A Scalable Collaborative Filtering Recommendation Model for Prediction of Movie Rating

C. Ugwu[1] and Ogundare, oluwagbenga emmanuel[2]

[1&2] *Department of Computer Science*
University of Port Harcourt. Nigeria.

Abstract *The persistent overwhelming effect on e-commerce users which is as a result of the availability of vast array of product choices demands new techniques of computational intelligence that have the potential of being flexible and producing a better predictive accuracy. The decoupling normalization technique was deployed to correctly represent the true level of user's interest for several movies in an e-commerce domain. The system used collaborative filtering technique with a hybrid of locality sensitive hashing algorithm and singular value decomposition approach to build the model. Different representative cases of movie ratings were examined from the Movie Lens ratings dataset to validate the model. The system was designed with Object-Oriented Analysis and Design (OO-AD) method and implemented with C-sharp programming language. The results achieved were evaluated with the Mean Average Error (MAE) and Root Square Mean Error (RSME) analysis metrics and the system was found to predict at an accuracy of 90.8%.*

Keywords - *collaborative filtering (cf), locality sensitive hashing, singular value decomposition, recommender system.*

I. INTRODUCTION

Consumers on various E-commerce sites enjoy the easy accessibility to millions of products made available by several online retailers. These consumers however, are inundated with a vast array of choices due to the exponential growth of online products because online retailers offer their products without putting into consideration how e-customers will get products that suit their preferences and match their interest [16]. This brings into play what is called a Recommender system; which is a software agent saddled with the responsibility of mentioning to e-customers products that suits their preferences and match their interest. Recommenders systems have been widely applied in many e-commerce sites for recommending different items such as books [9], music [11], movies [15] etc. for several users. Example of such e-commerce sites include Amazon.com, Netflix.com, Moviefinder.com, e-bay.com etc. [7]. In the e-commerce domain, the process of matching consumers with products that satisfy their need is not a trivial subject, it is a key to ensuring user's happiness and loyalty [1]

With the popularity of the internet and advances in information technology, information from websites tends to be too general and people require more personalized information. In order to meet users' demand for personalized services, personalized recommender systems are veritable tool for providing such services as they help to streamline their several possible choices of items to the most suitable items that suit their preferences [18]. In describing the use and the evaluation of collaborative filtering recommender systems, two tasks are required. The "predict" task: in a typical e-commerce domain where we have a user and an item, what is the probable rating that the user will give to the item? Secondly, the "recommend" task: what are the top ranked lists of items that will match the user's interest? [5].The available collaborative filtering models consume much time in determining the optimal solution. There is also the sparsity problem faced in the recommendation model which requires improvements.

In this paper a hybrid model with collaborative filtering techniques was designed and implemented to deal with time and sparsity problems as identified in the existing systems. The decoupling normalization model was used to represent the true preference level of the users, while the locality sensitive hashing algorithm was used to cluster similar users to improve to optimize the match time.

II. RELATED WORKS

A combination of personal information filtering agents and opinions from the community of users have been employed to produce a better recommendation systems. It was discovered that this combination outperforms the use of either agents or user's opinion alone [6].

[10], presented "Incremental collaborative filtering recommender based on regularized matrix factorization" Experiments were carried out on the Movie Lens and Netflix datasets and they compared the performance of models RMF and SI-RMF. Experimental results show that the performance of SI-RMF is more encouraging than RMF. This performance depends on the value of datasets used.

[3] borrowed the concepts of object typicality from cognitive psychology and proposed a novel typicality-based CF recommendation method called TyCo. TyCo was known to search for neighbors of users using user typicality degrees in groups as opposed to the co-rated items of users or common users of items as it is being practised in traditional CF. The experimental results showed an accuracy improvement of at least 6.35% in MovieLens dataset with a reduced time cost.

In a bid to resolve the two major problems that most Collaborative filtering approaches encounters, which are scalability and sparseness of the user profile, [19] presented a parallel algorithm designed for the Netflix Prize called Alternating-Least-Squares (ALS) with Weighted Regularization (ALS-WR). The experiment was carried out with a parallel Matlab on a Linux cluster to show empirically that the performance of ALS-WR monotonically increases with both the number of features and the number of ALS iteration when applied to the Netflix dataset with 1000 hidden features to obtain a RMSE score of 0.8985.

[17] examined Singular value decomposition (SVD) hybridized with demographic information to enhance the plain collaborative filtering algorithm and improve the quality of generated predictions. This approach was tested on user-based collaborative filtering and item-based collaborative filtering to prove its efficiency. The results showed a promising recommender systems which increased the level of accuracy of systems in which it was deployed.

Several other techniques have been successfully applied to collaborative filtering, but it was shown by many researchers' shows that combination of various algorithms typically outperforms single methods in terms of accuracy [6].

This paper presented a different approach that considered user rating habit as opposed to existing approaches discussed. This was used to clearly reflect users' true affinity level which improved the training process of the system. We also generated a compact representation of data object which were represented in compact sketches of similar users to achieve a scalable solution for the class of collaborative filtering problem. Scalability is crucial since collaborative filtering systems often have to manage millions of users or items.

