Addressing Data Sparsity in Collaborative Filtering Based Recommender Systems Using Clustering and Artificial Neural Network

Ashrf Althbiti [1,2,*] ,Rayan Alshamrani [1,2] ,Tami Alghamdi [1,3] ,Stephen Lee [4] , and Xiaogang Ma[1,*]
[1]*Department of Computer Science, University of Idaho,* Moscow, ID 83844- 1010, USA
[2]*Department of Information Technology, Taif University,* Taif, Makkah 26571, Saudi Arabia
[3]*Department of Computer Science, Albaha University,* Albaha, 65527, Saudi Arabia
[4]*Department of Mathematics and Statistical Science, University of Idaho,* Moscow, ID 83844- 1010, USA
[*]Correspondence: alth3428@vandals.uidaho.edu; max@uidaho.edu

***Abstract*—Collaborative Filtering (CF) is fundamentally characterized by Recommender Systems (RSs), which have recently attracted researchers' attention. The ever-increasing data about users and items and the emergence of machine learning approaches have motivated the recent development of CF. The sparsity caused by the lack of recorded transactions and data makes it challenging for CF to distinguish between users' similar preferences. As a result of the data sparsity issue, CF ultimately lacks the ability to generate useful recommendations and suffers from poor performance. This paper proposes a novel model that uses clustering and artificial neural network to address the issue of data sparsity in CF. The proposed model CANNBCF, a short name for Clustering and Artificial Neural Network Based Collaborative Filtering, is evaluated using four different datasets from four popular domains (books, music, jokes, and movies). The proposed model shows its superiority to solve the sparsity issue that the traditional CF technique encounters. In this paper, eight experiments are conducted to evaluate the performance of CANNBCF. The evaluation criteria include accuracy, precision, recall, F1-score, and Receiver Operating Characteristics used to examine the proposed model. The results of the experiments show that CANNBCF effectively solves the sparsity issue, improves the quality of recommendations, and demonstrates promising prediction accuracy.**

Keywords—Collaborative Filtering; Data Mining Algorithms; Data Sparsity; Evaluation; Recommender Systems

I. INTRODUCTION

Collaborative filtering (CF) is fundamentally characterized by recommender systems (RSs), which have recently gained and attracted researchers' attention [1]. The early evolution of RSs over the last few years is the direct result of the advancements of information retrieval (IR) and information filtering [2]. The main functionality of RSs can be seen as the social process of suggestions and recommendations which ultimately reduce the variety of selections and present more useful selections to users; thus, it was rational to consider RSs as a subfield of IR. Yet, the mid-1990s did witness the birth of RSs as an independent field of study [1,3]. The ever-increasing data about users and items and the emergence of machine learning approaches have motivated the recent development of RSs. Although the development of RSs technology has been influentially driven by the exceeding use of the web [3], the richness of RSs applications offers researchers many opportunities for building more robust applications [4].

CF can be defined as systems and software tools that automatically and effectively generate recommendations of the most suitable items to a target user by predicting a user's predilections and preferences [5]. The prediction process relatively depends on the previous knowledge of users' interests, the description of items, and the interactions between users and items [6]. The objective of developing CF systems is to effectively reduce the overload of available selections by exposing users to the most relevant and suitable items and services from a variety of alternatives. These systems generate personalized and un-personalized recommendations by developing models that analyze users' data, opinions and behavior, and also analyze items' descriptions and attributes to strategically predict a user's preferences. Yet, they analyze not only a user's data and opinions but also the opinions of the opinions of users with similar preferences for augmenting prediction accuracy [15].The evolution of the CF arises from a phenomenon in which people seek recommendations from others who share their preferences [1, 7, 8].

Using machine learning algorithms to develop CF systems is extremely valuable for both academia and industry and is becoming increasingly crucial for recommending items and services to users. Different techniques and methodologies employed in CF systems typically follow three steps carried out in succession: data preprocessing, model learning, and the interpretation of results [9].

978-0-7381-4394-1/21/$31.00 ©2021 IEEE

The particular objective of this empirical study is to propose a model that addresses the sparsity problem of CF. The research questions are as follows:

1- How could clustering and artificial neural network (ANN) solve the sparsity problem of CF?
2- Does the use of different datasets collected from popular domains present a significant difference in the accuracy of recommendation predictions when applying clustering and ANN?

The structure of this paper is as follows. Section II discusses research relevant to the present study. Section III discusses the proposed model. Section IV discusses the experiment. Section V deliberates the results and the evaluation. Section VI discusses the discussion of the results. Section VII considers the conclusion and the future work.

II. RELATED RESEARCH

A recommendation technique that depends on a rating structure encounters a common problem: sparsity [17, 18]. The data sparsity in CF negatively affects the quality of the recommendations. It arises in several application domains due to user interactions with a small portion of items [16]. For instance, MovieLens dataset includes a rating matrix where users rate movies. The rating matrix is not fully specified; approximately 90% of the matrix has null values. Thus, the traditional CF techniques suffer from the sparsity problem. As a result, it is difficult for CF to generate good recommendations. Previous research has been conducted to solve the sparsity problem. Researchers in [17] use the average of the provided ratings to fill the rating matrix. Other researchers propose models including Matrix Factorization (MF), computational intelligence, and mathematical calculation [19].

