VOL. 18, NO. 13, JULY 2023 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

AN APPROACH FOR MOVIE RECOMMENDATION USING COLLABORATIVE FILTERING WITH SINGULAR VALUE DECOMPOSITION

N. Anusha, Darmoju Deekshitha, Ghadiyaram Bhavya and Buchupalli Mohitha
Vidya Jyothi Institute of Technology, Aziznagar, Hyderabad, Telangana, India
E-Mail: ddeekshitha1305@gmail.com

ABSTRACT

Movie recommendation systems help movie enthusiasts by suggesting movies to watch without the hassle of having to go through the time-consuming process of deciding from a large collection of movie streaming platforms that recommend movies and TV episodes. News organizations that suggest articles to readers, and online stores that suggest products to customers all benefit from these recommendation systems. The algorithms implemented in this research train their models on the MovieLens dataset and provide users with tailored movie recommendations. The study compares different machine learning algorithms, which include a Content-based model, item-item and user-user collaborative filtering (CF), Collaborative filtering with Singular Value Decomposition (SVD), K Nearest Neighbors, and Non-negative Factorization. The algorithms are evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to measure their accuracy and performance. While the proposed system which is based on a collaborative approach using SVD determines the connection between various users and, depending on their ratings, recommends movies to others with similar tastes, subsequently allowing users to explore more. The proposed approach using collaborative filtering with SVD performs better with a minimal RMSE of 0. 880258 by giving accurate and appropriate recommendations to the user. The model is further evaluated using performance metrics like Precision, Recall, and f1 score. So, CF with the SVD recommendation model is chosen for implementation and is integrated into a web application that allows the platform users to rate and review the available digital content as well as allows them to restrict screen time using a parental control system. The results of the study in this paper are presented in the form of tables, graphs, and statistical analyses, and can be used to guide the development of new and improved recommendation algorithms.

Keywords: movie recommendation system, term frequency-inverse document frequency, content-based filtering, collaborative filtering, singular value decomposition.

Manuscript Received 7 April 2023; Revised 30 August 2023; Published 13 September 2023

INTRODUCTION

Many contemporary businesses and organizations rely heavily on recommendation systems to personalize the user experience by making suggestions for goods, services, or materials that would be of interest to the specific user [1]. Informative suggestions are generated by these systems by analyzing users' behavior, preferences, and other related data of the users' anticipated preferences and interests [1] [3]. Digital content streaming services which include a review-rating system and a parental control system to limit the user's screen time can be done [2]. Recommendation algorithms can be of various forms, including content-based filtering, collaborative filtering, and hybrid approaches [3]. The Content-based model uses Term Frequency (TF), Inverse Document Frequency (IDF), and information Gain (IG) to extract features and suggest relevant content [4]. Many surveys are conducted on collaborative techniques for recommendation systems that derive information based on users' choices. [4][5]. Collaborative filtering (CF) includes item-item CF and user-user CF. It constructs item-item and user-user similarity matrices respectively [5].

A probabilistic collaborative filtering approach was put forth by Langseth and Nielsen [13] and used on the MovieLens data. According to the experimental findings, the probability collaborative model can discover hidden variables that model some implicit knowledge about the target domain. In two areas, Bobadilla [14] offered a collaborative filtering technique. One is a joint suggestion from a user group, and the other is a recommendation from a group that is comparable to one that has a referenced item and uses joint collaborative filtering to utilize a limited reference or combine two situations. In a semi-supervised strategy to experiment, Jeong [15] provided a semi-explicit rating method depending on the unrated items. The outcome of the experiment demonstrates that semi-explicit rating data is preferable to pure explicit rating data.

A hybrid approach is implemented by combining both the content-based recommendation approach and the CF model for more accurate results [8]. Collaborative Filtering with SVD is generally used to decrease the database's dimensionality and to suggest personalized recommendations based on a database of explicit product ratings. Some of the researchers compared how two recommender systems affected the generation of Top-N lists using a database of an e-commerce website and if it performed better than SVD [9]. There are other wide ranges of machine learning algorithms, such as neural networks, decision trees, and clustering algorithms [10]. Some of the challenges that exist in recommendation models are sparsity and cold start. In which Sparsity is

defined [15], when we compare the number of users to the number of products, we find that users will only rate a small portion of the entire number of objects. As a result, the User-Item matrix employed in collaborative approaches will have a sparse data structure. The cold-start describes [16] the issue of being unable to recommend new things to existing users is known as cold-start. This is because until enough items/movies have already been rated by the new user, the CF method cannot propose items/movies to them. The CF methodology will also be unable to suggest new things.