III. MATERIALS AND METHODS

The fig 1 shows an extension of the SVD decomposition architecture adopted from Zhou et al., 2014. The decoupling normalization component and the locality sensitive hashing methods were incoperated to capture the true affinity level of users for several items in the ratings data. It was used exactly to reflect different user's affinity level which showed the wide and narrow rating habit of both tolerant and strict users respectively. In the same vein, the locality sensitive hashing method was deployed to construct functions that hashes objects into buckets such that objects that have close similarity were hashed in the same bucket with high probability. This maps similar data of arbitrary size to data of fixed size in form of a bucket and this compact representation of objects allowed estimations to be made from the compact sketches.

The dataset used was obtained from movieLen website which contained 100,000 ratings applied to 1682 movies by 943 users released on April, 2015. The data was normalized and were represented in real numbers ranging between 0.1 and 1

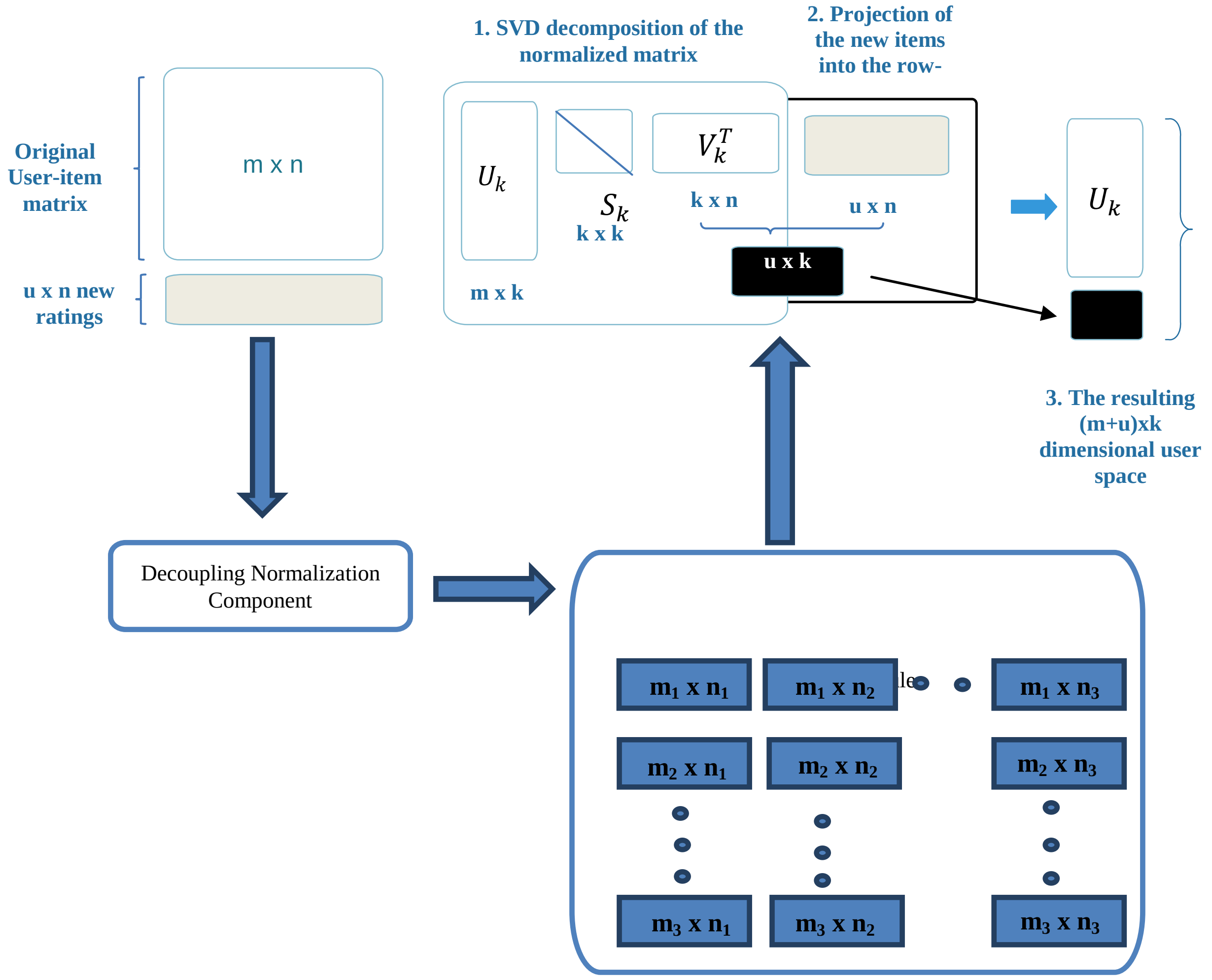

1. SVD decomposition of the normalized matrix
2. Projection of the new items into the row-
Original User-item matrix
m x n
u x n new ratings
U_k
S_k
k x k
V_k^T
k x n
u x n
u x k
m x k
U_k
3. The resulting (m+u)xk dimensional user space
Decoupling Normalization Component
m1 x n1
m1 x n2
m1 x n3
m2 x n1
m2 x n2
m2 x n3
m3 x n1
m3 x n2
m3 x n3

Fig 1: Architecture of the system.

