology, the data transactions volumes that are happening every second have increased dramatically. Currently, movie releases per year have increased with the number of movie watchers. However, every movie watcher has different interests. Therefore, not all data related to movies available on the internet gives results that give satisfaction to the movie watchers. Movie data in such a large amount will often become irrelevant, and without proper processing of this information, it will be wasted. In such cases, movie watchers have to run their search several times before they get the first look. Movie watchers have to spend a lot of time finding the desired/ interesting movies. But due to workloads, they are not willing to spend much time searching. Researchers have come up with movie recommendations to overcome this problem. A movie recommendation system provides needed information by considering the movie watcher's past choices. Movie data is filtered and customized to suit the needs of the movie watcher.

The movie recommendation systems have proven to be a powerful tool in providing satisfactory movie suggestions for movie watchers. The recommendations are made to help movie watchers efforts to handle information loads and to help them find the right movies quickly and easily [1]. It is a kind of filtration system, which tries to predict a movie watcher's preferences and recommends movies. As with movie recommendation systems, there are many other types of applications for recommendation systems. For example, Netflix can track users' interactions with various kinds of stories in users' feeds to find out what 2 kinds of stories/movies attract users. Sometimes, movie recommendation systems can improve their system by monitoring the activities of a large number of people. The movie recommendation system is an implementation of the machine learning algorithms [2, 3].

Movie Recommendation Systems are classified into four broad groups, namely Collaborative filtering systems, Content filtering systems, Knowledge-based systems, and Hybrid systems [4]. The Content-based filtering movie recommendation system works by assessing the behavioral needs of new movie watchers. This is a kind of keyword-specific recommendation system where specific keywords are used to identify movies. The collaborative filtering-based movie recommendation systems are based on similarity measures between the movie watcher's data/information and the movies. Movies are suggested to the new movie watcher by considering the other movie watcher's past browsing history. The collaborative filtering system uses the mean rating of the movies, recognizes similarities based on movie watchers' reviews, and generates new recommendations based on comparisons between movie watchers [5]. The knowledge-based movie recommender system attempts to recommend movies based on assumptions about a movie watcher's needs and preferences [6]. The hybrid movie

recommendation system performs its functions considering the integrated behavior of content-based and collaborative filtering techniques to suit a particular film. The hybrid movie recommendation system is considered the best movie recommendation system because of its ability to eliminate weaknesses. These hybrid movie recommendation systems usually build upon the programming languages like python / C++ [3].

II. RELATED WORK

Over the past decade, several recommendation systems have been developed for various fields like books, cosmetics, novels, movies, and are in use. These recommendation systems are implemented based on a variety of approaches, including the content-based filtering approach, the collaborative-based filtering approach (memory-based & model-based), the geographic approach, the knowledge-based approach, the utility-based approach, and the hybrid approach.

A. Content-based filtering

In movie recommendation, content-based filtering is a moderated learning algorithm [11]. This filtering technique gives recommendations for a movie by comparing the movie profile and the customer profile. Movies are recommended to users based on what the users like [11]. SRS Reddy et. al. proposed a content-based filtering system using genre correlation [12]. This system aims to recommend movies to users based on the similarity of genres. If the user rates highly for a particular movie, then this system recommends other movies with similar genres to that user. They used the MovieLens dataset in their research as and R is used as the data analysis tool. In this research, the main dataset is divided into two subsets. Its subgroups also have a list of movies with the genres in which they are classified. The other subset contains a list of user-rated movies on a scale of 1 to 5, with 5 being the highest [12].

Ramni Harbir Singh et. al. suggested a method for recommending movies to users using cosine similarity and KNN [13]. This system provides general recommendations for each user, based on popularity and/or the type of movie. They used cosine similarity and content-based filtering to predict the outcome and recommend a movie to the user by running the code in python using the NumPy and Panda libraries. This system can store a large amount of data and give effective results. This recommendation system searches for the best movies similar to the movie users watched based on genre and gives results. The researchers used cosine similarity to measure the similarity between movies based on different properties. Along with cosine similarity, they used KNN to find the nearest neighbor. It gives more accuracy than other distance metrics and the complexity is relatively low [13].