METHODOLOGY

The main contribution of the work in this paper is to provide a systematic evaluation of different recommendation algorithms which includes content-based filtering, item-item CF, user-user CF, and collaborative technique with SVD [16] and to identify the most effective algorithms for a given recommendation task. The work involves writing code to preprocess the dataset, implementing the algorithms being evaluated, and computing the evaluation metrics [17]. Recommendation algorithms are typically implemented in this paper using Python, along with some of the data analysis and machine learning libraries like Scikit-learn, Pandas, and NumPy. The study can also provide insights into the factors that affect the performance of recommendation systems, such as dataset characteristics and algorithm complexity.

The MovieLens dataset [11] is widely used in recommendation systems, and it consists of movie ratings that were collected from the MovieLens website. The dataset includes movie ratings, movie metadata (such as genre and year of release), and user demographic information. There are several versions of the MovieLens dataset, with the most used being the MovieLens 100K, 1M, 10M, and 20M datasets [10]. The MovieLens dataset is often used as a benchmark for recommender system algorithms, as it provides a large and diverse set of movie ratings that can be used for evaluation of the performance of different recommendation models.

Content-based filtering and CF models are the most ubiquitous kinds of personalized recommendation systems.

Content-Based Filtering

The Content-Based model recommends based on the similarity of items or users using their genres/description/metadata/profile. Content-based models use TF-IDF [3] to determine the relative importance of movies. TF is the frequency of a word in a document and if a term occurs throughout all documents, the IDF attempts to minimize the weight of that term. [6].

$$TF(t_i) = \frac{\text{Number of times term } t_i \text{ appers in a document}}{\text{Total number of terms in the document}}$$

$$IDF(t_i) = \log_{10} \frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t_i\ in\ it}$$

After calculating TF-IDF scores using the Vector Space Model, the closest objects are determined. It stores each object as a vector in n-dimensional space and calculates the angles between the vectors to identify their proximity. It uses the Tfidf Vectorizer function from sci-kit-learn to transform text to feature vectors and calculates Cosine Similarity as a numeric quantity that denotes the similarity among the movies [7]. There is no quantitative metric to judge content-based algorithms, so it must be done quantitatively. Recommended movies for Toy Story by the content-based model are shown in Fig. 3.

Collaborative Filtering

CF model is a technique based on the idea that users like a particular user can be used to predict how much users may like a movie/item those similar users have streamed except that user. The two main types of CF are user-user CF and item-item CF [5].

Before implementing the above two CF models [10], the MovieLens dataset is split into test and train datasets, and a copy of the train and test dataset is created. These datasets will be used for predicting the movies and evaluating the models [11]. In both cases, the model builds a similarity matrix by using any of the similarity metrics such as Jaccard similarity, Cosine Similarity, and Person similarity [10]. This model utilizes cosine similarity as a similarity measure between users to construct user-user similarity matrix items [3] and likewise, item-item similarity matrix which measures similarity between any two pairs of items [13]. To ignore recommending the same movie that the user already watched, a dummy train matrix is utilized.

To evaluate the model, it evaluates the movie already rated by the user instead of predicting it for the movie not rated by the user. The RMSE and MAE of the CF model are shown in Table-1. For the given dataset, the use of user-based outperformed item-based CF with a minimal RMSE of 1.5 and MAE of 1.2 [9].

Collaborative Filtering Using Singular Value Decomposition

Figure-1 is the basic architecture of the recommendation model using SVD. The input dataset i.e., the MovieLens dataset containing movie data and user data is supplied to the recommendation model [10]. Whenever the user performs certain queries on the recommendation user interface (UI), the recommendation engine performs certain operations, and it provides effective recommendations to the user.