The decoupling normalization was achieved with the following equation

$$\tilde{R} = P_i(R\ is\ preferred) = P_i(Rating \le P) - \frac{P_i(Rating = R)}{2} \ldots\ldots\ldots\ldots\ldots(1)$$

Where P_i (*R* is preferred) = true representation of how items with the ratings R is liked by a user and Pi (*Rating*<= *R*) and P_i (*Rating*=*R*) = the probability which implements how the user likes the item with the rating *R*.

Table 2 shows the decoupling normalization approach adopted to determine the user ratings given 3 users, U1, U2, U3 who have rated 5 items, I1, I2, I3, I4, I5 and their ratings are represented in table 1:

Table 1: Ratings before Normalization

	I1	I2	I3	I4	I5
U1	1	2	3	4	5
U2	3	3	4	5	5
U3	2	2	3	3	4

From the rating information in table 1, we can deduce the disparity that exists between the U1, U2 and U3 was as a result of their varying rating habits. User 1 tends to rate items using a wide range while User 2 and User 3 make use of a more narrow scale to represent their affinity level for the items. Therefore, user 1 seems to be a more “tolerant user” than user 2 who is also more tolerant than user 3. However, table 3 shows the normalized ratings which represented their users’ preference for the items.

Table 2: Determined Normalized Results

	Rating(R)	1	2	3	4	5
U1	PP_i (*Rating*=*R*)	0.2	0.2	0.2	0.2	0.2
	P_i (*Rating*<=*R*)	0.2	0.4	0.6	0.8	1.0
	$P_i(Rating \le R) - \frac{P_i(Rating = R)}{2}$	0.1	0.3	0.5	0.7	0.9
U2	P_i (*Rating*=*R*)			0.4	0.2	0.4
	P_i (*Rating*<=*R*)			0.4	0.6	1.0
	$P_i(Rating \le R) - \frac{P_i(Rating = R)}{2}$			0.2	0.5	0.8
U3	P_i (*Rating*=*R*)		0.4	0.4	0.2	
	P_i (*Rating*<=*R*)		0.4	0.8	1.0	
	$P_i(Rating \le R) - \frac{P_i(Rating = R)}{2}$		0.2	0.6	0.9	

P_i (*Rating*=*R*) = $\frac{\text{probability of number of each rating}}{\text{total number of ratings}}$

P_i (*Rating*<=*R*) is the cumulative sum of the P_i (*Rating*=*R*) beginning with the lowest level of preference to the highest level of preference.

The normalized results represented in table 3 was computed with the equation of decoupling normalization.

Table 3: Ratings Results after Normalization

	I1	I2	I3	I4	I5
U1	0.1	0.3	0.5	0.7	0.9
U2	0.2	0.2	0.5	0.8	0.8
U3	0.2	0.2	0.6	0.6	0.9

For the similarity measure the Cosine Similarity was used as its proximity computing approach [1] as shown in the equation 1 and further explained with table 1.

$$sim(\vec{a},\vec{b}) = \frac{\sum_{u_a \epsilon U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u_a \epsilon U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u_b \epsilon U}(r_{u,b} - \overline{r_u})^2}} \dots \dots \dots$$

Table 4: Rated data item

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7
Use r 1	4			5	1		
Use r 2	7	5	4				
Use r 3				2	4	5	
Use r 4		3					3

From table 4 the cosine distance between User 1 and User 2 is

$$\frac{4 * 5}{\sqrt{4^2 + 5^2 + 1^2}\sqrt{7^2 + 5^2 + 4^2}} = 0.325 \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . (3)$$

