Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

# An Ensemble Clustering Recommender Model Base on SVD Algorithms

Danlami Muhammad
Department of Mathematical Science
Abubakar Tafawa Balewa University Bauchi, Nigeria

Abdulsalam Ya'u Gital
Department of Mathematical Science
Abubakar Tafawa Balewa University Bauchi, Nigeria

Haruna Chiroma
Department of Mathematical Science
Abubakar Tafawa Balewa University Bauchi, Nigeria

Kabiru Musa Ibrahim
Department of Mathematical Science
Abubakar Tafawa Balewa University Bauchi, Nigeria

Muhammad Isah Lamir
Department of Mathematical Science
Abubakar Tafawa Balewa University Bauchi, Nigeria

Mustapha Abdulrahman Lawal
Department of Information Technology
SRM Institute of Science and Tech Chennai, India

***Abstract*— Recommender Systems received a big push forward with the adoption of the technology by Amazon.com at the end of the 1990's. Their own implementation is based on the similarities of items rather than users. Recommender system assist business managers with making insightful decision about there product and customers respectively. However, existing recommender system suffers from challenges such as cold start, scalability and data sparsity. To address these challenges, this study proposed an improve CF recommender system (RS) and its application using SVD, ensemble clustering and context-aware, where these algorithms are combined to produce accurate prediction that will address the problem of scalability and sparsity. Experimental results shows that the result from this research suggests that the ensemble base clustering with SVD and context aware approach has better performance than the ensemble base clustering with KNN and context aware approach.**

***Keywords*— *Hidden Markov Model, Vertibi, HoneyPots, Intrusion Detection, Cyber Attacks and Denial of Service.***

## I. INTRODUCTION

The increasing importance of the Web as a medium for electronic and business transactions has served as a driving force for the development of RS technology. An important catalyst in this regard is the ease with which the Web enables users to provide feedback about their likes or dislikes. For example, consider a scenario of a content provider such as Netflix. In such cases, users are able to easily provide feedback with a simple click of a mouse. A typical methodology to provide feedback is in the form of ratings, in which users select numerical values from a specific evaluation system (e.g., five-star rating system) that specify their likes and dislikes of various items [1].

RSs development initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine, daily decisions. For example, it is common to rely on what one's peers recommend when selecting a book to read; employers count on recommendation letters in their recruiting decisions; and when selecting a movie to watch, individuals tend to read and rely on the movie reviews that a film critic has written and which appear in the newspaper they read [2].

RSs received a big push forward with the adoption of the technology by Amazon.com at the end of the 1990's. Their own implementation is based on the similarities of items rather than users. This allows the company to make claims such as: "users who bought this item also bought these other items" [2].

In order to create accurate and reliable recommendation, some processes need to be followed. Consider the figure below which depict the recommendation process as a form of black box [3]. It contains two inputs which serve as sources of information to the system. They include the users' profile and item/product information. User preferences and stored profile information should be related and be given explicitly by the user. It can also be extracted from other external sources such as web pages, buying behaviors, etc. Fig. 1 depict the process of recommendation system.

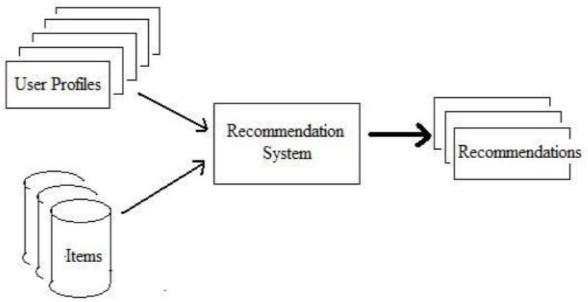


Fig. 1: Process of Recommendation (Source: [3]

However, Recommender system has been hampered by numerous challenges which include:

**i. Sparsity**

Sparsity is yet another problem encountered by recommender systems and this occurs mainly due to the fact that the number of items available to be rated is very high

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

when compared to the number of items already rated by the user. So, when a user item matrix is populated only a very few entries will be marked which causes the matrix to be sparse leading to poor recommendations. One of the possible solutions to this problem is by giving recommendations to a user by referring to the similarity in user profiles which assumes that if two users share similar interests, it is not really necessary to deduct conclusions solely based on the similarity of items they rated. This type of filtering is known as demographic filtering. Another method of addressing the sparsity problem was proposed in [4], which used Singular Value Decomposition (SVD) to reduce the dimensionality of sparse rating matrix [5].