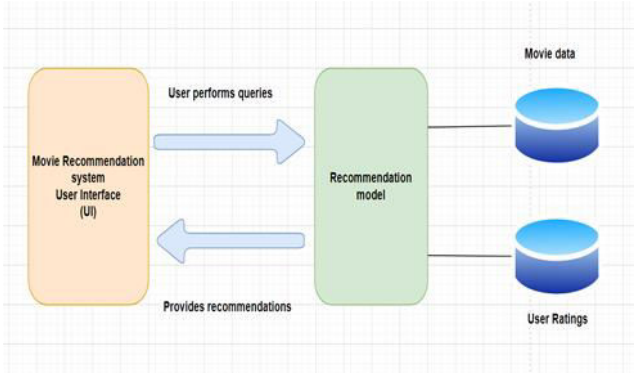


Figure-1. The architecture of the collaborative filtering model using SVD.

The first step in the CF model using SVD [16] is to prepare the user-item matrix in which rows and columns represent users and items respectively, the data fields in it represent the ratings given by users for a particular item. Next, the user-item matrix is decomposed using SVD into matrix U of order (m x k) representing user features, diagonal matrix Σ of order (k x k) representing the strength of latent factors, and matrix V^T of order (k x n) representing item features. The number of latent factors used to describe users and items in this case is k.

According to the formula $r_{ui} = U_u * V_i^T$, the rating of a user u for an item i is determined as the dot product of the respective user and item features. The predicted rating is then used to make recommendations, such as recommending items with the highest predicted ratings for a given user [9]. The matrices U, Σ, and V^T are optimized using gradient descent or other optimization techniques to reduce the variation between predicted ratings and actual ratings in the training data. Regularization is often used to prevent overfitting and improve the generalization performance of the model.

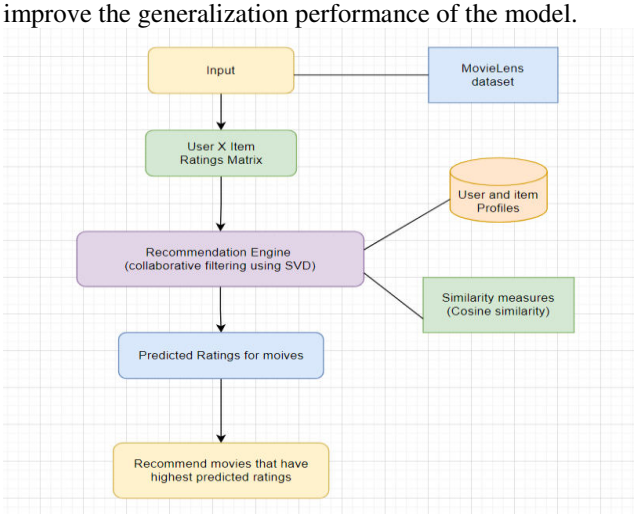


Figure-2. SVD recommendation model flowchart.

Similarity Metrics

To determine the similarity between items and users, generally recommendation systems utilize similarity metrics. The most used similarity metrics [18] for recommendation algorithms are Cosine Similarity, Pearson Correlation similarity, Jaccard similarity, Euclidean Distance, and Manhattan Distance. From these the approaches discussed above use Cosine similarity measures.

Cosine similarity: Items and ratings are taken into consideration as vectors, and the angles between these two vectors are considered while determining similarities [4]. Any two objects can be compared at any angle to see how similar they are. This similarity measure works better with sparse datasets since it generates precise values that range from -1 to +1. The formula for the cosine similarity is given by,

$$\cos(x, y) = x \cdot y / \|x\| * \|y\|$$

Performance Measures

The following are some of the performance metrics that are used in this research work to evaluate the accuracy and performance of the implemented models.

Root Mean Square Error: It calculates the error rate while predicting a non-rated item or movie for an active user [1].

$$RMSE = \sqrt{\Sigma(Pi - Oi)^2 / n}$$

Mean Absolute error: The MAE[10] is a statistic that can be utilized to evaluate how accurate a particular model is. It is determined by:

$MAE = (1/n) * \Sigma|yi - xi$

For the ith observation, Yi is the observed value.

For the ith observation, xi is the predicted value.

N is the total number of observations

Precision: Precision measures the percentage of instances or samples that are accurately classified among those that have been classified as positives.

Precision = TP/(TP + FP)

Where,

True Positives (TP)

True Negatives (TN)

False Positives (FP)

Recall: Recall quantifies the proportion of true positives that were accurately classified [1].