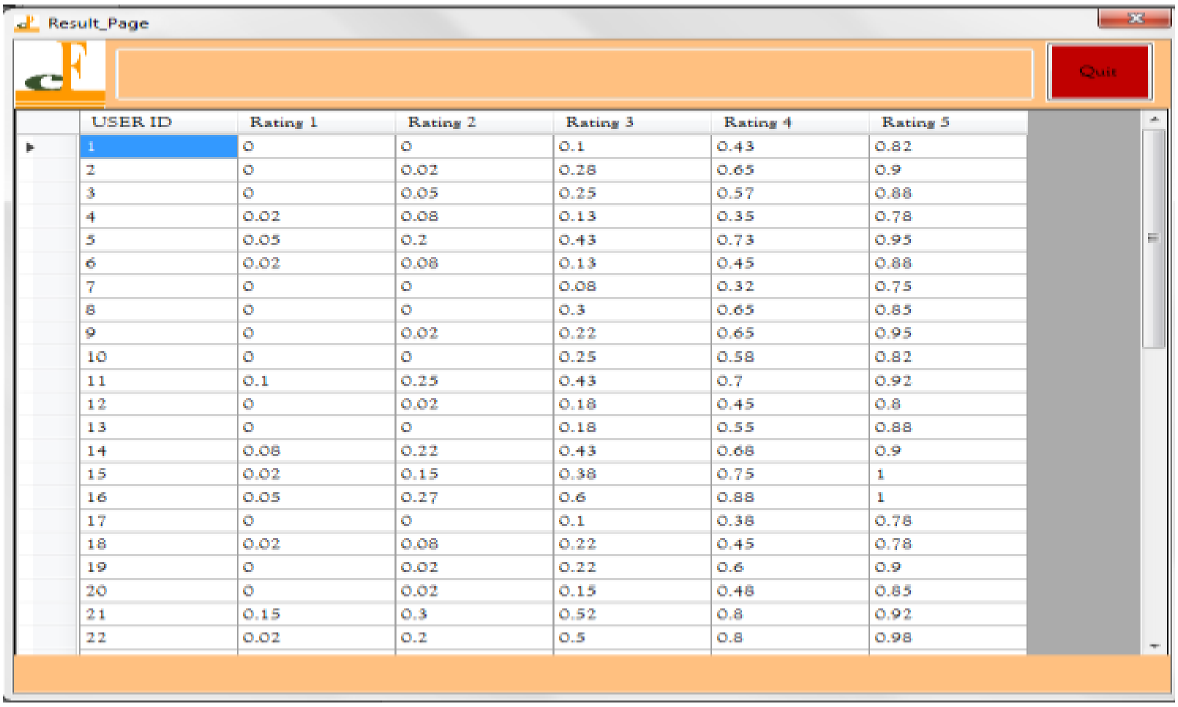
The cosine distance between User 1 and User 3 is

$$\frac{(5 * 2) + (1 * 4)}{\sqrt{4^2 + 5^2 + 1^2}\sqrt{2^2 + 4^2 + 5^2}} = 0.322 \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (4)$$

The distance value in User1 seems to be larger when compared to User2, which concludes that User 1 is closer or share more similarities with User 2 than User 3.

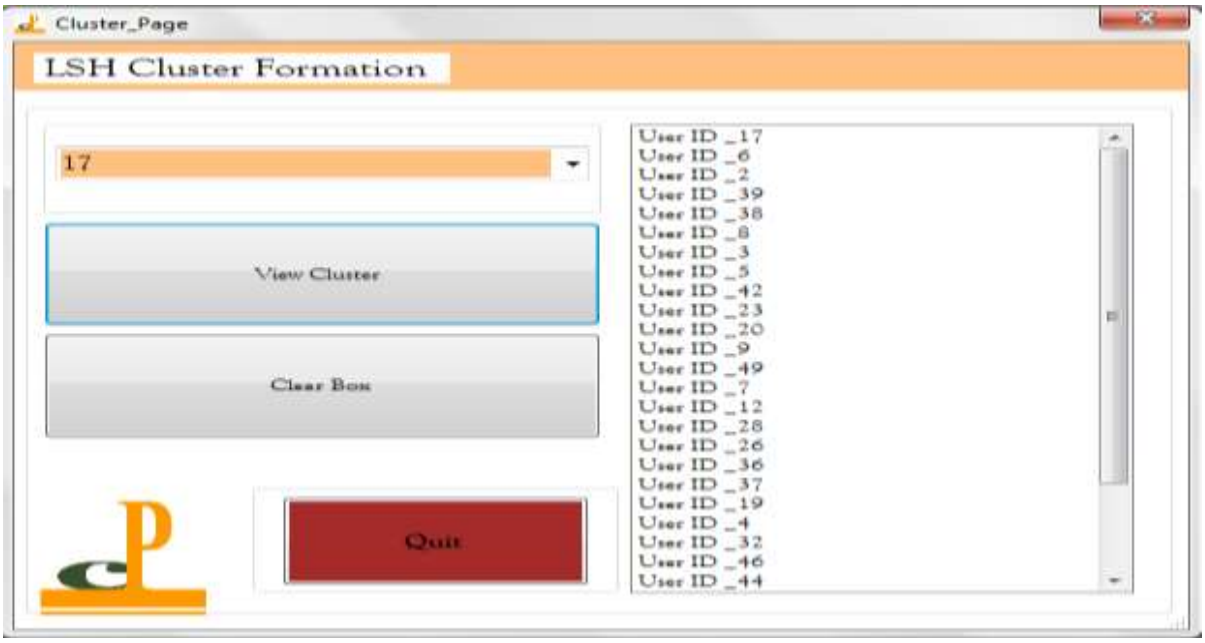
IV EXPERIMENTS AND RESULTS

We conducted a number of experiments using differ ent sets of features and similarity measures. Below, we describe in detail the setup used for the experiments, and report on some of the significant results we obtained. Experimentation using different sets of features and similarity measures was carried out on a set of randomly selected data from 100,000 ratings applied to 1682 movies by 943 users released on the MovieLens website on April, 2015. A detail description of the setup used for the experiments, and report on some of the significant results we obtained. The MovieLens dataset is a very popular dataset for research purpose and for evaluation of collaborative filtering algorithms. It is scaled from integer 1 to 5 and a user has at least rated nothing less than 20 movies. The normalization process reflected the true rating values of users, putting individual's rating habit into consideration as shown in fig 2. Fig 3 shows the user blocks obtained from the LSH method, the higher the number of value of similarity distance between any two users, the higher their co-occurrence and the higher the similarity between the two users as seen in fig 4. This co-occurrence approach creates ranks of neighbor users $u_a \in U$ by finding the number of collision of each user in the normalized rating matrix. The Singular value decomposition method was used to determine the dimensionality reduction of the matrix data as seen in fig 5 and evaluate the user-user similarity from the reduced matricesU_k. In U_k, each row corresponds to an individual user and in evaluating the degree of similarity between the active user and other users, the similarity between the active user row and other rows corresponds to the similarity between the active user and other users.