MF techniques are introduced to solve the problems of data sparsity and large dimensionality. Several participating teams have used MF when Netflix in 2009 offered a prize for the best CF system that yields a better accuracy [20]. MF techniques map users and items into a reduced latent space. The dimensionality reduction can capture high-level patterns in the rating matrix, explain the relationships between users and items, and capture their most hidden characteristics [21, 22]. Researchers in [25] presented a number of MF models for CF that leverage metadata as a bridge between the preferred items by users in different domains to solve the sparsity problem. They claimed that in case the underlying knowledge graph connects items from different domains, their models can provide better recommendations to new users while keeping a better trade-off between recommendation accuracy and diversity.

Other researchers have tackled the sparsity problem by proposing a model that relies on machine learning and data mining algorithms. Researchers [19] presented a sparsity alleviation recommendation approach that achieves a better product recommendation performance. In their research, the new sparsity alleviation approach addressed the null or zero values in the rating matrix. The process was done through the use of the multiplication convergence rule and constraint condition. Moreover, another method was proposed to overcome the data sparsity problem by combining MF model with Linked Open Data (MFLOD) [16].

Other approaches that involve mathematical calculations are characterized into probability methods and similarity methods. Researchers have proposed a model that learns domain concept and use a probabilistic method to find similar patterns in the rating matrix [23]. Moreover, a novel method called FRAIPA was designed to solve the sparsity and dynamic data problems, and it ultimately improved the prediction accuracy [24]. Considering the similarity methods, researchers in [26] proposed a novel approach that computes the similarity between users not only based on the items rather the attributes of the items. In their model, users' liking and disliking of the similar characteristics of a particular item were considered separately.

III. THE PROPOSED MODEL

The aim of this paper is to systematically solve the sparsity problem of CF using: (1) clustering and (2) ANN, as demonstrated by Fig. 1.

A. Data preprocessing

The first step is to perform the preprocessing process to prepare these datasets to be used by the proposed algorithm. While examining the main preprocessing steps that real-life data typically requires, this study performs these steps. Researchers in [10] point out that dropping users who rate less than the average rated items per user in the whole dataset may improve the algorithm's performance. It is noteworthy that the problem of cold start [11] occurs when a user rates a few items or has not rated any item. This problem is also applicable to items, as an item may have been rated by few users or not rated at all. Therefore, this study calculates the average number of rated items per user in each dataset and considers only users and items that have the minimum required number of ratings.

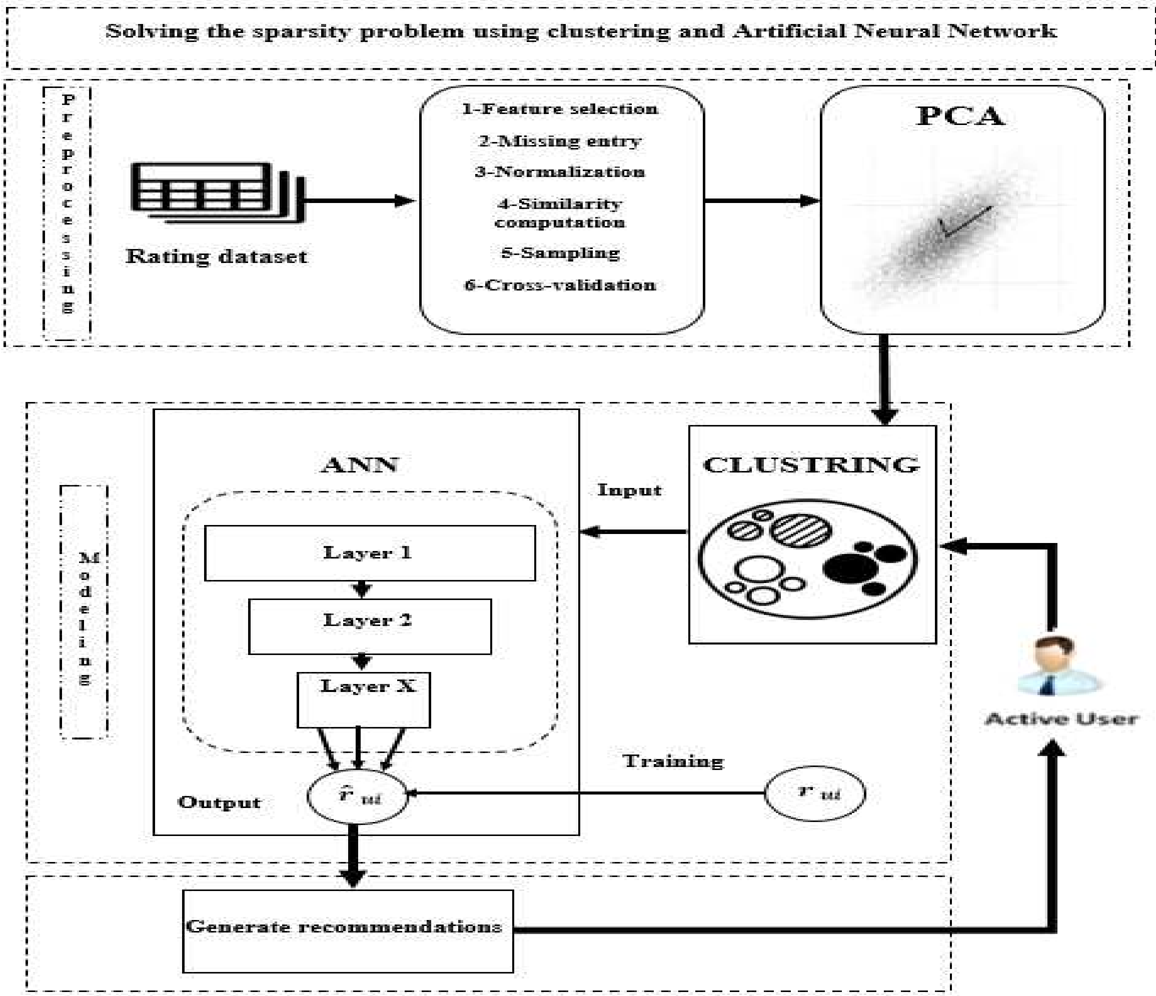

FIG. 1. THE FREAMWORK ARCHITECTURE OF CANNBCF

A missing entry or interaction in the rating datasets is a normal issue as users may rate a few items or items might be rated by a few users. These missing ratings introduce the problem of data sparsity, which is highly correlated with the problem of cold start. The missing entry or interaction in the rating datasets is addressed using the average of the provided ratings to fill the rating matrix.

