UIJRT ISSN: 2582-6832

United International Journal for Research & Technology

Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

Integrating Randomized Suggestions into Collaborative Filtering Applied to Movie Recommendation

Cayle Sam T. Panganiban[1], Jan Marini E. Punsalan[2], Rusell Kate B. Rodil[3], Vivien A. Agustin[4], Raymund M. Dioses[5], and Jonathan C. Morano[6]

[1,2,3,4,5,6]Computer Science Department, Pamantasan ng Lungsod ng Maynila, Manila, Philippines

***Abstract*—** Collaborative Filtering is a well-known algorithm used for recommendation systems. It predicts users' preferences using historical data, including past interactions, to recommend items they might like. The algorithm looks for similar users and uses this information for possible recommendations. But this existing algorithm still faces three problems: user-cold start, lack of diversity, and popularity bias. This paper introduces a modified version of Collaborative Filtering wherein the Linear Congruent Generator (LCG) is integrated into the algorithm. The LCG was utilized from start to finish of the process. This approach addressed the user-cold start issue by using LCG and generating a set of random users and their favorite top 10 movies. With this method, the user can select their preferred random user from the list based on their top movies and then use the selected user's data to implement Collaborative Filtering. The LCG was also used to address the issue of lack of diversity and popularity bias by generating a list of movies with the highest and least ratings. The addition of LCG to Collaborative Filtering solved the challenges that the original algorithm currently struggles with. The findings were backed up by utilizing evaluation metrics, specifically the Pearson correlation coefficient, Intra-list similarity, and Novelty.

***Keywords*—** collaborative filtering, cold start, diversity, linear congruent generator, pearson correlation coefficient, popularity bias.

I. INTRODUCTION

A. Background of the Study

On the Internet, where the variety of options is overwhelming, users have a specific interest they want to find. Hence, it is necessary to filter and provide user preferences without the need to browse numerous options and to help them avoid spending unnecessary time. This is where the recommendation system solves the problem.

A recommendation system is an application that filters personalized information and understands the users' tastes to suggest relevant things to them by considering their preferences [1]. The recommendation system is beneficial for users to find the content they desire. Several approaches for developing a recommendation system have been developed, including collaborative filtering, content-based filtering, and hybrid filtering [2].

Among the several approaches, Collaborative filtering recommendation is one of the most used algorithms in recommender systems [3]. In CF algorithms, the recommendations for each user are generated using the rating information from other users and items. Collaborative filtering has two types which are user-based CF and item-based CF. User-based collaborative filtering (UB-CF) is based on the ratings given by the target user to an item and then offers recommendations based on the rating given by other similar users to the same item [4]. In comparison, item-based collaborative filtering (IB-CF) focuses on the similarities of items than the similarities of the users [5].

B. Statement of the Problem

The existing collaborative filtering recommender systems are widely used these days, especially now that streaming services are in demand. But despite its functionality, collaborative filtering has several limitations and still suffers from specific problems.

1. *Collaborative Filtering suffers from a cold-start problem*. According to [6], the cold-start problem describes the difficulty of making recommendations when the users are new. For instance, it would be challenging for the system to recommend if a new user has not rated some items. It remains a great challenge for CF. The effectiveness of collaborative filtering dramatically depends on the amount of available information about the user and the item.
2. *Collaborative Filtering lacks diversity in recommendations*. [7] discussed the cause of the lack of recommendation diversity as a consequence of the recommender system's way of learning from too similar items based on a user's preferences. As a result, the algorithm will keep suggesting closely

similar recommendations. Thus, collaborative filtering will suffer from a lack of diversity where no fresh or other items are shown.

3. *Collaborative Filtering has a popularity bias problem.* [8] defined popularity bias as a result of undesired effects from recommender systems wherein the popular items keep being more popular by being frequently recommended. [9] highlight that the suggestion of only popular items prevents the user from discovering new items and disregards a user's niche interests.

C. Objective of the Study

To modify the Collaborative-based filtering algorithm and then apply it to a movie recommendation system. Specifically, this paper seeks to:

1. Remove the cold-start problem by providing new users with random movie suggestions, which the algorithm can use to personalize their recommendations based on other users' interests and existing data.
2. Eliminate the lack of diversity of collaborative filtering by integrating top-rated movie suggestions to help users discover recommendations outside of their preferences.
3. Mitigate the underrepresentation of less popular items, known as popularity bias, by implementing a random generation of low-rated items.