***ii.*** **Scalability**

Scalability issue arises as the number of users, items and ratings information grows day by day. Even with the growing amount of information recommender systems are expected to respond quickly with recommendations for the online customers and it demands a higher scalability. The implementation of such system becomes complex and costly. The key challenge is in designing an efficient learning algorithm which is capable of handling such large datasets which keeps on growing. One of the solutions proposed is to use an online learning algorithm which processes the updates related to each user immediately and sequentially. Another method proposed to address the scalability issue uses a distributed algorithm where the computations are done in parallel in multiple machines.

**iii. Cold start**

It refers to a condition in which the RS cannot make the required prediction due to incomplete information about the user or item. This usually occur when either a new user or item get added into the system and is not having any initial rating [6, 7]. The probability of recommending an unrated item is very low and hence they might go unnoticed. One of the possible ways of tackling this situation is by having a set of motivated users who will be responsible for rating every new item. Also, when the user enters their first ratings into the system they expect to start getting recommendations which does not happen. It is because the number of ratings given to the system by the user is not sufficient enough to make good recommendation [8].

**iv.** ***Grey sheep***

The "grey sheep" problem occurs in CF technique and it happens when a user can be classified in more than one group of users. The similarity of this user with two or more groups is equal which makes the recommendations he will get inaccurate [9].

**V. Over-specialization**

The set of recommended items will be very homogeneous, the items will be very similar to the items the user already rated [9]. Recommendation system only recommends the items or product that user has liked or rated the highly in the past. Based on the past data available, system recommends similar
type of the items or products. System does not recommend these items that are different from anything that the user has seen before. Sometimes this might become problem because the user might want to try something new and the system would never make it happen. This is a problem that is associated with content based recommendation [10].

**vi. Popularity bias**

System cannot recommend items to someone with unique tastes. Sometime the user has unique taste than all other users in the system, that problem is known as popularity bias problem. This can be solved by the hybrid approach by using content-based filtering over collaborative-filtering [10].

**vii.** ***Shilling attack***

In a recommendation system where everyone can give the ratings, people may give lots of positive ratings for their own items and negative ratings for their competitors. It is often necessary for the collaborative filtering systems to introduce precautions to discourage such kind of manipulations [11].

To address these challenges, this study is expected to contribute to the improvement of CF RS and its application using SVD, ensemble clustering and context-aware, where these algorithms are combined to produce accurate prediction that will address the problem of scalability and sparsity. This study can be a learning paradigm in the field of RS as a contribution to the body of knowledge. RS is an area that is attracting a lot of research and it is perceived to be the future of e-commerce. This study will help to understand and evaluate further the important elements of measuring the performance, the feasibility of CF RS in the aspects of producing accurate, efficient and scalable RS. Besides, it will be helpful for other researchers who will be interested in doing advanced work on the same topic. The findings in this research will give an idea on the solution that best overcomes the issues of scalability, and sparsity in RS.

The subsequent parts of the papers are organized as follows: We review state of art algorithms in section 2. While in section 3 we look at research methodology, Experimental set up and approach along with the simulation environment with the performance metrics are also described. section 4, present the implementation, result and analysis. While in section 5, we summarize the research findings, conclusion and recommendations.

## II. STATE OF THE ART METHODS OF EVALUATION RS

### *A. SVD*

Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modelled as inner products in that space. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. For example, when the products are movies, factors might measure obvious dimensions such as comedy vs. drama, amount of action, or orientation to children; less well-defined dimensions such as depth of character development or "quirkiness"; or completely uninterpretable dimensions.

Accordingly, each item $i$ is associated with a vector $q_i \in R^f$, and each user u is associated with a vector $p_u \in R^f$. For a given item $i$, the elements of $q_i$ measure the extent to which the item possesses those factors, positive or negative. For a given user $u$, the elements of $p_u$ measure the extent of interest the user has in items that are high on the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

corresponding factors (again, these may be positive or negative). The resulting dot product,1 $q_i^T p_u$, captures the interaction between user u and item $i$ - i.e., the overall interest of the user in characteristics of the item. The final rating is created by also adding in the aforementioned baseline predictors that depend only on the user or item.

*B. SVD++*

Prediction accuracy is improved by considering also implicit feedback, which provides an additional indication of user preferences. This is especially helpful for those users that provided much more implicit feedback than explicit one. As explain earlier, even in cases where independent implicit feedback is absent, one can capture a significant signal by accounting for which items users' rate, regardless of their
rating value. This led to several methods [12-14] that modeled a user factor by the identity of the items he/she has rated. Here we focus on the SVD++ method [15], which was shown to offer accuracy superior to SVD. A second set of item factors is added, relating each item $\mathbf{i}$ to a factor vector $\mathbf{y_i} \in \mathbf{Rf}$ . Those new item factors are used to characterize users based on the set of items that they rated.