Another essential preprocessing step is considered when preparing the data. This step ascertains the rating normal distribution [11]. Although the distribution of user-item interactions in real-life datasets is normally skewed, several normalization techniques are introduced to uniformly distribute the ratings over the items. The normalization step also helps to map users' ratings to a representative universal scale, since the rating process depends on each user's personal scale. One common normalization scheme is often used to identify the polarity of a rating whether it is positive, neutral, or negative. It is known as mean-centric. Yet, although the normalization of ratings might improve the performance of the algorithm, there is a chance that it will introduce undesirable effects in some cases.

Moreover, calculating the similarity between users and items is a crucial step for generating predictions and recommend items. It can have a positive effect on both the accuracy and performance of CF [11]. Computing the similarity allows the ratings of trusted neighbors to be used while predicting the ratings for unseen items to a similar user. It also helps to determine the effect of the neighbors' ratings. These algorithms highly depend on selecting an appropriate distance measure that computes the similarity weights. The first common approach to compute the similarity weights is Cosine Vector. It measures the similarity between two instances after representing them in the form of a vector x_a and x_b. Another common measure that depends on the effects of the mean and the variance of the ratings is known as Pearson Correlation coefficient.

Concerning the sampling size to train and test the proposed algorithm, this study uses a common technique *K*-fold cross-validation to avoid the overfitting problem (a.k.a., over-specialization problem). The process is done for *k* times until each group is treated as validation, while the remaining is treated as training data. Hence, *5*-fold cross-validation is used where the average performance of the *K*-learned models is calculated.

The last step of the preprocessing process is dimensionality reduction. The most representative method is the Principal Component Analysis (PCA) [9]. It is used for the feature extraction process, where an m- dimensional space is reduced into an n-dimensional space. The reduction process is done while most of the information is retained and kept.

B. Clustering

The unsupervised machine learning algorithm used first is clustering. The main problem of developing a CF algorithm is the number of operations needed to compute distances between neighbors to find the best k-nearest neighbors. The clustering algorithms can play a vital role to compute the distance between two objects. Researchers claim that clustering is used to improve efficiency because the number of operations is reduced [9]. Clustering assigns items to groups so that items in the same group are more similar than items in other groups. Minimizing intra-cluster distances while maximizing inter-cluster distances is the main goal of a clustering algorithm. The k-means clustering algorithm is used because it is an extremely efficient algorithm [9]. In particular, the used clustering technique is the locality-sensitive hashing (LSH). LSH is a set of techniques that dramatically hashes similar input items into the same "buckets" with high probability [12].

C. ANN

The second supervised machine learning algorithm that is used is ANN. It is characterized as forecasting algorithms that predict the future ratings for unseen items based on the previously recorded patterns [5,27]. ANN learns a model by composing layers that perform functions in order to predict ratings for unseen items. It is noteworthy that ANN can have any number of layers. They are input, hidden, and output layers. Also, there are different functions that interconnect these layers. For instance, the simplest implementation of ANN is the perceptron model, which has two functions known as Threshold and Summing. This demonstrates the simplest case of ANN that trains a model with sufficient data to obtain the required rating prediction.

The proposed neural network architecture comprises three inputs: (1) user features, (2) item features, and (3) similar user features, as demonstrated by Fig. 2. It passes each input to a separate fully connected layer and batch normalization Rectifier (ReLU). Moreover, it takes the average of the embedding obtained for the nearest neighbor user to obtain a single representation for the similar users to a target user. It then concatenates the three representations and passes them through three layers of fully connected layers, two of which have ReLU as their activation function, and the third of which the last layer outputs the predicted rating.

IV. EXPERIMENT

A. Dataset

In order to address the proposed research questions, this study conducts a set of experiments. It identifies the most used datasets from several research papers [10, 13, 14]. The researchers survey the publicly available recommendation datasets, which provide researchers a way to evaluate the performance of their proposed algorithms. The selection of these domains is motivated by [14], since the researchers claim that the most used datasets are from these domains.

These datasets are "Jester," "MovieLens," "Book-Crossing" and "Last.fm". The availability of these datasets on the Web and the typical use by developers and researchers in CF techniques motivate the selection. Therefore, this study uses these datasets to evaluate the proposed algorithms.

(1) Jester dataset
The Jester is a WWW-based joke RS. The Jester dataset contains 4.1 million ratings entered by 73,496 users for 100 jokes. User ratings are explicitly recorded on a real value, ranging from -10 to +10.

(2) MovieLens 1 million
The MovieLens dataset is a movie RS. It has 1.1 million ratings entered by 6,040 users for 3,592 movies. User ratings are explicitly recorded on a real value, ranging from 1 to 5.

(3) Book-Crossing

The Book-Crossing dataset is a book RS. It has 1 million ratings entered by 278,858 users for 271,379 books. User ratings are explicitly recorded on a real value, ranging from 1 to 5.