II. RELATED WORKS

A. Recommender System

Recommender systems are systems that use a user's history in terms of user behaviors, personal tastes, thinking styles, and the likes to offer suggestions [10]. Hence, it attempts to identify and construct the most relevant and suited recommendation based on the user's preference.

Collaborative filtering is the most used algorithm in building recommender systems. Collaborative filtering utilizes browsing, rating, and clicks to develop a precise recommendation [11].

B. Problems

The existing user-based collaborative filtering algorithm still faces many problems up until now. These problems include user cold start, lack of diversity, and popularity bias.

One of the problems that the user-based collaborative filtering algorithm faces is the cold start problem. According to [12], the cold start problem could be divided into three categories: user cold start, item cold start, and user-item cold start. User cold start happens when a new user is registered to the system and has no recorded information. An item cold start occurs when a new item is introduced, and other users have not yet rated it. User-item cold start happens when a new user and a new item exists.

Another problem that the collaborative filtering algorithm suffers from is its lack of diversity. In recommender systems, it is known that diversity is vital to ensure that users are entertained by not relying only on items that the users already like. Therefore, it is essential for the recommender system to be diverse to satisfy each of the users' tastes. [13] claimed that one of the primary technical goals of recommender systems is increasing recommendation diversity and making sure that the users are presented with different items.

Popularity bias is also one of the drawbacks of the collaborative filtering algorithm because items are usually not all equally presented in the data in the recommender system; hence some are more popular than others and garner more user behaviors. Because of this, [14] thought that the recommendation system would be more affected by these well-liked items, which would skew the results and make the recommendations biased toward them.

C. Randomization

[15] proposed a new approach that overcomes the problem of user-biased and enhances the computational speed in the recommender system by implementing a randomized algorithm. A recommender system with a randomized algorithm can solve the problem encountered in collaborative filtering and content-based filtering. The author concluded that the implementation of randomization in recommender systems is advantageous in addition to producing the best recommendation to users.

Randomness generation is classified into two categories: True Random Number Generator (TRNG) and Pseudo Random Number Generator (PRNG). [16] elaborated on different pseudorandom number generators, including Linear Congruent Generators. This creates a recurrent sequence of numbers based on a set initial key. In conclusion, linear congruent generators are relevant in fields like computational modeling as a consequence of their execution's speed and simplicity.

Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

III. PROPOSED METHODS

A. Existing Algorithm

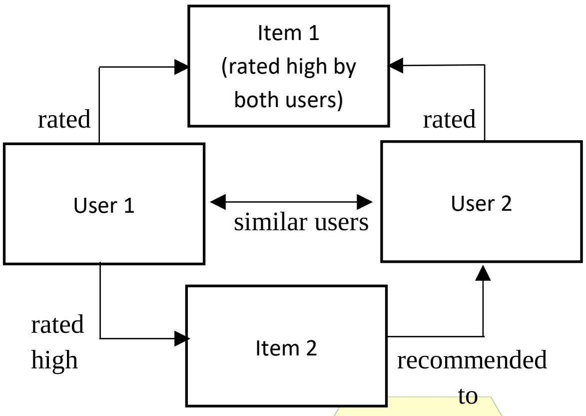

Fig. 1. *Traditional User-Based Collaborative Filtering*

Fig. 1 presents the traditional process of a User-based Collaborative Filtering algorithm. The basic idea of the algorithm: First, Users 1 and 2 rated the same item, item 1, which makes them similar users or neighbors. Then, because they both rated item 1 high, other items that User 1 rated high can be recommended to User 2 [17]. The User-based collaborative filtering algorithm is a unique way to recommend other items that the target user might like. It is based on the ratings given to that item by other users who have a similar taste to the target user [18].