*C. Incremental SVD*

In a recommender system, the entire algorithm works in two separate steps. The first step is the offline step and the second step is the online execution step. The user-user and item-item similarity computation is done at offline stage of a recommender system. However, the actual prediction generation is done at run-time in the online step. Usually, the offline computations are very time consuming and is computed infrequently. For instance, a movie recommender site may compute the user-user or item-item similarity tables only once a day or even once a week. If the ratings database is static and if the user behaviour does not change significantly over a short period of time, this method works well.

The incremental algorithm based on SVD shown in Figure 2 is divided into an offline procedure and an online procedure. The offline stage is computationally intensive, and is performed only once. Finally, it can obtain three matrices $U_1$, $S_1$ and $V_1$ after performing SVD algorithm on $A_1$. The online stage is performed once the new matrix *A*2 enters, and also produces three matrices $U_2$, $S_2$ and $V_2$ after performing the incremental algorithm on the updated matrix $A_1, A_2$, using the results of the offline part [16].

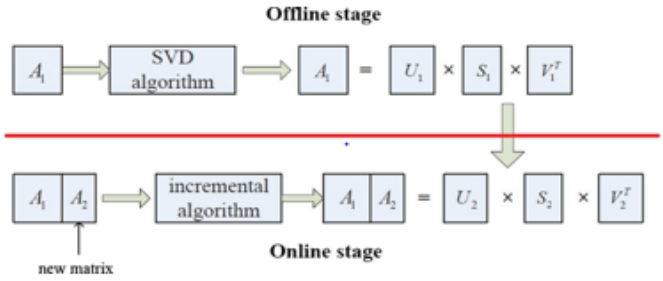


Fig. 2: Flowchart of offline and online stage of updating SVD Source: [16]

*D. Ensemble Clustering*

Clustering is performed to get insights into the data whose volume makes it problematic for analysis by humans. Due to this, clustering algorithms have emerged as meta learning tools for performing exploratory data analysis. A Cluster is defined as a set of objects which have a higher degree of similarity to each other compared to objects not in the same set [17].

The clustering model is introduced into collaborative filtering algorithm for, the clustering of users or items, the similar users or items clustered into the same cluster, looking for the nearest neighbour query can be directly completed within the class, do not have the traditional collaborative filtering algorithm in the whole data set in the query, greatly reduce the search scope and the amount of calculation [18]. Clustering method groups similar items or users into separate clusters to identify neighbourhood. Clustering techniques have been used either directly or as a pre-processing stage in recommender systems [19-22].

## III. RESEARCH FRAMEWORK

Improving accuracy of recommendation has been the major goal of every recommender system. Incorporating dimensionality reduction and clustering with ontology has no doubt improve scalability and sparsity of recommendation, but still left with problem of improving the accuracy of the clustering result as the EM clustering may lead to slow convergence [23] due to high overlapped clusters or unbalanced mixing coefficient [24], converges to local optima and/or may require both forward and backward probability [25]. This has no doubt reduce the accuracy of the clustering result and as such the accuracy of the recommendation is affected. This informs the need to further improve scalability and sparsity issue of recommendation which in the long run will improve the accuracy, precision and reliability of recommendation.

The solution to the above problem is of three-fold. Firstly, to improve accuracy of clustering result and reduce the dimension of the dataset, the research will use clustering ensemble. This will help improve scalability, accuracy, and reliability of recommendation. Secondly, reduce sparsity issue of recommendation by using a new similarity measure which will be applied on the selected partition in order to make rating prediction on the missing values. Thirdly, context-aware and SVD will be applied on both the item/user-based CF to improve scalability issue of recommendation. The research framework is depicted in Fig. 3