(4) Last-fm
The Last-fm is a popular music platform. It contains 92,834 ratings entered by 1,892 users for

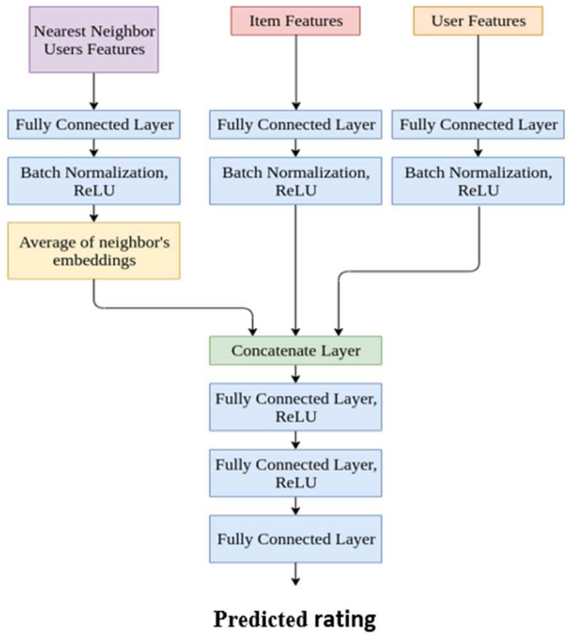

FIG. 2. THE ARCHITECTURE OF THE NEURAL NETWORK USED FOR PREDICTING THE RATINGS

17,632 items. User ratings are implicitly inferred by mapping listening counts into real values of 1 to 5. Equation (1) is defined in [10] as follows:

$$r = \begin{cases} \lfloor log_{10} l \rfloor + 1, if\ \lfloor log_{10} l \rfloor + 1 \leq 5 \\ 5, otherwise \end{cases} \quad (1)$$

where l is the listening count, r is the implicit rating after transforming and $\lfloor . \rfloor$ is the rounding operator towards zero.

B. Experimental setup

We implement the codes using PyTorch library for the neural network training and inference. We also use Faiss library for fast nearest neighbor search. The classes are not balanced; therefore, the distribution of the number of samples per each class is not uniform. The imbalance will lead to a biased prediction because during the learning, the classifier gets biased to the highly repeated classes. As a result, we use weighted cross-entropy as our loss function, where the weights are proportional to the inversion of the class frequencies. The training samples are the triples of (userID, itemID, rating) mapped into (userFeatures, userNeighborFeatures, itemFeatures, rating). The userFeatures and itemFeatures are obtained from the PCA output, and userNeighborFeatures are obtained by finding the nearest neighbors of the user. We train the rating predictor for 150 epochs using Adam optimizer with the learning rate of e^{-4}.

C. Evaluation

The evaluation process is done based on historical datasets. This type of evaluation is known as offline evaluation, in which the data are previously collected. This research aims to evaluate the proposed method over a range of evaluation criteria widely used when deciding which algorithm to use. These evaluation criteria include accuracy, precision, recall, the area under the receiver operating characteristic (AUROC), and F-score. They are used to evaluate the accuracy of usage predictions. The process of this evaluation considers using datasets consisting of users and their consumed or preferred items.

(1) Accuracy is the most intuitive performance measure and is estimated as a ratio of correctly predicted observations to the total observations, given by (2),

$$Accuracy = \frac{tp+tn}{tp+fp+fn+tn} \quad (2)$$

(2) Precision (a.k.a., positive predictive value) is a ratio of the number of true positive recommended items to all the recommended items, given by (3),

$$Precision = \frac{tp}{tp+fp} \quad (3)$$

(3) Recall (a.k.a., true positive rate, sensitivity) is a ratio of the number of true positive recommended items to all the consumed items, given by (4),

$$Recall = \frac{tp}{tp+fn} \quad (4)$$

(4) AUROC curve is a useful measurement for comparing several algorithms independently of application [5]. It is a graph showing the performance of a classification model at all classification thresholds. The receiver operating characteristic (ROC) is a probability curve, and an area under the curve (AUC) represents the degree or measure of separability. It tells how much a model is capable of distinguishing between classes.

(5) F1-score is the harmonic mean between the precision and the recall and reveals a better quantification than either the precision or the recall [3], given in (5),

$$F1 = \frac{2*precision*recall}{precision+recall} \quad (5)$$

V. RESULTS

In our experiments, we respectively compute the accuracy, precision, recall, AUROC, and F1-score. These evaluation criteria are computed using the previously discussed datasets for the baseline (NBCF) and the proposed (CANNBCF) algorithms. Table 1 demonstrates the comprehensive comparisons of the evaluation criteria between NBCF and CANNBCF.

We demonstrate the ROC and Precision-Recall results of NBCF algorithm and the CANNBCF algorithm in Figs. 3-18. The macro and micro average ROC curves are also shown in Figs. 3-18. In macro-average, we compute the AUROC independently for each class and then take the average, hence treating all classes equally. However, in micro-average, we aggregate the class frequencies to compute the average AUROC. For the NBCF, the AUROC of each class is highly correlated with the class frequency. For example, in the Book-crossing dataset, rating 4 and 3 have the highest class frequency, and their corresponding AUROC are the highest. The least frequent classes are those which often have lower AUROC and area under precision-recall scores. The AUC of macro average ROC is lower than the AUC of micro average ROC in all cases. The reason is that the classes that the model is underperforming often have a small amount of training data, so their corresponding class frequency is low. As a result, when their class frequency contribution is considered equal to the other classes, it adversely reduces the macro-average score. The area under ROC and Precision-Recall curves are both improved for all the five classes by using CANNBCF compared to the NBCF.