B. Proposed Modification of the Algorithm

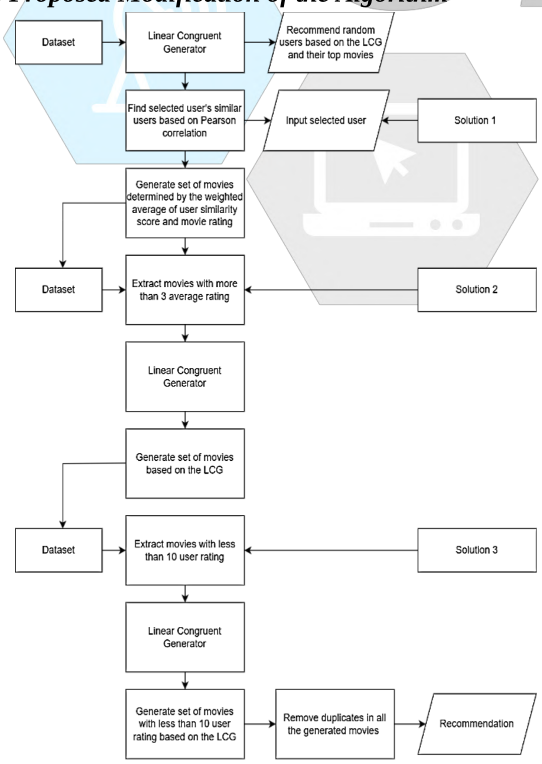

Fig. 2. *Proposed Modification of Collaborative Filtering*

Fig. 2 shows the modified Collaborative Filtering. Solution one is used to address the cold start issue. The process will start in the linear congruent generator. It will be used to recommend random users from the dataset and display their user ID and their top movies. Based on the shown movies, the user can now input their preferred user. After that, it will use the PCC to find a similar user to their selected user. Then, it will use traditional collaborative filtering to generate a recommendation.

Solution two is utilized to fulfill the need for more diversity. The proponents will use the linear congruent generator to extract movies with more than three average ratings. The recommended movies are a set of random top movies. Hence, integrating top-rated movie recommendations can help users discover more suggestions outside their preferences.

Solution three is used to address the popularity bias. The proponents will extract movies with less than 10 user ratings. It will also use the linear congruent generator to recommend movies with less than 10 user ratings. Therefore, recommending a low-rated item can help mitigate the underrepresentation of less popular items.

B.1. Pearson Coefficient Matrix

Pearson's recommendation is calculated using the weighted average of user similarity score and movie ratings. To elucidate, two matrices are used to come up with a list of recommendations, namely the user-item and user-user similarity matrix.

Table I: *Sample Pearson Correlation Coefficient Matrix for User-Item Similarity*

	I 1	I 2	I 3	I 4	I 5	…
U 1	4.0	0.0	4.0	0.0	0.0	…
U 2	0.0	0.0	0.0	0.0	0.0	…
U 3	0.0	0.0	0.0	0.0	0.0	…
U 4	0.0	0.0	0.0	0.0	0.0	…
U 5	4.0	0.0	0.0	0.0	0.0	…
…	…	…	…	…	…	…

Table I illustrates the user-item similarity found in the dataset. Specifically, it shows the movie preference of all 600 users based on movies they have rated. This is calculated by using their historical data of movie ratings from 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, and 5.00 as the highest. Therefore, producing a matrix with the calculated value of 1.00 - 5.00 in each cell. Consequently, the value 0.0 represents that there is no available user-item data.

The above similarity matrix will be used to find a selected user and their ten most similar users from the perspective of movies they rated. As a result, the significant values from user-item similarity will be used to come up with a user-user matrix.

Table II: *Sample Pearson Correlation Coefficient Matrix for Similar Users*

	U 1	U 2	U 3	U 4	U 5	…
U 1	1.000000	0.0	0.0	0.391797	0.180151	…
U 2	0.0	1.0	0.0	0.0	0.0	…
U 3	0.0	0.0	0.0	0.0	0.0	…
U 4	0.391797	0.0	0.0	1.000000	-0.394823	…
U 5	0.180151	0.0	0.0	-0.394823	1.000000	…
…	…	…	…	…	…	…

The table above shows the Pearson correlation coefficient matrix used to identify the degree of user similarity in the dataset. The values range from 1.000000 as the highest, which is a positive Pearson correlation. 0 represents the users without any similar movie ratings, while -1.000000 shows a value for a negative Pearson correlation.

The aforementioned similarity matrix will then be used to calculate and generate a list of 10 users with the most significant value of similarity, the ideal of which is 1.00.

B.2. Linear Congruential Generator

The proponents proposed a modified approach to collaborative filtering to eliminate the cold start problem, lack of diversity, and popularity bias. The proponents will integrate the Linear Congruent Generator (LCG) method into the whole process to generate a random recurrent sequence according to the formula that [16] discussed.

$$X_n = (a \cdot X_{n-1} + b) \mod m \qquad (Eq. 1)$$

where:

• Xn is currently generated element of sequence.

• $Xn-1$ is previous element in the sequence.

Additionally, the parameters,

$$a, b, m(0 < a < m, 0 \leq b < m, m > 0$$

are fixed during the entire process of producing the pseudorandom numbers.