B. Collaborative Memory-based filtering

Ramil G. Lumauag et. al. proposed an improved collaborative memory-based algorithm by formulating a similarity measure to identify the number of co-rated movies, calculate the user's similarity by selecting the nearest neighbor. They used the Movielens 100K dataset for their research. This dataset includes 943 users and 1682 movies and also it contains 100,000 rating records on a scale of 1 to 5 points, with 1 being the lowest and 5 being the highest. They found that the MAE of their improved algorithm has a small prediction error compared to the traditional collaborative filtering algorithm [14].

Kyung Soo Kim et. al., proposed a content metadata-based approach that uses content metadata efficiently. In their research, they collected a large amount of user movie ratings and user trust network information from Epinions.com and crawled movie metadata from the web. The dataset collected consists of 91,735 users, 611,741 user trust links, 26,527 movie content items along with their metadata, and 170,797 user content reviews. Furthermore, they provided two content representations that were generated from content metadata such as content-metadata TF-IDF vector and content link network. Content metadata was used to improve the process of finding live content for a target content element in the context of an object-based CF. Content metadata exploited with current user content rating scores and confidence in networks to predict classification results more accurately for target contents [15].

C. Collaborative Model-based filtering

Rohan Mehter et. al. suggested a movie recommendation engine using collaborative filtering with two algorithms namely Alternative Least Square (ALS) and Single Value Decomposition (SVD) [16]. This movie recommendation system was once created with an Alternative Least Square algorithm and again with the Singular Value Decomposition algorithm. The researchers used Apache Spark, a machine learning library, to train the dataset more efficiently and with flexible search to retrieve results quickly. Researchers used the MovieLens dataset to recommend movies. This movie data set contains an anonymous movie rating. They have used Jupyter notebook for machine learning, data visualization, and code performance. They integrated the recommendation system using the Python programming language. Apache Spark is used to process big data while HDFS used to store a large amount of data which is processed with spark according to their study [16].

Mirza Ilhami et. al. proposed a system for recommending films using the Matrix factorization and collaborative filtering. They have used the MovieLens dataset in their research which contains 855,598 ratings for 10,197 movies and 2,133 users. They labeled the rating data as a three-way list representing user ID, item ID, and ratings. The researchers have split the data set into two parts, 80% for training and 20% for algorithm testing. The dataset contains actors, countries, directors, and genres. However, for

experiments, the researchers used the rating data from users to make recommendations. In this approach, first, normalize the data to remove bias. Second, perform a matrix analysis to obtain forecasts and recommendations, and it has also been used to remove element space dimensions and restore inherent relationships between dataset elements [17].

To overcome the limitations of the content-based filtration approach and the collaborative-based filtration approach, a hybrid approach was introduced. Rahul Katarya et al., proposed a hybrid and optimization-based technical suite to enhance the predictive accuracy of films. And this proposed solution helps to reduce the limitations of the collaborative filtering-based recommendation system. They use K-mean as the Clustering algorithm and Cuckoo search as an optimization algorithm. In their research, they apply these two algorithms to the MovieLens dataset to optimize the effective recommendation systems [18].

III. PROPOSED SYSTEM

There are a lot of approaches to recommend movies to users. In our hybrid recommendation approach, we combine content-based filtering and collaborative filtering to provide better results to users. Fig.1 elaborates the top structure of the proposed system.

There are 3 main modules in this system named collaborative memory-based filtering, content-based filtering, and collaborative model-based filtering.