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

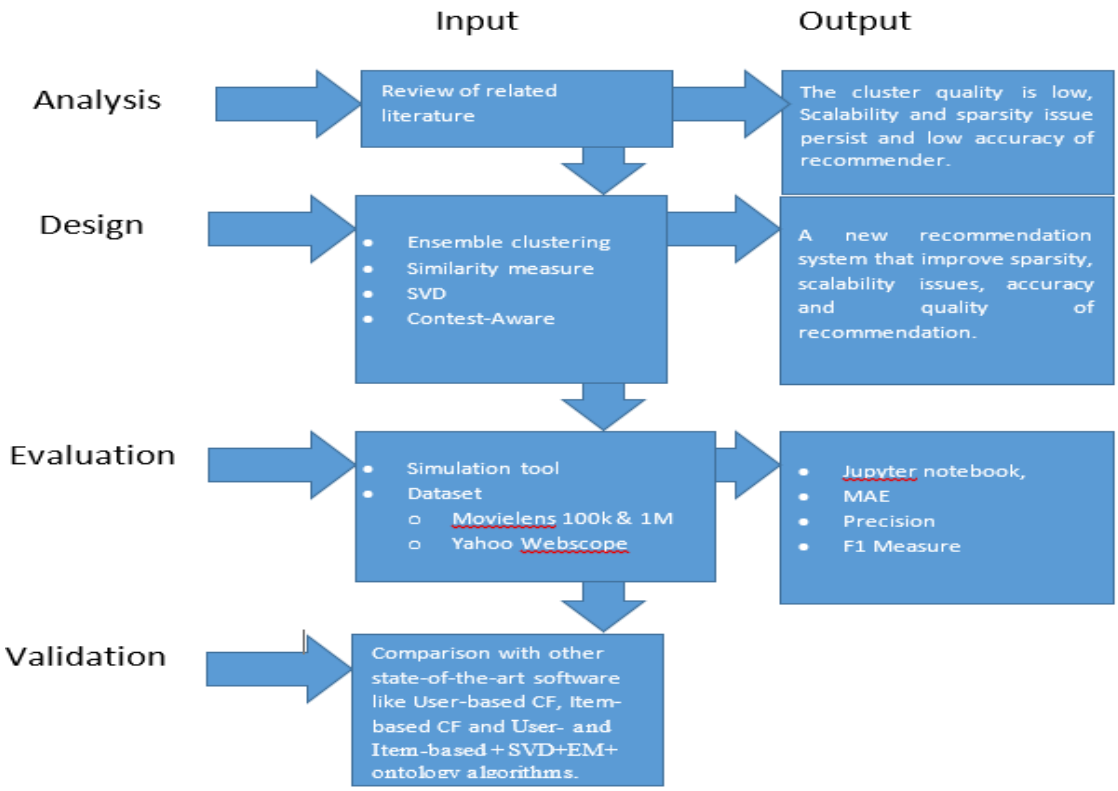


Fig. 3 Research Framework

## IV. RESULT AND ANALYSIS

The simulation tool for this research is the Jupyter notebook. Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Its Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

The proposed recommender system is evaluated using Movie Lens 100k. This dataset (MovieLens) is one of the well-known movie datasets that has been used for the evaluation of recommender systems. For the 100k dataset, it contains 100,209 anonymous ratings with the number of users and movies of 6040 and 3952, respectively. While for the 1M dataset, the files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.

In both datasets, the users have provided ratings on a 1-5-star scale, where 1 means that the user dislike the movie, and 5 point means the user like the movie so much. We select the users in the dataset who have provided at least 20 ratings.

The sparsity of the data 100k dataset is 0.936953, while 1M dataset is 0.9573. this was calculated using the equation below:

$$S_k = 1 - \frac{N_u}{m \times n} \quad (1)$$

Where $S_k$ =sparsity of the $m \times n$ matrix, $N_u$ is the count of non-zero element of $m \times n$ matrix and $m \times n$ is the total number of elements of the matrix which indicate that $m$ is the number of rows and $n$ the number of columns. Table 1 depict the features of the datasets comprising of movie, rating and user features,

TABLE I. DATASETS FEATURE

| | userId | movieId | rating | timestamp | title | genres |
|---|---|---|---|---|---|---|
| **0** | 1 | 31 | 2.5 | 1260759144 | Dangerous Minds (1995) | Drama |
| **1** | 7 | 31 | 3.0 | 851868750 | Dangerous Minds (1995) | Drama |
| **2** | 31 | 31 | 4.0 | 1273541953 | Dangerous Minds (1995) | Drama |
| **3** | 32 | 31 | 4.0 | 834828440 | Dangerous Minds (1995) | Drama |
| **4** | 36 | 31 | 3.0 | 847057202 | Dangerous Minds (1995) | Drama |
| **...** | ... | ... | ... | ... | ... | ... |
| **99999** | 664 | 64997 | 2.5 | 1343761859 | War of the Worlds (2005) | Action\|Sci-Fi |
| **100000** | 664 | 72380 | 3.5 | 1344435977 | Box, The (2009) | Drama\|Horror\|Mystery\|Sci-Fi\|Thriller |
| **100001** | 665 | 129 | 3.0 | 995232528 | Pie in the Sky (1996) | Comedy\|Romance |
| **100002** | 665 | 4736 | 1.0 | 1010197684 | Summer Catch (2001) | Comedy\|Drama\|Romance |
| **100003** | 668 | 6425 | 1.0 | 993613478 | 6th Man, The (Sixth Man, The) (1997) | Comedy |

100004 rows × 6 columns

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

From Table 1. The dataset contains: 100004 ratings of 9125 movies. We computed the statistical analysis to get the genre ratings as depicted in Table 7. The MovieLens dataset consists of ratings on a scale of 1-5 where 1 represents lowest rating while 5 represents the highest rating. However, different ratings could have different meanings to users. For instance, a rating of 3 might be good for one user while average for another user. To solve this ambiguity, big giants such as Netflix or YouTube have moved to bin. Therefore, in this work, we will work on binary ratings instead of continuous ratings to keep ourselves in sync with the latest research. Table 2. Depict the average rating of romance and science fiction (Sci-fi) movie.