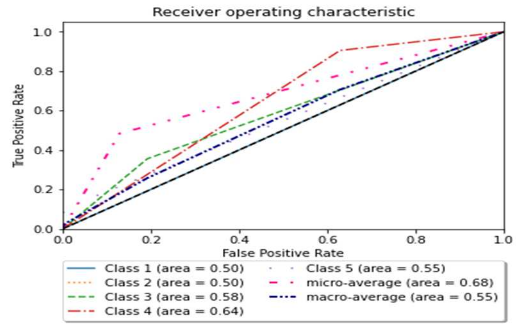

FIG.3. ROC FOR BOOK-CROSSING DATASET USING NBCF

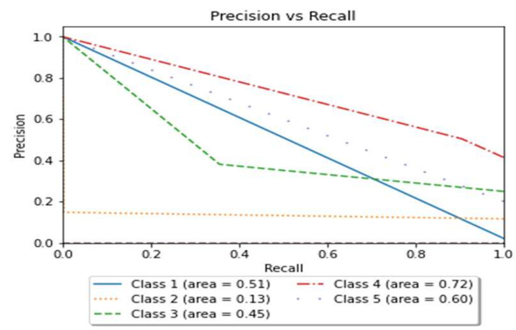

FIG.4. PRECISION-RECALL FOR BOOK-CROSSING DATASET USING NBCF

TABLE 1 THE COMPREHENSIVE COMPARISONS

	NBCF					CANNBCF				
	Accuracy	*Precision*	*Recall*	*AUROC*	*F1 Score*	*Accuracy*	*Precision*	*Recall*	*AUROC*	*F1 Score*
Book-crossing	0.53	0.50	0.48	0.59	0.40	**0.75**	**0.73**	**0.73**	**0.91**	**0.73**
Jester	0.56	0.57	0.51	0.64	0.43	**0.80**	**0.76**	**0.75**	**0.94**	**0.75**
Last.fm	0.63	0.65	0.66	0.61	0.61	**0.93**	**0.89**	**0.89**	**0.96**	**0.89**
Movielens	0.51	0.46	0.40	0.54	0.27	**0.71**	**0.53**	**0.53**	**0.79**	**0.53**