Recall = TP/ (TP + FN)

Where,

False Negatives (FN)

F1 Score: The harmonic mean of recall and precision is the F1 score [1].

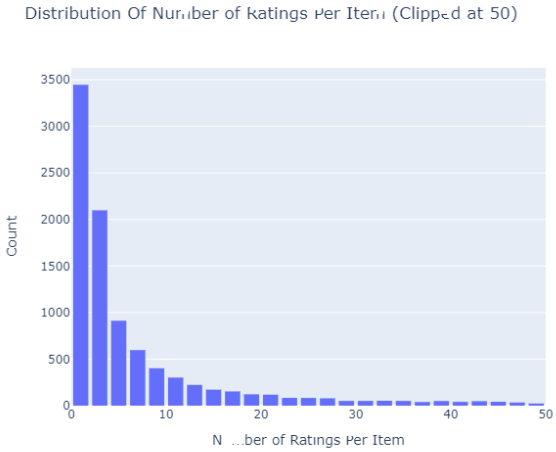
Accuracy = (TP + TN)/ (TP + TN + FP + FN)

F1 score = 2 * (precision * recall) / (precision + recall)

RESULTS AND DISCUSSIONS

Graph-1. Distribution of ratings.

Graph-1 depicts the percentage of users who rated the available movies in the Movie Lens dataset on a scale of 1 to 5 [1]. The graph tells about the users who have rated the movies, here in the Graph the x-axis states the rating that was given by the users, and the y-axis states the count of how many users have given the rating.



Graph-2. Distribution of several ratings per Item.

Graph-2 shows the number of ratings each item in the MovieLens dataset has received [1] [11]. The distribution of the number of ratings for each item received is shown in this graph. The x-axis of this graph shows the number of ratings that were provided for each item. Additionally, the number of ratings that were provided for each item is indicated on the y-axis.

```
genre_recommendations('Toy Story (1995)').head(20)
```

```
1050              Aladdin and the King of Thieves (1996)
2072                          American Tail, An (1986)
2073          American Tail: Fievel Goes West, An (1991)
2285                       Rugrats Movie, The (1998)
2286                            Bug's Life, A (1998)
3045                          Toy Story 2 (1999)
3542                        Saludos Amigos (1943)
3682                           Chicken Run (2000)
3685       Adventures of Rocky and Bullwinkle, The (2000)
236                          Goofy Movie, A (1995)
12                                    Balto (1995)
241                        Gumby: The Movie (1995)
310                       Swan Princess, The (1994)
592                           Pinocchio (1940)
612                         Aristocats, The (1970)
700                         Oliver & Company (1988)
876     Land Before Time III: The Time of the Great Gi...
1010         Winnie the Pooh and the Blustery Day (1968)
1012                  Sword in the Stone, The (1963)
1020                   Fox and the Hound, The (1981)
Name: title, dtype: object
```

Figure-3. Recommended movies by content-based filtering model.

Figure-3 shows the list of recommended movies by the content-based filtering recommendation algorithm using the TF-IDF model and cosine similarity measure [17].

Table-1. Table differentiating RMSE and MAE of user-user and item-item collaborative filtering model.

Model	Root Mean Square Error	Mean Absolute Error
User-User CF	1.56	1.21
Item-Item CF	2.51	2.21

The typical user-based recommendation engine makes 1.2 errors, or (MAE) with a 1.5 RMSE score. An item-based recommendation engine with an RMSE score of 2.51 predicts user ratings with an inaccuracy of 2.21 (MAE). (refer to Table-1) [10].

Table-2. Comparison of average RMSE and execution time.