TABLE II. THE AVERAGE RATING OF ROMANCE AND SCIENCE FICTION (SCI-FI) MOVIE

| | avg_romance_rating | avg_scifi_rating |
|---|---|---|
| 1 | 3.50 | 2.40 |
| 2 | 3.59 | 3.80 |
| 3 | 3.65 | 3.14 |
| 4 | 4.50 | 4.26 |
| 5 | 4.08 | 4.00 |

The next step is to incorporate context-aware into the recommendation. As seen in figure 2, we have both user-context and item-context. For each result of the cluster, we perform similarity computation based on either user context or item context to further improve the scalability of the recommendation. The essence is to find a cluster base on the partition that produce better cluster than any of the partitions.

The clustering ensemble techniques proposed in this study consist of two steps; generation step and consensus step. The generation step consists of some techniques available, but in this study, we use three clustering algorithms as a partition and then subject the result of each cluster to a voting technique as the consensus function in order to come-up with the final cluster as depicted in Fig. 4. This approach has significantly reduced the dimension of dataset to overcome slow convergence.

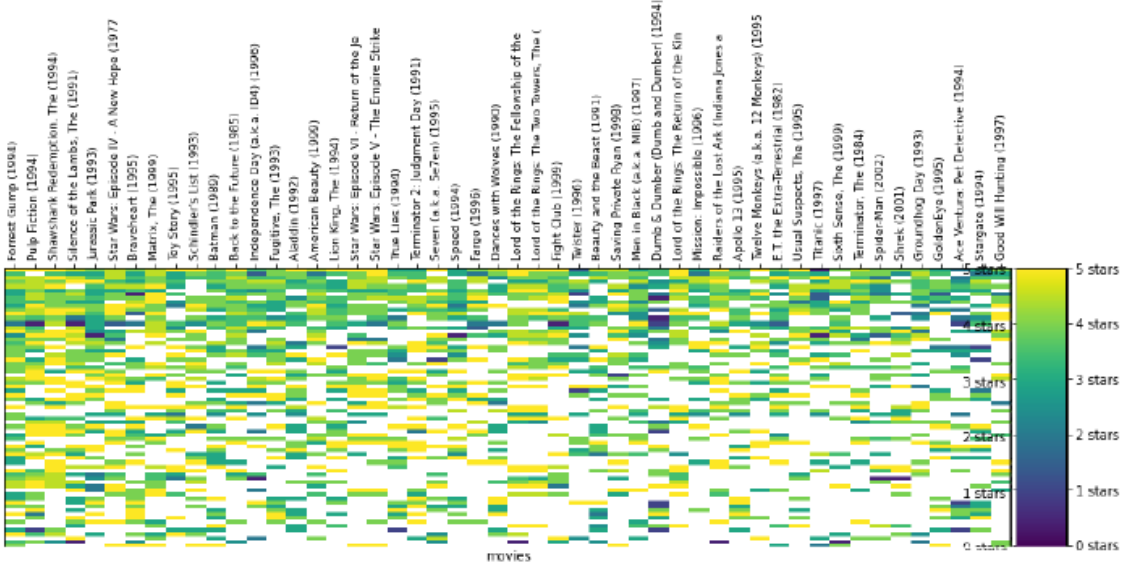


Fig. 4 Final cluster instances using the proposed ensemble clustering technique

By using the ensemble clustering algorithms to generate partitions and use consensus function on each of the partitions in order to aggregate the partitions into a single cluster shown in Fig. 4. This has enables us to use evaluation function to evaluate the generated cluster into an accurate and reliable group.

As stated earlier, recommendation system is facing the problem of how to overcome sparsity, scalability and cold start issues. Many researches were conducted in order to improve accuracy of recommendation and also to overcome the above-mentioned problem. But there is the need for improvement. To address this problem, After the clustering stage, similarity calculation is performed on the selected cluster. This similarity measure calculates the similarity between users/items. The result is use to predict the ratings of a missing values as depicted in Table 3. This is with a view to reduce the sparsity of the ratings and improve the accuracy of the result.