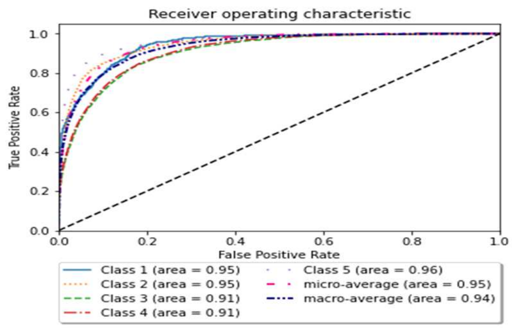

FIG.5. ROC FOR BOOK-CROSSING DATASET USING CANNBCF

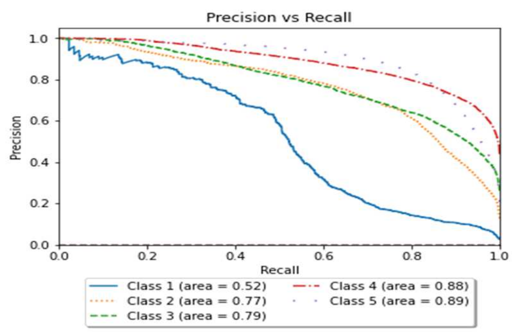

FIG.6. PRECISION-RECALL FOR BOOK-CROSSING DATASET USING CANNBCF

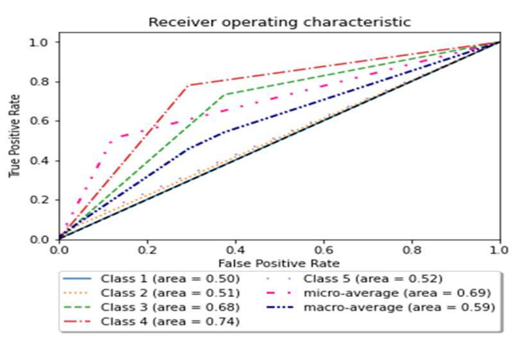

FIG.7. ROC FOR JESTER DATASET USING NBCF

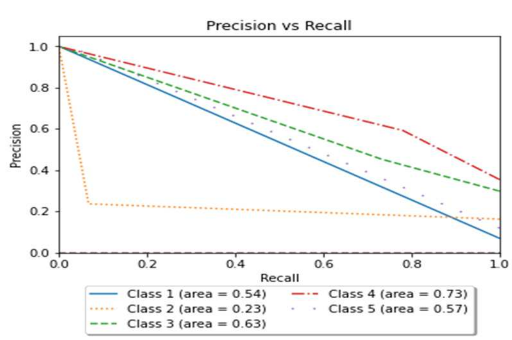

FIG.8. PRECISION-RECALL FOR JESTER DATASET USING NBCF

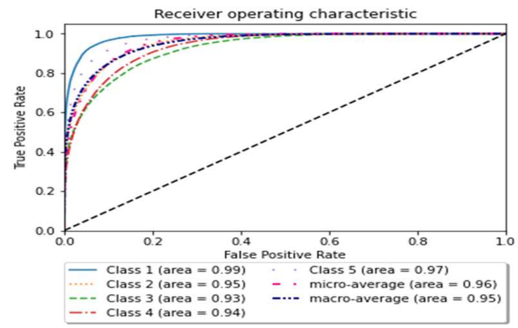

FIG.9. ROC FOR JESTER DATASET USING CANNBCF

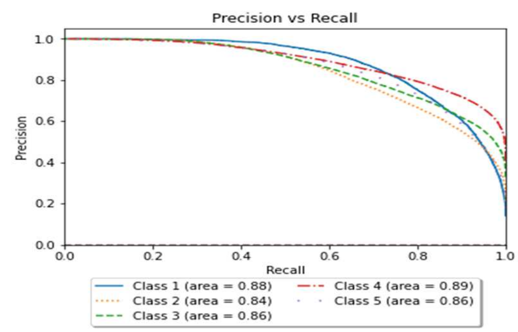

FIG.10. PRECISION-RECALL FOR JESTER DATASET USING CANNBCF

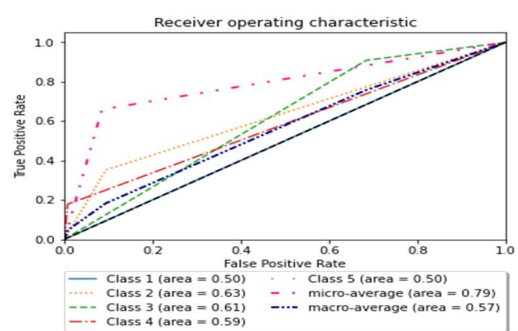

FIG.11. ROC FOR LAST.FM DATASET USING NBCF

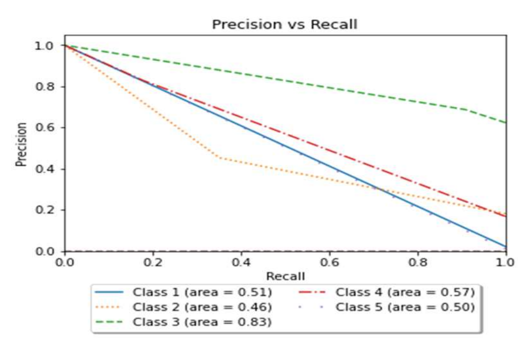

FIG.12. PRECISION-RECALL FOR LAST.FM DATASET USING NBCF

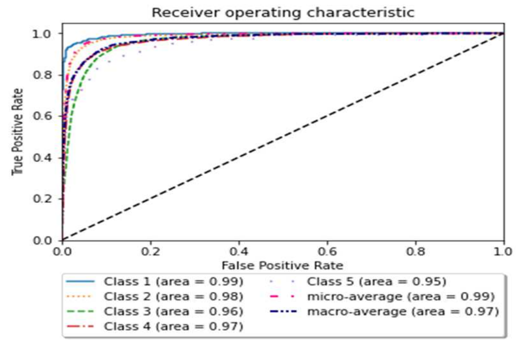

FIG.13. ROC FOR LAST.FM DATASET USING CANNBCF

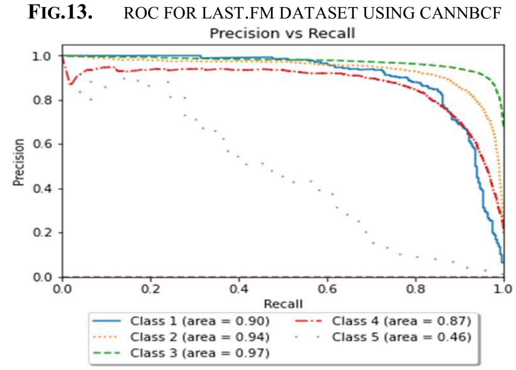

FIG.14. PRECISION-RECALL FOR LAST.FM DATASET USING CANNBCF

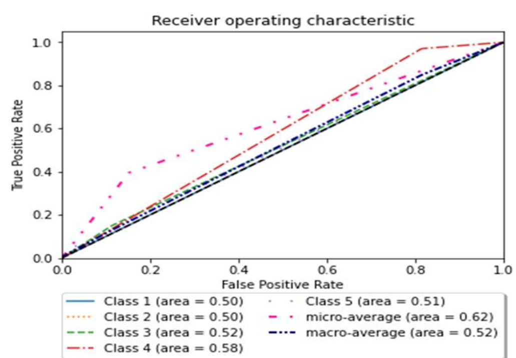

FIG.15. ROC FOR MOVIELENS DATASET USING NBCF

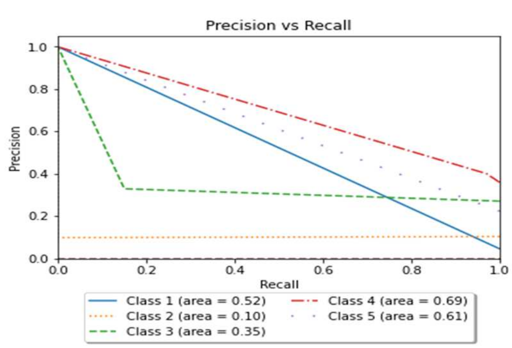

FIG.16. PRECISION-RECALL FOR MOVIELENS DATASET USING NBCF

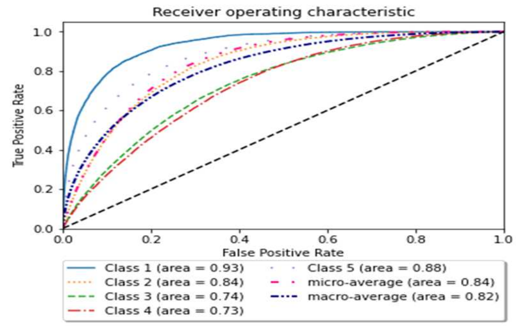

FIG.17. ROC FOR MOVIELENS DATASET USING CANNBCF

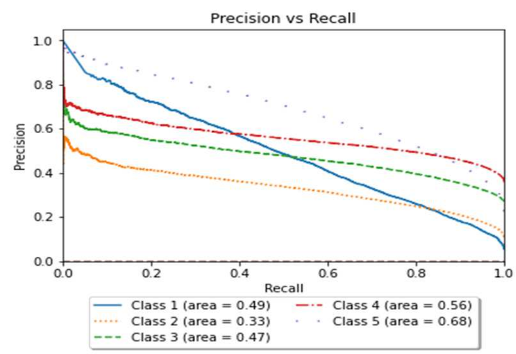

FIG.18. PRECISION-RECALL FOR MOVIELENS DATASET USING CANNBCF

VI. DISCUSSION

The use of an ANN allows learning the higher degrees of statistical inter-dependence between user-user and user-item features. The co-relation of the features captured by the neural network adds higher-order information as opposed to the collaborative filtering where we use first-order statistics such as averaging. In addition, we apply a dimensionality reduction algorithm before feeding the data into the neural network, which further alleviates the sparsity problem of collaborative filtering. The proposed approach outperforms the rating prediction accuracy of the NBCF algorithm by 24% on average. The accuracy increase for Book-Crossing, Jester, LastFM, and MovieLens datasets are 22%, 24%, 30%, and 20%, respectively. Figs. 19-22 demonstrate the value of the accuracy based on the different datasets for both NBCF and CANNBCF algorithms.

We sort the datasets based on the value of the accuracy from high to low and explain their characteristics accordingly:

(1) Last.fm includes 1826 users and 2384 items after the preprocessing steps. The sparsity level after the preprocessing steps is 98%. It has the highest imbalance ratio where 78% of

the ratings are 3, which leads to higher accuracy compared to the other datasets.