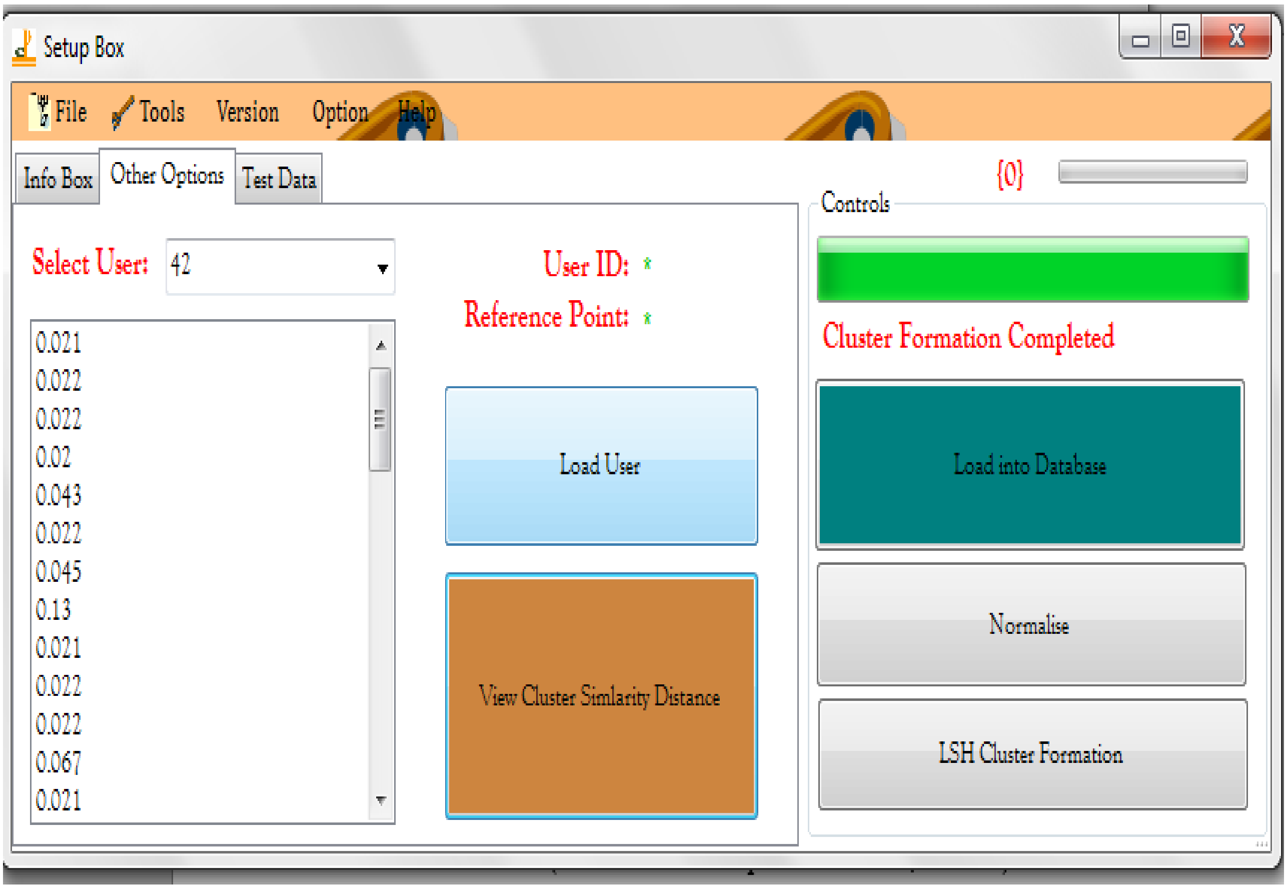
USER ID	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
1	0	0	0.1	0.43	0.82
2	0	0.02	0.28	0.65	0.9
3	0	0.05	0.25	0.57	0.88
4	0.02	0.08	0.13	0.35	0.78
5	0.05	0.2	0.43	0.73	0.95
6	0.02	0.08	0.13	0.45	0.88
7	0	0	0.08	0.32	0.75
8	0	0	0.3	0.65	0.85
9	0	0.02	0.22	0.65	0.95
10	0	0	0.25	0.58	0.82
11	0.1	0.25	0.43	0.7	0.92
12	0	0.02	0.18	0.45	0.8
13	0	0	0.18	0.55	0.88
14	0.08	0.22	0.43	0.68	0.9
15	0.02	0.15	0.38	0.75	1
16	0.05	0.27	0.6	0.88	1
17	0	0	0.1	0.38	0.78
18	0.02	0.08	0.22	0.45	0.78
19	0	0.02	0.22	0.6	0.9
20	0	0.02	0.15	0.48	0.85
21	0.15	0.3	0.52	0.8	0.92
22	0.02	0.2	0.5	0.8	0.98