TABLE III SPARSE INSTANCES FROM THE DATASETS

| title | "Great Performances" Cats (1998) | $9.99 (2008) | 'Hellboy': The Seeds of Creation (2004) | 'Neath the Arizona Skies (1934) | 'Round Midnight (1986) | 'Salem's Lot (2004) | 'Til There Was You (1997) | 'burbs, The (1989) | 'night Mother (1986) |
|---|---|---|---|---|---|---|---|---|---|
| userId | | | | | | | | | |
| 1 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 3 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 4 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 5 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 6 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 4.0 | NaN |

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

Having calculated the sparsity of the dataset which was found to be 98.35% for the case of the MovieLens datasets. It is evident that there are a lot of 'NaN' values as most of the users have not rated most of the movies (see Table 3). This type of datasets with a number that is high of 'null' values are called 'sparse datasets. This is research aim is with a view to reduce sparsity of the dataset by predicting the missing values as shown in Table 4.

TABLE IV PREDICTED RATINGS OF A MISSING VALUES

| | Pulp Fiction (1994) | Jurassic Park (1993) | Star Wars: Episode IV - A New Hope (1977) | Shawshank Redemption, The (1994) | Toy Story (1995) | Silence of the Lambs, The (1991) | Matrix, The (1999) | Schindler's List (1993) | Star Wars: Episode V - The Empire Strikes Back (1980) | Forrest Gump (1994) |
|---|---|---|---|---|---|---|---|---|---|---|
| **3** | 5.0 | 3.0 | 5.0 | 2.0 | 2.0 | 5.0 | 5.0 | 4.0 | 5.0 | 1.0 |
| **8** | 5.0 | 4.0 | 4.0 | 5.0 | 4.0 | 4.0 | 3.0 | 5.0 | 4.0 | 5.0 |
| **18** | 3.5 | 3.0 | 4.0 | 4.0 | 4.0 | 3.0 | 4.0 | | 4.5 | 3.5 |
| **4** | 5.0 | 4.0 | 4.0 | 4.0 | 3.0 | 3.0 | | 4.0 | 5.0 | 5.0 |
| **33** | 5.0 | 5.0 | 5.0 | 5.0 | 4.0 | | 4.0 | | 4.0 | 5.0 |

5 rows × 300 columns

Dimensionality reduction such as SVD was performed on the output of the similarity measures to produce ranked list of items. It is performed on past ratings in order to get the target user/item similarities to other users/items. Thus, we define function to get the most rated movies and display via the heatmap shown in Fig. 5

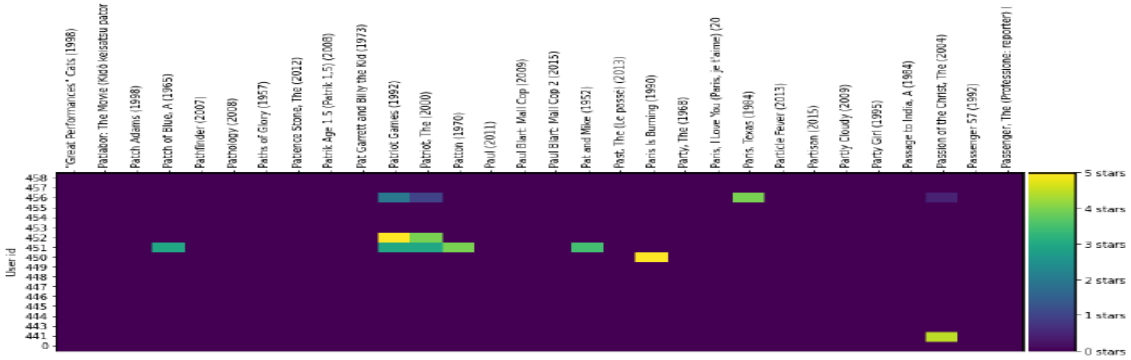


Fig. 5 Ranking of movies by ratings

To understand this heatmap in Fig. 5 Each column is a different movie. Each row is a different user. The cell's color is the rating that each user has given to each film. The values for each color can be checked in the scale of the right. The white values correspond to users that haven't rated the movie. Therefore, for each cluster, we apply SVD to obtain the decomposition matrices and for each matrix obtained from the decomposition step, we apply context (user or item context) and calculate the similarity. Finally, the output recommendation was made by the model as shown in Table 5.

A. Result Evaluation

To show the effectiveness of the proposed method in improving the scalability issue, we evaluate our method on MovieLens 100k. we evaluate the performance of the model by assessing how well the modelling performed in terms of KNN and SVD. We used predictive accuracy using statistical metrics (MAE and RMSE) in which the MAE and RMSE between the predicted and the actual ratings is measured. Thus, in table 6 and table 7, we present the performance results of our experiments for the two different methods base on SVD and KNN proposed in this study in terms of RMSE, MAE and Model Fitting.