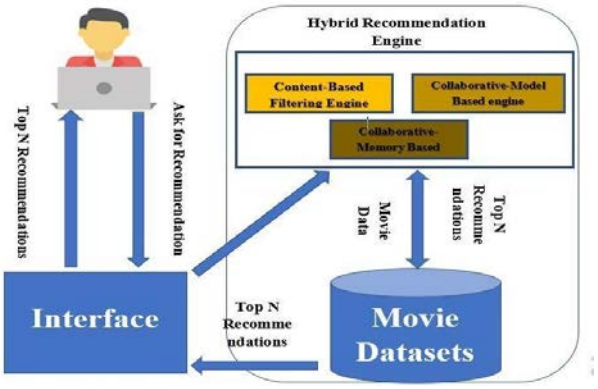

Fig. 1. Top Level Architecture for Proposed System

A. *Content-based filtering*

The content-based filter uses data about movies, users and looks for similarities before recommending movies. The similarity of different movies computed based on user past and current data. There are different techniques or similarity measures used to calculate similarity. In this paper, we use the cosine similarity technique to find similar users. This scale is used to calculate textual similarity data. We convert this text movie data into vectors and check the cosine angle between these two vectors. If the angle between them is 0 or 1, then they are the same or not.

There are various concepts to build content-based filters. Term Frequency-Inverse Document Frequency (TF-IDF) is one of those concepts used in our study. Fig.2 shows the content-based filter approach.

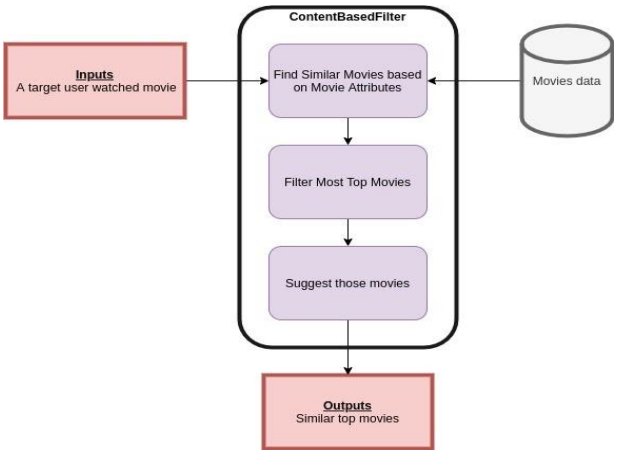

Fig. 2.Content-based filter

B. *Collaborative- Memory-based filtering*

The memory-based collaborative filtering approaches can be further subdivided into two main parts namely user-based filter and item-based filter. User-based filtering takes a specific user, searches for users similar to that user, based on rating similarity, and recommends movies depends on similar users' interest. But, item-based filtering takes a pair of movies and finds the similarity of those movies by considering the rating of those pairs of movies by all users.

There are various concepts to build this module. Truncated SVD is one of those concepts which is used in our study. Unlike regular SVDs, a truncated SVD generates a factorization where the number of columns for multiple disconnections can be specified. It works with sparse matrices efficiently. Along with Truncated SVD, the Pearson correlation similarity technique is used to find similar items. Fig.3 shows the collaborative- memory-based filter approach.

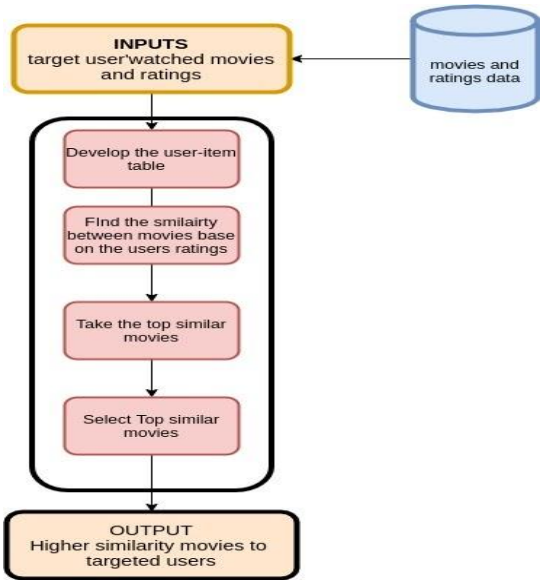

Fig. 3. Collaborative- Memory-based filter

C. *Collaborative- Model-based filtering*

The model-based filter learns a user's preferences by directing itself on past user interactions and creating a user profile. This model-based approach considers a basic "generative" model, which explains user-movie interactions and attempts to detect them to make new predictions. This model is trained to reconstruct the values of user movie interactions from the own representation of users and movies. New movie recommendations can be made based on this model.