Fig 2: Results of the normalization Process



User ID _17
User ID _6
User ID _2
User ID _39
User ID _38
User ID _8
User ID _3
User ID _5
User ID _42
User ID _23
User ID _20
User ID _9
User ID _49
User ID _7
User ID _12
User ID _28
User ID _26
User ID _36
User ID _37
User ID _19
User ID _4
User ID _32
User ID _46
User ID _44

Fig 3: Results from the LSH clustering



Select User: 42

0.021
0.022
0.022
0.02
0.043
0.022
0.045
0.13
0.021
0.022
0.022
0.067
0.021

User ID: *

Reference Point: *

{0}

Cluster Formation Completed

Fig 4: Results of the similarity distance

This evaluation process yielded the difference between mean value of the predicted rating and actual rating.

i. Difference ($|r_{um} - \hat{r}_{um}|$) =150

ii. Square($|r_{um} - \hat{r}_{um}|)^2$ = 220

To determine the Mean Average Error (MAE) and the Root Mean Square Error (RMSE) we used equations 5 and 6.

$$MAE = \frac{\sum_{m=1}^{N} |r_{um} - \hat{r}_{um}|}{N} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . (5)$$

$$RMSE = \sqrt{\frac{\sum_{m=1}^{N} (r_{um} - \hat{r}_{um})^2}{N}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. (6)$$

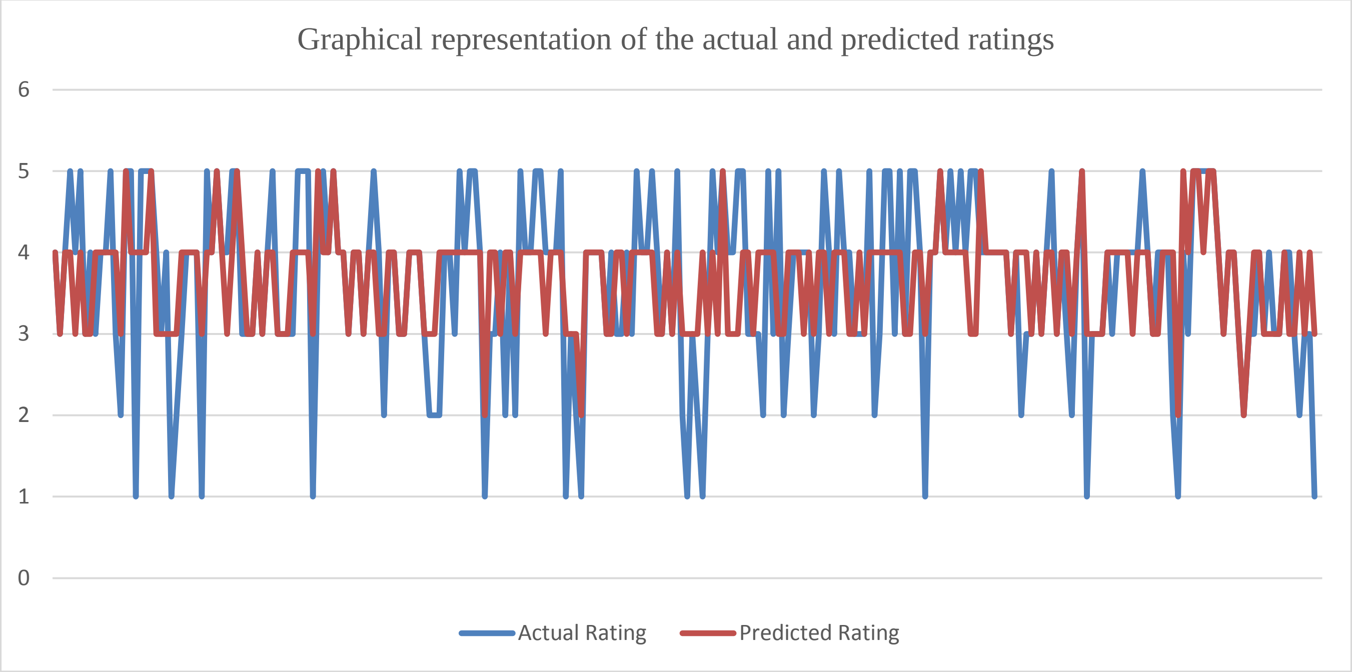


Fig 5: Graphical representation of the actual and predicted ratings.