TABLE VI RMSE FOR PROPOSED METHODS ON DIFFERENT NUMBERS OF TOP-N MOVIELENS

| Sparsity (%) | MovieLens | |
|---|---|---|
| | User and Item+ensemble +SVD+Context (RMSE) | User and Item+ensemble +KNN+Context (RMSE) |
| 97 | 80.31 | 71.01 |
| 97.5 | 80.45 | 71.06 |
| 98 | 80.48 | 71.08 |
| 98.5 | 80.51 | 81.13 |
| 99 | 90.54 | 81.18 |
| 99.5 | 94.59 | 91.34 |
| 100 | 94.69 | 92.55 |

Recommender System accuracy is popularly evaluated through two main measures: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Both are nice as they allow for easy interpretation: they are both on the same scale as the original ratings. From Table 6. It is notice that the best prediction accuracy in terms of the root mean square error (RMSE) was achieved by the proposed ensemble clustering+SVD+context aware achieving prediction accuracy of 94.69 at 100% sparsity level on the MovieLens datasets. This was better than the baseline algorithm (ensemble clustering+KNN+context aware) which attains 92.55% at 100% sparsity level.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

TABLE VII. RECOMMENDATIONS BY THE PROPOSED MODEL

| | movieId | title |
|---|---|---|
| 0 | 34 | Babe (1995) |
| 1 | 309 | Red Firecracker, Green Firecracker (Pao Da Shu... |
| 2 | 1 | Toy Story (1995) |
| 3 | 337 | What's Eating Gilbert Grape (1993) |
| 4 | 2 | Jumanji (1995) |
| 5 | 145 | Bad Boys (1995) |
| 6 | 378 | Speechless (1994) |
| 7 | 3 | Grumpier Old Men (1995) |
| 8 | 517 | Rising Sun (1993) |
| 9 | 129 | Pie in the Sky (1996) |
| 10 | 4 | Waiting to Exhale (1995) |
| 11 | 312 | Stuart Saves His Family (1995) |
| 12 | 291 | Poison Ivy II (1996) |
| 13 | 5 | Father of the Bride Part II (1995) |
| 14 | 344 | Ace Ventura: Pet Detective (1994) |
| 15 | 308 | Three Colors: White (Trzy kolory: Bialy) (1994) |
| 16 | 6 | Heat (1995) |
| 17 | 20 | Money Train (1995) |
| 18 | 343 | Baby-Sitters Club, The (1995) |
| 19 | 7 | Sabrina (1995) |
| 20 | 519 | RoboCop 3 (1993) |
| 21 | 215 | Before Sunrise (1995) |
| 22 | 8 | Tom and Huck (1995) |
| 23 | 9 | Sudden Death (1995) |
| 24 | 270 | Love Affair (1994) |
| 25 | 154 | Beauty of the Day (Belle de jour) (1967) |

Similarly, table 7 depict the MAE for proposed methods on different numbers of sparsity level on the MovieLens datasets.

TABLE 7. MAE FOR PROPOSED METHODS ON DIFFERENT NUMBERS OF TOP-N MOVIELENS

| Sparsity (%) | MovieLens | |
|---|---|---|
| | **User and Item+ensemble +SVD+Context (MAE)** | **User and Item+ensemble +KNN+Context (MAE)** |
| 97 | 77.93 | 67.93 |
| 97.5 | 77.97 | 68.87 |
| 98 | 77.95 | 79.51 |
| 98.5 | 87.98 | 79.91 |
| 99 | 87.95 | 87.95 |
| 99.5 | 97.99 | 87.03 |
| 100 | 98.07 | 98.04 |

From Table 7. It is notice that the best prediction accuracy in terms of the mean absolute error (MAE) was achieved by the proposed ensemble clustering+SVD+context aware achieving prediction accuracy of 98.07 at 100% sparsity level on the MovieLens datasets. This was better than the baseline algorithm (ensemble clustering+KNN+context aware) which attains 98.04% at 100% sparsity level.

In general, it was observed from the experiment, the prediction accuracy increases as the sparsity level increases. Thus, the result from this research suggests that the ensemble base clustering with SVD and context aware approach has better performance than the ensemble base clustering with KNN and context aware approach.

## V. CONCLUSION AND FUTURE WORK

Recommender system assist business managers with making insightful decision about their product and customers respectively. However, existing recommender system suffers from challenges such as cold start, scalability and data sparsity. To address these challenges, this study proposed an improve CF recommender system (RS) and its application using SVD, ensemble clustering and context-aware, where these algorithms are combined to produce accurate prediction that will address the problem of scalability and sparsity. Experimental results shows that the result from this research suggests that the ensemble base clustering with SVD and context aware approach has better performance than the ensemble base clustering with KNN and context aware approach.