There are various concepts to build this module such as matrix factorization, clustering, and deep learning. In this paper, SVD, which is a classical method of linear algebra is used to implement a model-based filter. Matrix factorization works by decomposing a user-item interaction matrix into the product of two low-dimensional rectangular arrays. Not every user gives ratings to all movies, there are many missing values in the matrix and this results in a sparse matrix. SVD provides a solution for the sparsity problem. It uses a matrix framework where each row represents a user, and each column represents a movie. Elements of this matrix are the ratings that users give to movies. Fig.4 shows the collaborative-model-based filter approach.

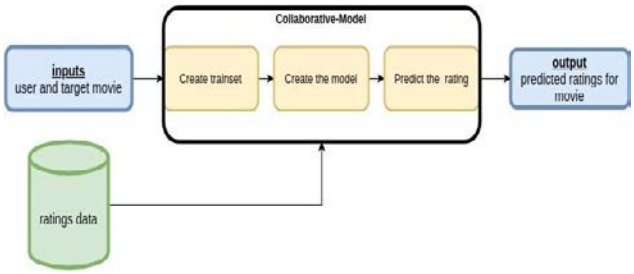

Fig. 4. Collaborative- Model-based filter

IV. IMPLEMENTATION

To implement the proposed system, we used the TMDB dataset which combines movie plot description, Metastore ratings, comments and user ratings and reviews, release dates and many more aspects. In this data, we have four sub-datasets which are the movies dataset includes 45,000 movies with attributes such as movie description, average rating, number of votes, genre, etc. The rating dataset includes 26 million rating details from 270,000 users. 10000 movie data and 100000 rating data are used in our system.

A. *Content-based filter*

Under the content-based filtering approach, the title, cast, crew, genres, keywords, and the original language of movies are extracted and TF-IDF vector matrix is created to get rid of weight down (lose the accuracy) the features of movies and based on that matrix find the similarity of the target movies whose have watched by target user by using similarity algorithms and suggest the top similarity movies to that user.

To find the similarity between the movies a similar matrix is created by using TF-IDF classifiers and cosine similarity metrics (based on TF-IDF vector).

B. *Collaborative-Memory-based filter*

An item-based memory filter is also considered in this study. Movie rating data that was given by users is also used. To provide the recommendation to the target user, the user-movies matrix which has a similarity ratio of each movie for that target user based on the ratings of other users is used. Then, the top most similar movies are considered to find their top similar movies which are suggested to the target user.

To find the similarity between the movies, Pearson's correlation coefficient matrix is used. It is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample. The Truncated SVD classifier is used to create a user-item matrix for finding the similarity between movies.

C. *Collaborative-Model-based filter*

Panda and surprise python libraries are used under the collaborative-model-based filter. The panda package is used to read the ratings which are in CSV format and convert them into a python data frame. The surprise dataset function is used to convert the rating data frame as a train set for the model-based predictions algorithms. Once the train set is created based on the users' ratings over the movie, the train set is used to predict the target user's ratings over a movie as a test set by using the surprise SVD functions. Surprise SVD classifier is used to create the model based on the train data of user rating, and predict the rating for the test data. The hyper-parameters of the SVD were tuned to minimize the error values.

D. *Hybrid Approach*

By adding either a content-based filter with the collaborative-model-based filter or a collaborative-memory-based filter with a collaborative-model-based filter, the hybrid approach is implemented. If it is the content-model hybrid approach, the user is selected first and find the high rated movies of that user then find the most similar movies and for those movies then take those most similar movies to predict ratings on behalf of that user and out of those prediction ratings, then top rated movies are selected and suggest the top similar and high rated movies to users.