(2) Jester includes 13043 users and 47 items after the preprocessing steps. The sparsity level of this dataset is reduced from 56% to zero as most users were required to rate most jokes, and after filtering, those items and users who rated less than average are dropped. The ratio of users to items is relatively high, which helps to learn a more accurate model for this dataset compared to the Book-crossing and Movielens datasets.

(3) Book-crossing includes 10612 users and 21507 items after the preprocessing; therefore, the user to items ratio is 0.49. The ratio is relatively smaller than Last.fm and Jester, which makes it hard to predict the ratings accurately. In addition, the class frequency distribution is much more balanced compared to the Last.fm, which makes the prediction task more challenging. There is a 75% sparsity level, and the entropy of the class frequency distributions are 0.77 and 1.36 for Last.fm and Book-crossing datasets, respectively.

(4) Movielens includes 1887 users and 1212 items after the preprocessing steps. Similar to the Book-crossing dataset, it is harder to learn a model compared to Last.fm and Jester datasets. In Movielens, there is a 75% sparsity level. The entropy of the class frequency distribution for Movielens is 1.42, which is considered the highest among all other datasets. Thus, it is hard to predict the correct rating for all five classes.

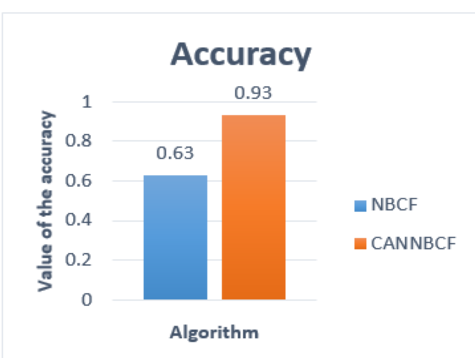

FIG.19. THE COMPARISON OF THE PREDICTIVE ACCURACY FOR LAST.FM DATASET

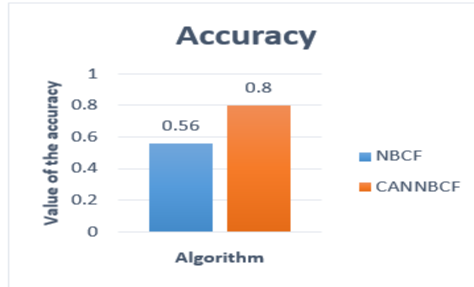

FIG.20. THE COMPARISON OF THE PREDICTIVE ACCURACY FOR JESTER DATASET

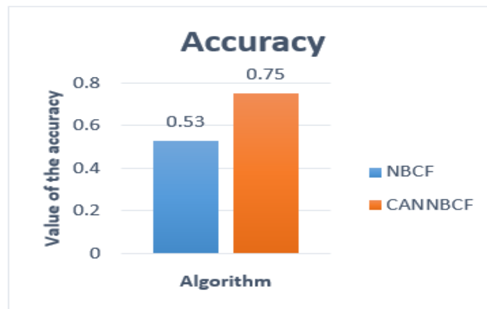

FIG.21. THE COMPARISON OF THE PREDICTIVE ACCURACY FOR BOOK-CROSSING DATASET

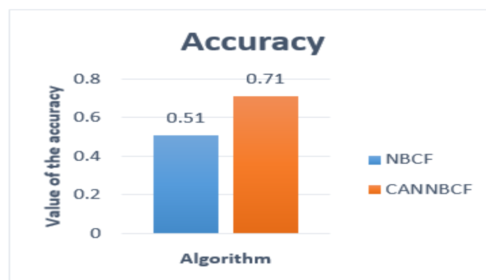

FIG.22. THE COMPARISON OF THE PREDICTIVE ACCURACY FOR MOVIELENS DATASET

VII. CONCLUSION AND FUTURE WORK

In this paper, we address the sparsity problem and aim to improve the accuracy of the recommendation in CF. We propose a hybrid model using ANN and clustering techniques. Our proposed model shows that accuracy, precision, recall, F-score, and ROC are improved when using a dimensionality reduction algorithm before feeding the data into the neural network, which further alleviates the sparsity problem of CF. The effectiveness of CANNBCF is experimentally examined using four different datasets collected from popular domains. We conduct eight experiments to examine the performance of the two algorithms NBCF and CANNBCF. The experiments indicate that our proposed algorithm can solve the sparsity problem and attain significantly better recommendation quality than NBCF. Even though the classes of the datasets are imbalanced, the proposed algorithm is not biased towards the frequent classes.