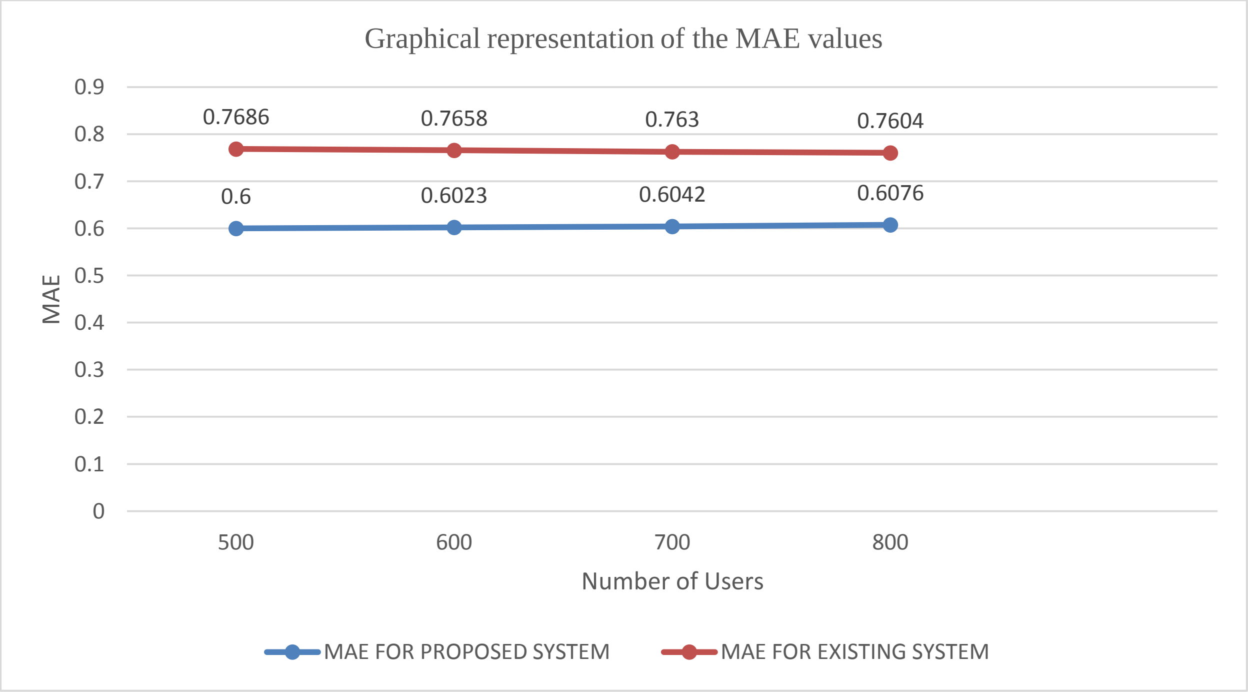


Fig 6: Graphical representation of the MAE values.

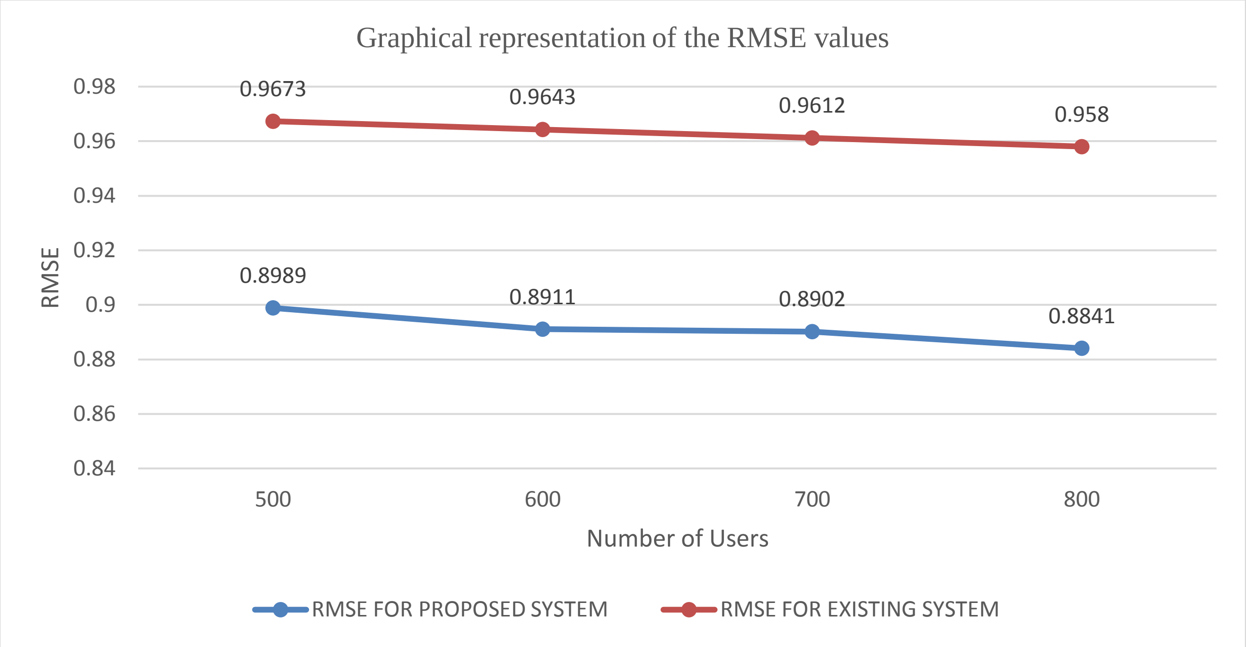


FIG 7: GRAPHICAL REPRESENTATION OF THE RMSE VALUES.