For the item-model hybrid approach, the user is selected first and find the top-rated movies and for those top rated movies find the similar movies then took those most similar movies and predict rating and suggest the top similar and high rated movies to users.

V. EVALUATION

Cross-validation is performed to assess the effectiveness of the model. This is a statistical method used to evaluate the performance of our models. It is used to protect against the overfitting of the predictive model, especially when the amount of data is small. In cross-validation, the system generates a certain number of folds, in this study it is five, which carry out the analysis in each part and then calculate the average total error rate. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were selected to perform cross-validation on Collaborative- memory & model filters. With the use of metrics in Sci-kit learn, MAE and RMSE are calculated. And precision, recall, and F1 are used to evaluate content-based filters.

The precision is calculated considering predicted values and true values. Equation 1 shows the formula of precision. The recall means how many values are true in prediction out of real true values. Equation 2 shows the formula of recall. F1 score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not easy to understand as accurate, but F1 is usually more useful than accuracy. Equation 3 shows the formula for the F1 score.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

MAE is the difference between the actual value (rating) and the predicted value. Equation 4 shows the formula of the MAE matrix. In 4, $\mathbf{y_i}$ denotes the observed value for the i^{th} observation, $\mathbf{x_i}$ denotes the predicted value for the i^{th} observation and **n** denotes the total number of observations. RMSE is similar to MAE but the only difference is that the absolute value of the residual is squared and the square root of the whole term is taken for comparison. Equation 5 shows the formula of the RMSE matrix. The advantage of using RMSE over MAE is that it penalizes the term more when the error is high.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}} \quad (5)$$

A. *Content-based filter*

Table 1 shows the testing results of content-based filters.

Table 1. Content-based filter results

Filter System	**Precision**	**Recall**	**F1 Score**
Content-Based	0.83	0.83	0.84

From Table 1, we get to know about the prediction accuracy of our content-based recommendation system based on the Precision, Recall, and F1 scores. Where it shows the Precision and Recall as 0.83 and the F1 score as 0.84. The F1 score provides a more Standard way of accuracy measurement by doing the weights of Precision and Recall.

B. *Collaborative- Memory based filter*

The following Table 2 shows the testing results of a collaborative- memory-based filter where each fold contains a different combination of test and train data. Based on the train and test data the average RMSE and MAE of the

Collaborative model are identified as 0.8999, 0.6939 respectively.

Table 2. Testing Results of Collaborative Memory

Metrics	RMSE	MAE
Fold 1	0.9045	0.7003
Fold 2	0.8886	0.6864
Fold 3	0.9070	0.6995
Fold 4	0.8886	0.6780
Fold 5	0.9108	0.7053
Mean	0.8999	0.6939

a.

C. *Collaborative- Model-based filter*

Table 3 shows the testing results of the collaborative-model-based filter where each folds folding a different combination of test and train data. Based on the train and test data the average RMSE and MAE of the Collaborative model are identified as 0.8359, 0.6356 respectively.

Table 3. Testing Results of Collaborative Model-Based Filters

Metrics	RMSE	MAE
Fold 1	0.8349	0.6348
Fold 2	0.8370	0.6362
Fold 3	0.8384	0.6379
Fold 4	0.8358	0.6349
Fold 5	0.8336	0.6343
Mean	0.8359	0.6356

D. *Hybrid System*

In this study, each filter is improved, when these filters are combined as a hybrid system which can be either a content-based filter with a collaborative-model-based filter or a collaborative- memory-filter with a collaborative-model filter, the performance of the hybrid system will be increased..