Assume that the random numbers that will be generated are set into three. It will generate three random numbers using the LCG. Then, it will get the three user IDs from the dataset and display their chosen movies. The proponents show a visualization of the application of a Linear Congruent Generator to the traditional collaborative filtering applied in a movie recommendation and how the proponents will use it to eliminate the cold start problem (See Fig. 3).

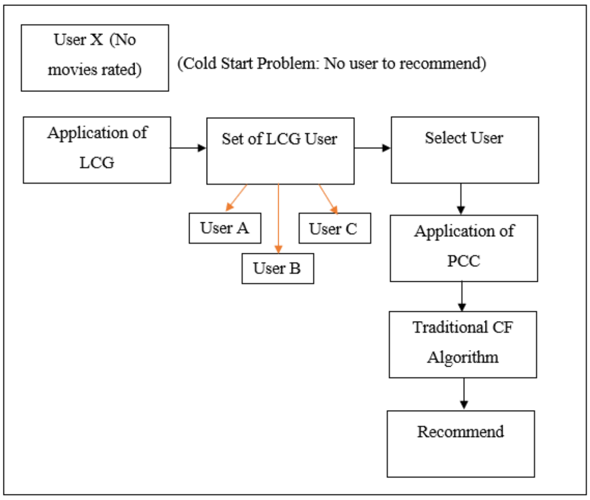

Fig. 3. *Application of LCG*

As seen in Fig. 3, assume that there is a new user, User X, who has no rated items. Therefore, traditional Collaborative Filtering cannot recommend items for the new user. Using the Linear Congruent Generator, it can generate three random users through the datasets, and the new user can now choose from the collection of LCG users. The Pearson Correlation Coefficient will be used to search for similar users after the user selection. It may now run the standard Collaborative Filtering algorithm and recommend a new user to the system. The LCG method will also be applied to eliminate the lack of diversity by randomizing the displayed movies with more than three average ratings. Furthermore, LCG will also be used to randomize movies with less than ten user ratings in the dataset to solve the popularity bias problem.

C. Methodology

The researchers' proposed modifications for the traditional collaborative filtering method by integrating the Linear Congruent Generator will be applied in a movie recommendation. Thus, providing a user with a variety of options outside their interests.

Evaluation metrics will then be implemented on the modified collaborative filtering method to assess whether the results show a relevant recommendation in the context of similarity, diversity, and bias. The datasets utilized in this research are from the September 2018 latest update of the MovieLens Datasets.

United International Journal for Research & Technology
Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

C.1. Evaluation Metrics

C.1.1. Similarity Matrix: Pearson Correlation Coefficient

According to [19], Pearson Correlation Coefficient (PCC) is one of the most popular similarity measures for Collaborative filtering recommender systems. It is used to evaluate how much two users are correlated. The PCC formula is shown below.

$$sim(u,v) = \frac{\sum_{i\in I_{u,v}} \left(R_{u,i} - \overline{R_u}\right)\left(R_{v,i} - \overline{R_v}\right)}{\sqrt{\sum_{i\in I_{u,v}} \left(R_{u,i} - \overline{R_u}\right)^2} \cdot \sqrt{\sum_{i\in I_{u,v}} \left(R_{v,i} - \overline{R_v}\right)^2}} \qquad (Eq. 2)$$

The similarity value computed using the above equation falls within the range of [-1, 1]. The larger the similarity value represents, the more similar the two users are [20]. This similarity measure will be used as an evaluation metric to determine if the proposed modification for collaborative filtering solves the cold start problem.

C.1.2. Intra-list Similarity

[21] described intra-list similarity as the average cosine similarity of all items in a given list. Typically, the similarity of items is established using metadata specific to the domain, such as movie genres. When a recommender system suggests lists of items that are very similar to individual users, the result is that the intra-list similarity will be high.

The formula for computing the cosine similarity is:

$$\cos(x,y) = \frac{x \, . \, y}{||x|| \, * \, ||y||} \qquad (Eq. 3)$$

To compute the intra list similarity, here is the formula:

$$intra = \left(\frac{1}{n}\right) * \sum(i = 1 \; to \; n) \sum(j = i + 1 \; to \; n) \; cos(x_i, x_j) \qquad (Eq. 4)$$

A low value is commonly favored when using intra-list similarity as a diversity metric in recommender systems because it implies that the items in the recommended list are diverse and dissimilar. On the contrary, a high intra-list similarity suggests that the items in the recommended list are similar, and it may not solve the diversity problem the current recommender systems are still facing.