V. DISCUSSION OF RESULTS

To evaluate the hybrid model, the dataset was divided into two; training and test datasets. The training dataset took 80% while test dataset took 20%. For proper evaluation of the model, part of the data was used for performance evaluation to show its level of accuracy. The results of the predicted reports from the system shows that the mean average error and the root mean square error reduced drastically as the number of users was increased which signified an increase in accuracy of the system. The percentage accuracy of 90.2% was achieved when one thousand (1000) ratings data was used. The predicted results have been represented graphically in figs 6 to 7 showing an increase in prediction accuracy as number of users' increased. Fig 5 shows the graph of actual rating as against the predicted rating, from the graph the predicted was very close to actual, show the high predictive a ability of the system. Fig 6 presented the graph of the mean average error of the system as 0.6, 0.5991, 0.5985, 0.5982 for 500, 600, 700, 800 users respectively as compared with the mean average error of the existing system as 0.7686, 0.7658, 0.763, and 0.7604 for 500,600,700,800 users respectively. Fig 7 in the same manner presented the graph of the root mean square error of the system as 0.8989, 0.8911, 0.8902, 0.8841 for 500, 600, 700, 800 users respectively as compared to the root mean square error of the existing system of 0.9673, 0.9643, 0.9612 and 0.958 for 500, 600, 700, and 800 users respectively. These results implied an improvement in the predictive accuracy of the system.

VI. CONCLUSION

The computational speed and predictive ability of the system has been enhanced. The system maximized the potentials of pre-processing to clean up the dataset, the decoupling normalization method to generate a better representation that fully captures the true interest of users on the movies. The locality sensitive hashing technique was used to generate blocks and the singular value decomposition operation for dimensionality reduction to enhance the predictive ability of the recommendation model. Lastly, the results from the model shows a significant improvement of prediction accuracy and runtime of the system.

ACKNOWLEDGMENT

We wish to acknowledge the Department of Computer Science University of Port Harcourt , Nigeria for their support in the provision grant that was used to conduct this research. Also, we want to acknowledge Prof . E. O. Nwachukwu for his Mentorship.

REFERENCES

[1] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions. *IEEE Tras. on Knowledge and Data Eng.*, *17*(6), 734-749.

[2] Baker, K. (2005). *Singular Value Decomposition Tutorial.*

[3] Cai, Y., Leung, H., Li, Q., Min, H., Tang, J., & Li, J. (2013). Typicality-based Collaborative Filtering Recommendation. *IEEE*.

[4] Charikar, M. S. (2002). Similarity estimation techniques form rounding algorithms. *34th STOC*, (380-388).

[5] Ekstrand, M., Riedl, J., & Konstan, J. (2010). Collaborative filtering recommender system. *Trends Hum.-Comp Interact*, 81-173.

[6] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining Collaborative Filtering with Personal Agents for Better Recommendations. *GroupLens Research Project*.

[7] Konstan, A., & Riedl, T. (1999). Application of Dimensionality Reduction in Recommender System -- A Case Study. *GroupLens Research Group / Army HPC Research Center*, 1-12.

[8] Liang, H., Wang, Y., Christen, P., & Gayler, R. (2014). Noise-tolerant approximate blocking for dynamic real-time entity resolution. *PAKDD*, 449-460.

[9] Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing 7*, 76-80.

[10] Luo, X., Xia, Y., & Zhu, Q. (2012). Incremental Collaborative Filtering recommender based on Regularized Matrix Factorization. *Elsevier*, 271-280.

[11] Miller, S., & Reimer P., N. S. (2010). Geoshuffle: Location-aware, content-based music browsing using self-organizing tag cloud. *In Proceedings of 11th International Conference on Music Information retrieval*.

[12] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, j. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *In proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94,* (175-186). New York: ACM.

[13] Rong, J., Luo, S., & Zhai, C. (2003). Collaborative filtering with decoupled models for preferences and ratings. *In the Proc. of the 12th Conference on Information and Knowledge Management (CIKM).*

[14] Schafer, J., Konstan, B., & Riedl, J. (1999). Recommender Systems in E-commerce. *Proceedings of the First ACM Conference on Electronic commerce*, (158-161). Denver.

[15] Shardanand U., a. M. (1995). Social Information filtering: Algorithm for automating 'Word of mouth'. *In Proc. of CHI '95.* Denver.

[16] Than, C., & Han, S. (2013). Improving Recommender Systems by Incorporating Similarity, Trust and Reputation. *Journal of Internet Services and Information Security (JISIS)*, *4*, 64-76.

[17] Vozalis, M., & Margaritis, K. (2007). Using SVD and demographic data for the enhancement of generalized Collaborative Filtering. *Elsevier*, 3018-3037.

[18] Zhou, X., He, J., Huang, G., & Zhang, Y. (2014). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 717-733.

[19] Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale Parallel Collaborative Filtering for the Netflix Prize. *HP labs,* (1-12).