In the future, we will use decision-support metrics (precision and F1) to compare the recommended items with the relevant ones by counting the overlap. Furthermore, we evaluate the general performance of the proposed method with other method from the literature using different recommendation datasets such as 1M Real World datasets..

## Acknowledgement

We wish to thank research our supervisors Ass. Prof. A.Y Gital, Prof. Haruna Chiroma and Dr. K. I. Musa for their immense contribution and academic guidance towards the success of this research work.

## REFERENCES

[1] Aggarwal, C.C., *Recommender systems*. Vol. 1. 2016: Springer.
[2] Levinas, C.A., *An analysis of memory based collaborative filtering recommender systems with improvement proposals*. 2014, Universitat Politècnica de Catalunya.
[3] Osmanlı, O.N., *A singular value decomposition approach for recommendation systems*. 2010.
[4] Sarwar, B., et al. *Analysis of recommendation algorithms for e-commerce*. in *Proceedings of the 2nd ACM conference on Electronic commerce*. 2000.
[5] Nilashi, M., O. Ibrahim, and K. Bagherifard, *A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques.* Expert Systems with Applications, 2018. **92**: p. 507-520.
[6] Park, S.-T. and W. Chu. *Pairwise preference regression for cold-start recommendation*. in *Proceedings of the third ACM conference on Recommender systems*. 2009.
[7] Park, Y.-J. and A. Tuzhilin. *The long tail of recommender systems and how to leverage it*. in *Proceedings of the 2008 ACM conference on Recommender systems*. 2008.
[8] Akhil, P. and S. Joseph, *A SURVEY OF RECOMMENDER SYSTEM TYPES AND ITS CLASSIFICATION.* International Journal of Advanced Research in Computer Science, 2017. **8**(9).
[9] Bouraga, S., et al., *Knowledge-based recommendation systems: a survey.* International Journal of Intelligent Information Technologies (IJIIT), 2014. **10**(2): p. 1-19.
[10] Shah, L., H. Gaudani, and P. Balani, *Survey on recommendation system.* International Journal of Computer Applications, 2016. **137**(7).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 02, February-2023

[11] Zhou, W., et al., *Shilling attack detection for recommender systems based on credibility of group users and rating time series.* PloS one, 2018. **13**(5).
[12] Koren, Y., R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems.* Computer, 2009. **42**(8): p. 30-37.
[13] Paterek, A. *Improving regularized singular value decomposition for collaborative filtering*. in *Proceedings of KDD cup and workshop*. 2007.
[14] Mnih, A. and R.R. Salakhutdinov. *Probabilistic matrix factorization*. in *Advances in neural information processing systems*. 2008.
[15] Koren, Y. *Factorization meets the neighborhood: a multifaceted collaborative filtering model*. in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008.
[16] Zhou, X., et al., *SVD-based incremental approaches for recommender systems.* Journal of Computer and System Sciences, 2015. **81**(4): p. 717-733.
[17] Nerurkar, P., et al., *Empirical analysis of data clustering algorithms.* Procedia Computer Science, 2018. **125**: p. 770-779.
[18] Xiaojun, L., *An improved clustering-based collaborative filtering recommendation algorithm.* Cluster computing, 2017. **20**(2): p. 1281-1288.
[19] Adomavicius, G. and A. Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.* IEEE transactions on knowledge and data engineering, 2005. **17**(6): p. 734-749.
[20] Gong, S., *A collaborative filtering recommendation algorithm based on user clustering and item clustering.* JSW, 2010. **5**(7): p. 745-752.
[21] Nilashi, M., O. bin Ibrahim, and N. Ithnin, *Hybrid recommendation approaches for multi-criteria collaborative filtering.* Expert Systems with Applications, 2014. **41**(8): p. 3879-3900.
[22] Pham, M.C., et al., *A clustering approach for collaborative filtering recommendation using social network analysis.* J. UCS, 2011. **17**(4): p. 583-604.
[23] Hua-Yan, S., et al. *Accelerating EM Missing Data Filling Algorithm Based on the K-Means*. in *2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC)*. 2018. IEEE.
[24] Xiang, W., et al., *An exact line search scheme to accelerate the EM algorithm: Application to Gaussian Mixture Models identification.* Journal of Computational Science, 2020: p. 101073.
[25] GeeksforGeeks. *Expectation-Maximization Algorithm*. 2019 14/05/2019 [cited 2021 09/03/2021]; Available from: https://www.geeksforgeeks.org/ml-expectation-maximization-algorithm/.