C.1.3. Novelty

According to [22], there are numerous concepts for evaluating recommendations that different researchers have considered, and one of them is a novelty. The concept of novelty often refers to including novel items in the recommendation. This novelty metric will be used to evaluate the popularity bias. The equation below shows the novelty metric.

$$novelty = 1 - \frac{\log_{10}(number \; of \; ratings)}{\log_{10}(maximum \; number \; of \; ratings)} \qquad (Eq. 5)$$

This formula shows that items with fewer ratings are more novel than items with more excellent ratings since they are less well-known or popular. The formula uses the base-10 logarithm to calculate the novelty score. The novelty score ranges from 0 to 1, with 1 representing the novelest item.

IV. RESULTS AND DISCUSSION

A. Evaluation Metrics

A.1. Pearson Correlation Coefficient Matrix

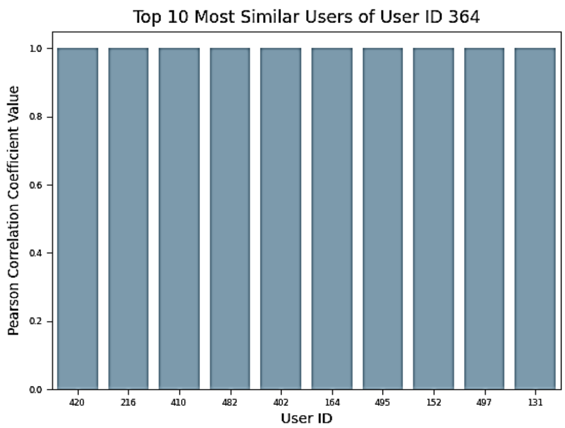

Fig. 4. *The Top 10 Most Similar Users of User ID 364*

The bar graph illustrated in Fig. 4 shows the result of the 10 most similar users for a chosen user ID of 364. The diagram outputs the Pearson correlation coefficient matrix based on user-item and user-user matrices. To elaborate on the graph details, proceed to Table III.

Table III: *Degree of Pearson Correlation Coefficient Values of 10 Most Similar Users*

User ID	Similarity Score	Correlation
420	1.00	Perfect
216	1.00	Perfect
410	1.00	Perfect
482	1.00	Perfect
402	1.00	Perfect
164	1.00	Perfect
495	1.00	Perfect
152	1.00	Perfect
497	1.00	Perfect
131	1.00	Perfect

UIJRT ISSN: 2582-6832

United International Journal for Research & Technology

Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

The table presents the top 10 most similar users for user 364. The list was utilized in generating the top movie recommendations alike to user 364's interest. As seen on the table above, the Pearson correlation values are all 1.00 which represents a perfect correlation. According to [23], there are degrees of correlation in the Pearson similarity values which are interpreted as:

$$r_{xy} = 1.00 \textit{ is a Perfect Positive Correlation}$$

$$0.8 < r_{xy} < 1.00 \textit{ is a Strong Positive Correlation}$$

$$0.3 < r_{xy} < 0.6 \textit{ is a Moderate Positive Correlation}$$

$$0 < r_{xy} < 0.3 \textit{ is a Weak Positive Correlation}$$

$$0 < r_{xy} < -0.3 \textit{ is a Weak Negative Correlation}$$

$$-0.3 < r_{xy} < -0.6 \textit{ is a Moderate Negative Correlation}$$

$$-0.8 < r_{xy} < -1.00 \textit{ is a Strong Negative Correlation}$$

$$r_{xy} = -1.00 \textit{ is a Perfect Negative Correlation}$$

where r represents the Pearson correlation coefficient and the subscript is the two vectors, x and y.

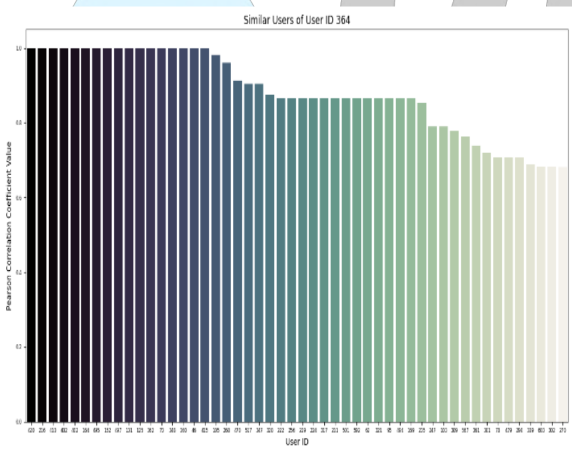

Fig. 5. *The 50 Most Similar Users of User ID 364*

The figure above displays all 50 similar users in terms of user 364. The graph was generated to ensure that the Top 10 Most Similar Users of User 364 are accurate with the highest scores. Also, the bar graph illustrates the level of similarity based on its color—with dark colors as the most similar users and the lighter ones are the least similar.