Algorithm	test-rmse	fit_time	test_time
SVD	0.880258	1.222029	0.568647
NMF	0.934978	1.792422	0.311564
KNNBasic	0.957676	0.112290	5.726139
NormalPredictor	1.425498	0.090737	0.308566

Table-2 shows the RMSE score, fit time, and test time for algorithms like SVD, NMF, KNN, and normal predictors. In comparison with models like SVD, NMF, KNN basic, and normal Predictor, SVD performs better with RMSE of 0.8802, fit time of 1.2202, and test time of 0.5686 [9].

	uid	iid	rui	est	details	Iu	Ui	err
11274	224	969	1.0	4.653160	{'was_impossible': False}	40	23	3.653160
9504	573	8376	0.5	4.180823	{'was_impossible': False}	226	27	3.680823
5326	594	1407	0.5	4.195018	{'was_impossible': False}	167	57	3.695018
21929	598	593	0.5	4.221855	{'was_impossible': False}	15	198	3.721855
23571	594	799	0.5	4.234590	{'was_impossible': False}	167	16	3.734590
10735	543	59900	0.5	4.294476	{'was_impossible': False}	61	10	3.794476
22396	210	296	0.5	4.425568	{'was_impossible': False}	105	224	3.925568
2573	543	35836	0.5	4.480227	{'was_impossible': False}	61	55	3.980227
23245	441	527	0.5	4.868898	{'was_impossible': False}	34	160	4.368898
23005	543	213	0.5	5.000000	{'was_impossible': False}	61	2	4.500000

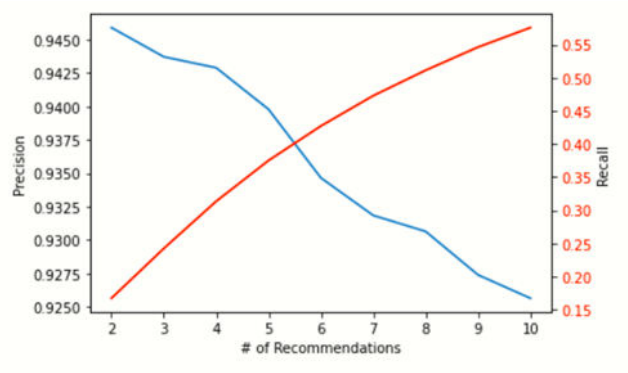
Figure-4. Comparison of the estimated user's rating with the user's actual ratings and determined error.

Figure-4 depicts the estimated ratings (est) of a user with user id (uid) for an item/movie (iid) computed by the CF model using SVD. And by comparing estimated ratings (est) with actual ratings of a particular user error (err) is calculated as shown in Figure-4.

	threshold	tp	fp	tn	fn	Precision	Recall	F1
0	0.0	25209	0	0	0	1.000000	1.000000	1.000000
1	0.5	25209	0	0	0	1.000000	1.000000	1.000000
2	1.0	24867	341	1	0	0.986473	1.000000	0.993190
3	1.5	24133	1056	16	4	0.958077	0.999834	0.978510
4	2.0	23597	1401	107	104	0.943956	0.995612	0.969096
5	2.5	21270	2697	700	542	0.887470	0.975151	0.929247
6	3.0	18174	2451	2352	2232	0.881164	0.890620	0.885867
7	3.5	10587	2644	7191	4787	0.800166	0.688630	0.740220
8	4.0	4052	831	12237	8089	0.829818	0.333745	0.476034
9	4.5	441	183	19535	5050	0.706731	0.080313	0.144235
10	5.0	14	9	21851	3335	0.608696	0.004180	0.008304

Figure-5. Table showing computations of precision, recall and f1 score for different threshold values for CF with SVD.

For each threshold rating on a scale from 0 to 5, the precision, recall, and f1 score for the CF model using SVD are calculated. As shown in Figure-5, the threshold value is 2.5, and since these values [1] are the least for this value, 2.5 is chosen as the threshold value for the rest of the computation.



Graph-3. Visual representation of change precision and recall for different values of k.

Graph-3 is the visual representation of change in precision and recall for different values of k. From Graph-3, it can be observed that when k=4 there is a significant drop in values of precision. So, k=4 is chosen for further computations [1].

	movieId	title	genres
906	1204	Lawrence of Arabia (1962)	Adventure|Drama|War
2453	3266	Man Bites Dog (C'est arrivé près de chez vous)...	Comedy|Crime|Drama|Thriller
2582	3451	Guess Who's Coming to Dinner (1967)	Drama
9618	177593	Three Billboards Outside Ebbing, Missouri (2017)	Crime|Drama

Figure-6. Top four recommendations for users with id 67.

Figure-6 shows the top four recommendations for a random user with a user id of 67. The recommendations [9] are computed with a threshold rating of 2.5 and k as 4. Since precision, recall, and f1 scores are low with threshold = 2.5 and k = 4.