VI. CONCLUSION & FUTURE WORK

The recommendation system proposed in this paper is a Hybrid type of recommendation system where each of the filters has been analyzed with the possible classifiers and created based on the best and improved method according to the evaluations. The content-based filter has been made by adding more movie features. This will improve the accuracy of the filter to align and recommend a similar movie to users. The item-based collaborative- memory filter has been implemented and this will improve its accuracy and scalability compared with the user-based approach. The collaborative- model has been created with an improved model. It will lead reducing the difference between the actual and predicted movies based on user ratings. Further, for future work, the execution time of the system has to be reduced to provide immediate response with high accuracy.

REFERENCES

[1] G. Rodríguez, “Introduction to Recommender Systems,” Tryolabs.com, 09-May-2018. [Online]. Available: https://tryolabs.com/blog/introduction-to-recommender-systems/. [Accessed: 13-Dec-2021].
[2] B. C. A. G. P. S. Vishwa Gosalia, "Movie Recommendation System," International Research Journal of Engineering and Technology (IRJET), vol. 05, no. 02, pp. 300-302, Feb-2018. [Accessed: 13-Dec-2021].
[3] Medium.com. [Online]. Available: https://medium.com/@madasamy/introduction-to-recommendation-systems-and-how-to-design-recommendation-system-that-resembling-the-9ac167e30e95. [Accessed: 13-Dec-2021].
[4] “Classifying different types of recommender systems,” Bluepiit.com, 14-Nov-2015.
[5] dataaspirant, Editorial Team, K. Taylor, and K. Wiggers, “An introduction to recommendation engines - dataconomy,” Dataconomy.com, 13-Mar-2015. [Online]. Available: https://dataconomy.com/2015/03/an-introduction-to-recommendation-engines/. [Accessed: 13-Dec-2021].
[6] S. M. F. C. S. D. Geetha G, "A Hybrid Approach using Collaborative filtering," National Conference on Mathematical Techniques and its Applications (NCMTA 18), 2018.
[7] Y. S. P. A. Prateek Sappadla, "Movie Recommender System," Search Engine Architecture, Spring 2017, NYU Courant, 2017.
[8] D. Y. A. S. V. K. G. Manoj Kumar, "A Movie Recommender System: MOVREC," International Journal of Computer Applications (0975 – 8887), vol. 124, no. 3, August 2015.
[9] S. K. A. A. N. P. Gharbi Alshammari, "Improved Movie Recommendations Based on a Hybrid," Vietnam Journal of Computer Science, vol. 6, no. 3, p. 363–376, 2019.
[10] Geeksforgeeks.org. [Online]. Available: https://www.geeksforgeeks.org/python-implementation-of-movie-recommender-system/. [Accessed: 13-Dec-2021].
[11] S. M. S. S. K. S. Roy, 2019 Global Conference for Advancement in Technology (GCAT), pp. 1-5, 2019.
[12] S. N. S. K. S. A. SRS Reddy, "Content-Based Movie Recommendation System Using Genre Correlation," 2018.
[13] S. M. T. T. T. N. G. S. Ramni Harbir Singh, "Movie Recommendation System using Cosine Similarity and KNN," International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 5, June 2020.
[14] A. M. S. R. P. M. Ramil G. Lumauag, "An Enhanced Memory-Based Collaborative Filtering Algorithm Based on User Similarity for Recommender Systems," International Journal of Recent Technology and Engineering (IJRTE), vol. 7, no. 6s, March 2019.
[15] D. S. C. Y. S. C. Kyung Soo Kim, "Boosting Memory-Based Collaborative Filtering Using Content-Metadata," April 2019.
[16] D. P. G. Rohan Mhetre, "Movie Recommendation Engine using Collaborative Filtering with Alternative Least Square and Singular Value Decomposition Algorithms," International

Journal of Advanced Research in Computer and Communication Engineering, vol. 8, no. 2, February 2019.

[17] S. Mirza Ilhami, "Film Recommendation Systems using Matrix," International Conference on Information Technology Systems and Innovation (ICITSI) 2014, pp. 1-6, November 2014.

[18] O. V. Rahul Katarya, "An effective collaborative movie recommender system with cuckoo search," Egyptian Informatics Journal, vol. 18, pp. 105-122, July 2017.