In conclusion, Fig. 5 reveals the calculated similarity of 50 users in relation to users 364 historical data of rated movies. The movies are arranged in descending order, thus showing that the top 10 most similar users in Table III have the highest degree of similarity compared to the remaining users. Hence, proving the precision of movie recommendations.

A.2. Intra-list Similarity

The proponents used the intra-list similarity to evaluate the lack of diversity in the recommender system. To compute intra-list similarity, the proponents first compute the cosine similarity matrix between all movie genres of all movies, which measures the similarity between all pairs of movies based on their genre.

The proponents created the binary feature matrix with a numerical representation that calculates the cosine similarity.

The reason for using a binary feature matrix to calculate cosine similarity is that the genres of the movies are categorical variables and cannot be directly used in a cosine similarity calculation.

Therefore, the genres are transformed into a binary feature matrix that is shown in Table IV Each row represents a movie, and each column represents a genre. If a movie belongs to a specific genre, the corresponding entry in the matrix is set to 1. Otherwise, it is set to 0.

Table IV: *Binary Feature Matrix*

	Action	Adventure	Animation	Children	…
0	0	1	1	1	…
1	0	1	0	1	…
2	0	0	0	0	…
3	0	0	0	0	…
4	0	0	0	0	…
…	…	…	…	…	…

The cosine similarity between two movies is high if they have many common genres and low if they have few common genres.

Once you have the cosine similarity matrix that computes the genre similarity, you can calculate the intra-list similarity for a set of recommended movies.

The intra-list similarity is then used to evaluate the diversity of a set of recommended movies based on their genre similarity.

The average pairwise cosine similarity between the recommended movie genres is used to calculate it.

This metric shows how diverse the recommended movies are to each other based on their genre similarity.

Table V: *Intra-list Similarity of the Recommended Movies*

Recommended High-rated Movies	Genres
Public Enemies (2009)	Crime | Drama | Thriller
The Never-Ending Story (1984)	Adventure | Children | Fantasy
Body Heat (1981)	Crime | Thriller
Videodrome (1983)	Fantasy | Horror | Sci-Fi | Thriller
Insomnia (1997)	Drama | Mystery | Thriller
Jetée, La (1962)	Romance | Sci-Fi
Razor's Edge, The (1984)	Drama
Die Hard 2 (1990)	Action | Adventure | Thriller
24 Hour Party People (2002)	Comedy | Drama | Musical
Nick of Time (1995)	Action | Thriller
Intra-list Similarity:	0.2924

According to [24], the score is standardized to lie in [−1, 1] or [0, 1].

When utilizing intra-list similarity as a diversity metric in recommender systems, a low value is typically preferred because it signifies that the items in the recommended list are diverse and dissimilar to one another.

A high intra-list similarity, on the other hand, shows that the items on the recommended list are similar.

Table V shows the recommended high-rated movies with their genres. The intra-list similarities score is 0.2924.

The result indicates that the recommended movies are diverse and dissimilar. Therefore, the modified collaborative filtering solves the lack of diversity.

A.3. Novelty

The proponents used Novelty as an evaluation metric to measure how many times the users have rated a movie in the past and to calculate its novelty score based on how frequently it was rated.

It is used to ensure that the recommended movies are unique and new to the users while combating the popularity bias problem.

Table VI: *Novelty Scores of Recommended Movies*

Recommended Least-rated Movies	Number of Ratings	Novelty Score
Tanguy (2001)	1	1.0
White Ribbon, The (Das weiße Band) (2009)	1	1.0
Journey to the Center of the Earth (1959)	2	0.88035
Holywood Ending (2002)	1	1.0
3 dev adam (Three Giant Men) (1973)	1	1.0
8 Women (2002)	3	0.81036
Mr. Blandings Builds His Dream House (1948)	1	1.0
Kiss Me Kate (1953)	1	1.0
Mountains of the Moon (1990)	1	1.0
Toy Soldiers (1991)	4	0.76070

Table VI demonstrates the recommended least-rated movies' novelty scores. In computing novelty scores, the score ranges from 0 to 1. A novelty score of 0 means that the movies are not very unique, as they have been rated many times. Specifically, in this study, a value of 0 means that it has been rated 328 times. However, a novelty score of 1 signifies that the movies are unique, and many users have not rated them. Hence, the result of the least-rated recommended movies was proven to be accurate and high in novelty, as it can be seen in the table that they were mainly composed of 1.