CONCLUSIONS

The above research on various recommendation systems compares different machine learning algorithms for movie recommendations and evaluates their accuracy based on RMSE and MAE. The collaborative filtering approach with SVD is more effective and performs better when compared to other models with a minimal RMSE of 0.880258. The proposed approach undergoes evaluation using additional metrics, including precision, recall, and the f1 score. The recommendation approach is built into a web service that lets users assess movies and then suggests suitable films based on other users' ratings. The study only used the MovieLens dataset for training the models, which limits the generalizability of the findings. Future research could consider using a more diverse dataset that includes a wider range of movies and user preferences. The study only compared a limited number of machine-learning algorithms for movie recommendations. Future research could explore other approaches, such as deep learning models and neural networks to see if they can further improve the accuracy of the recommendations. These are some potential areas for further enhancement.

REFERENCES

[1] Jing Yu., Jinaiing Shi., Yunwen Chen., Daqi Ji., Wenhai Liu., Zhijun Xie., Kai Liu and Xue Feng. 2021. Collaborative Filtering Recommendation with Fluctuations of User' Preference. 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE).

[2] Anusha N., Darmoju Deekshitha., Ghadiyaram Bhavya and Buchupalli Mohitha. 2023. Digital Content Streaming Service with Review, Rating and Parental Control System. 12(1): 725-733.

[3] Mathew. P., Kuriakose B. and Hegde V. 2016. Book Recommendation System through content based and collaborative filtering method, 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, India. 47-52.

[4] Adilaksa Y. and Musdholifah A. 2021. Recommendation System for Elective Courses using Content-based Filtering and Weighted Cosine Similarity, 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia. 51-55.

[5] Breese J. S., Heckerman D. and Kadie C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. 43-52.

[6] Yang Y. 2017. Research and Realization of Internet Public Opinion Analysis Based on Improved TF - IDF Algorithm. 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES).

[7] Adilaksa Y. and Musdholifah A. 2021. Recommendation System for Elective Courses using Content-based Filtering and Weighted Cosine Similarity, 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 51-55.

[8] Bharath S. G. G. M. S. R and Indumathy M. 2021. Course Recommendation System in Social Learning Network (SLN) Using Hybrid Filtering. 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India. 1078-1083.

[9] Bansal S., Gupta C. and Arora A. 2016. User tweets based genre prediction and movie recommendation using LSI and SVD. 2016 Ninth International Conference on Contemporary Computing (IC3).

[10] Chang A. D., Liao J -F. , Chang P -C., Teng C -H and Chen M. -H.. 2014. Application of artificial immune systems combines collaborative filtering in movie recommendation system. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, Taiwan. 277-282.

[11] Maxwell Harper F. and Joseph A. Konstan. 2015. The movieLens datasets: history and context, ACM Transactions on Interactive Intelligent Systems.

[12] Langseth H. and Nielsen T. D. 2012. A latent model for collaborative filtering. International Journal of Approximate Reasoning. 53(4):447-466.

[13] Bobadilla J., Ortega F., Hernando A. and Bernal J. 2012. Generalization of recommender systems: Collaborative filtering extended to groups of users and restricted to groups of items. Expert Systems with Applications. 39(1): b172-186.

[14] Jeong B., Lee J. and Cho H. 2009. An iterative semi-explicit rating method for building collaborative recommender systems, Expert Systems with Applications. 36(3): 6181-6186.

[15] Badrul M., Sarwar, George Karypis., Joseph Konstan and John Reidl. 2022. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT).

[16] Sahoo A. K., Pradhan C. and Prasad Mishra B. S. 2019. SVD based Privacy Preserving Recommendation Model using Optimized Hybrid Item-based Collaborative Filtering, 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 0294-0298.

[17] Rahmah A., Santoso H. B. and Hasibuan Z. A. 2019. Exploring Technology-Enhanced Learning Key Terms using TF-IDF Weighting. 2019 Fourth International Conference on Informatics and Computing (ICIC).

[18] R. C. K. and Srikantaiah K. C. 2021. Similarity Based Collaborative Filtering Model for Movie Recommendation Systems. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India. 1143-1147.