V. CONLUSION AND RECOMMENDATION

A. Conclusion

The proponents have modified the traditional user-based collaborative filtering by integrating Linear Congruent Generator. The existing collaborative filtering algorithm suffers from the user cold start problem, lack of diversity, and popularity bias. Three evaluation metrics, Pearson Correlation Coefficient, Intra-list Similarity, and Novelty, were applied to them to evaluate the accuracy and precision of generated recommendation results.

The research showed that the modified user-based collaborative filtering method could produce an accurate movie recommendation with a PCC value of 1.00. The algorithm also generated a diverse range of movies in terms of popularity. The diversity score of the movies averaged an intra-list similarity score of 0.2924 which represents how different the movies are in terms of the similarity of each movie genre. Finally, the popularity bias was resolved by suggesting the least-rated movies. The movie recommendations were evaluated using a novelty metric. The suggested movies only have 1 to 6 user ratings, which is within the range of the study's threshold of less than ten user ratings.

B. Recommendation

For the future development of this algorithm, the researchers recommend that the dataset should have more users with ratings of the current movies relevant today. The proponents noticed that most of the users in the dataset leaned toward older movies. The proponents also suggest that future researchers explore integrating randomized solutions in collaborative filtering, as currently, limited studies are available. Furthermore, it is recommended that this research be further implemented on existing users because the study targeted the problems that arise from being a new user. Lastly, applying the study to different forms of entertainment and real-life data is suggested where a randomized suggestion can wield its full potential.

ACKNOWLEDGMENT

This research would not have been accomplished if it were not for the unrelenting support, advice, and encouragement of our loved ones and respected advisers.

The research proponents extend their utmost gratitude to Computer Science professors, namely Prof. Ariel Sison, for the advice and recommendation towards the research topic approval. Also, Mr. Raymund Dioses and Mr. Jonathan Morano are appreciated for their constructive criticisms as research panelists. Lastly, we thank Prof. Vivien Agustin for the thesis writing guidance from the beginning until the completion of this research study.

Furthermore, we would like to thank our wonderful parents for being generous and understanding of our needed time and patience. We also acknowledge our peers for extending their help.

Finally, we, the research proponents, thank God for His grace in providing us with the intellectual, emotional, and financial capability needed to complete this study.

REFERENCES

[1] D. Das, L. Sahoo, and S. Datta, “A survey on recommendation system,” International Journal of Computer Applications, vol. 160, no. 7, pp. 6–10, Feb. 2017. doi:10.5120/ijca2017913081

[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, Jun. 2015. doi:10.1016/j.eij.2015.06.005

[3] G. Xu, Z. Tang, C. Ma, Y. Liu, and M. Daneshmand, “A collaborative filtering recommendation algorithm based on user confidence and time context,” Journal of Electrical and Computer Engineering, vol. 2019, pp. 1–12, Jun. 2019. doi:10.1155/2019/7070487

[4] I. Dwicahya, P. H. Rosa, and R. Nugroho, “Movie recommender system comparison of user-based and item-based collaborative filtering systems,” Proceedings of the International Conference of Science and Technology for the Internet of Things, 2018. doi:10.4108/eai.19-10-2018.2282541

[5] P. Boström and M. Filipsson, “Comparison of user based and item based collaborative filtering recommendation services,” Digitala Vetenskapliga Arkivet, http://diva-portal.org/smash/get/diva2:1111865/FULLTEXT01.pdf (accessed April 13, 2023).

[6] X. Zhao, “Cold-start collaborative filtering,” ACM SIGIR Forum, vol. 50, no. 1, pp. 99–100, Jun. 2016. doi:10.1145/2964797.2964819

[7] E. Isufi, M. Pocchiari, and A. Hanjalic, “Accuracy-diversity trade-off in recommender systems via graph convolutions,” Information Processing & Management, vol. 58, no. 2, Mar. 2021. doi:10.1016/j.ipm.2020.102459

[8] M. Elahi et al., “Investigating the impact of recommender systems on user-based and item-based popularity bias,” Information Processing & Management, vol. 58, no. 5, Sep. 2021. doi:10.1016/j.ipm.2021.102655

[9] H. Abdollahpouri, R. Burke, and B. Mobasher, “Managing popularity bias in recommender systems with personalized re-ranking,” arXiv.org, https://arxiv.org/abs/1901.07555 (accessed Apr. 13, 2023).

[10] T. Mohammadpour, A. M. Bidgoli, R. Enayatifar, and H. H. S. Javadi, “Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm,”

Genomics, Jan. 2019,doi: https://doi.org/10.1016/j.ygeno.2019.01.001.

[11] M. F. Aljunid and M. Dh, "An Efficient Deep Learning Approach for Collaborative Filtering Recommender System," Procedia Computer Science, vol. 171, pp. 829–836, 2020, doi: https://doi.org/10.1016/j.procs.2020.04.090.

[12] S. Motadoo, "Resolving Cold Start Problem Using User Demographics and Machine Learning Techniques for Movie Recommender Systems," doi: https://doi.org/10.31979/etd.548a-yyn2.

[13] Hanafi, N. Suryana, A. Samad Bin, and H. Basari, "AN UNDERSTANDING AND APPROACH SOLUTION FOR COLD START PROBLEM ASSOCIATED WITH RECOMMENDER SYSTEM: A LITERATURE REVIEW," vol. 96, p. 9, 2018, Accessed: May 13, 2023. [Online]. Available: http://www.jatit.org/volumes/Vol96No9/26Vol96No9.pdf

[14] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He†, "Bias and Debias in Recommender System: A Survey and Future Directions," ACM Transactions on Information Systems, Oct. 2022, doi: https://doi.org/10.1145/3564284

[15] Shakil et al., "The Impact of Randomized Algorithm over Recommender System," Procedia Computer Science, vol. 194, pp. 218–223, 2021, doi: https://doi.org/10.1016/j.procs.2021.10.076.

[16] N. Antonov, "Random number generator based on multiplicative convolution transform," 2020. Accessed: May 13, 2023. [Online]. Available: https://dspace.cvut.cz/bitstream/handle/10467/88140/F3-DP-2020-Antonov-Nikolai-Thesis.pdf?sequence=-1&isAllowed=y

[17] A. J. Nadal, A. Reyes, M. C. Blanco, A. Alipio, and D. M. Cortez, "Enhancement of collaborative filtering using Myers-Briggs Type Indicator (MBTI) applied in recommendation system," South Asian Journal of Engineering and Technology, vol. 12, no. 1, Mar. 2022. doi:10.26524/sajet.2022.12.016

[18] H. Wang, Z. Shen, S. Jiang, G. Sun, and R.-J. Zhang, "User-based collaborative filtering algorithm design and implementation," Journal of Physics: Conference Series, vol. 1757, no. 1, 2021. doi:10.1088/1742-6596/1757/1/012168

[19] L. Sheugh and S. H. Alizadeh, "A note on pearson correlation coefficient as a metric of similarity in recommender system," 2015 AI & Robotics (IRANOPEN), Apr. 2015, doi: https://doi.org/10.1109/rios.2015.7270736.

[20] K. G. Saranya, G. Sudha Sadasivam, and M. Chandralekha, "Performance Comparison of Different Similarity Measures for Collaborative Filtering Technique," Indian Journal of Science and Technology, vol. 9, no. 29, Aug. 2016, doi: https://doi.org/10.17485/ijst/2016/v9i29/91060.

[21] R. Sargar, "Recommender System using Reinforcement Learning," 2020. Available: https://keep.lib.asu.edu/_flysystem/fedora/c7/224647/Sargar_asu_0010N_19940.pdf

[22] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," International Journal of Machine Learning and Cybernetics, vol. 10, no. 5, pp. 813–831, Dec. 2017, doi: https://doi.org/10.1007/s13042-017-0762-9.

[23] V. A. Profillidis and G. N. Botzoris, "Statistical Methods for Transport Demand Modeling," Modeling of Transport Demand, pp. 163–224, 2019, doi: https://doi.org/10.1016/b978-0-12-811513-8.00005-4.

[24] M. Jesse, C. Bauer, and D. Jannach, "Intra-list similarity and human diversity perceptions of recommendations: the details matter," User Modeling and User-Adapted Interaction, Dec. 2022, doi: https://doi.org/10.1007/s11257-022-09351-w.