Future research can improve the quality of the CANNBCF by determining how to use incremental learning approaches ultimately applied to both extremely sparse datasets and other datasets. Moreover, the volume of the datasets that contain user-item interactions continues to increase over time. Hence, considering the scalability problem while addressing the sparsity problem is an open issue in CF and RSs that researchers can address in CF and RSs.

ACKNOWLEDGEMENT

The presented study is supported by Taif University (Taif, Saudi Arabia), expressed as a scholarship to the first author.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to Recommender Systems Handbook,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2015, pp. 1–34.

[2] N. J. Belkin and W. B. Croft, “Information Filtering and Information Retrieval: Two Sides of the Same Coin,” *Communications of the Acm*, vol. 35, no. 12, pp. 29–38, 1992.

[3] C. C. Aggarwal, *Recommender Systems*. Cham: Springer International Publishing, 2016.

[4] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decision Support Systems*, vol. 74, pp. 12–32, Jun. 2015.

[5] P. Melville and V. Sindhwani, “Recommender Systems,” in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2017, pp. 1056–1066.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural Collaborative Filtering,” in *Proceedings of the 26th International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland, 2017, pp. 173–182.

[7] A. Halavais, *Search engine society*, Second edition. Cambridge, UK Medford, MA, USA: Polity, 2018.

[8] U. Shardanand and P. Maes, “Social Information Filtering: Algorithms for Automating ‘Word of Mouth,’” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1995, pp. 210–217.

[9] Amatriain, X., Jaimes, A., Oliver, N., & Pujol, J. M., “Data Mining Methods for Recommender Systems” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2015, pp. 227–262.

[10] T. Ma *et al.*, “Social Network and Tag Sources Based Augmenting Collaborative Recommender System,” *IEICE Trans. Inf. & Syst.*, vol. E98.D, no. 4, pp. 902–910, 2015.

[11] X. Ning, C. Desrosiers, and G. Karypis, “A Comprehensive Survey of Neighborhood-Based Recommendation Methods,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA: Springer US, 2015, pp. 37–76.

[12] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, “Learning to Hash for Indexing Big Data—A Survey,” *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016, doi: 10.1109/JPROC.2015.2487976.

[13] M. K. Najafabadi and M. N. Mahrin, “A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback,” *Artif Intell Rev*, vol. 45, no. 2, pp. 167–201, Feb. 2016.

[14] M. K. Najafabadi, A. Hj. Mohamed, and M. N. Mahrin, “A survey on data mining techniques in recommender systems,” *Soft Comput*, vol. 23, no. 2, pp. 627–654, Jan. 2019.

[15] A. Althbiti and X. Ma, “Collaborative Filtering,” in *Encyclopedia of Big Data*, L. A. Schintler and C. L. McNeely, Eds. Cham: Springer International Publishing, 2019, pp. 1–4.

[16] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, “Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data,” *Expert Systems with Applications*, vol. 149, p. 113248, Jul. 2020, doi: 10.1016/j.eswa.2020.113248.

[17] Y. Du, C. Yao, S. Huo, and J. Liu, “A new item-based deep network structure using a restricted Boltzmann machine for collaborative filtering,” *Frontiers Inf Technol Electronic Eng*, vol. 18, no. 5, pp. 658–666, May 2017, doi: 10.1631/FITEE.1601732.

[18] Z. Huang, H. Chen, and D. Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, Jan. 2004, doi: 10.1145/963770.963775.

[19] X. Yang, S. Zhou, and M. Cao, “An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews,” *Mobile Netw Appl*, vol. 25, no. 2, pp. 376–390, Apr. 2020, doi: 10.1007/s11036-019-01246-2.

[20] D. Jannach, Ed., *Recommender systems: an introduction*. New York: Cambridge University Press, 2011.

[21] L. Huang, W. Tan, and Y. Sun, “Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis,” *Multimed Tools Appl*, vol. 78, no. 7, pp. 8711–8722, Apr. 2019.

[22] R. Bell, Y. Koren, and C. Volinsky, “Modeling relationships at multiple scales to improve accuracy of large recommender systems,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’07*, San Jose, California, USA, 2007, p. 95.

[23] J. Lee, H.-J. Kim, and S. Lee, “Conceptual collaborative filtering recommendation: A probabilistic learning approach,” *Neurocomputing*, vol. 73, no. 13–15, pp. 2793–2796, Aug. 2010, doi: 10.1016/j.neucom.2010.04.005.

[24] B. Ait Hammou and A. Ait Lahcen, “FRAIPA: A fast recommendation approach with improved prediction accuracy,” *Expert Systems with Applications*, vol. 87, pp. 90–97, Nov. 2017, doi: 10.1016/j.eswa.2017.06.001.

[25] I. Fernández-Tobías, I. Cantador, P. Tomeo, V. W. Anelli, and T. Di Noia, “Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization,” *User Model User-Adap Inter*, vol. 29, no. 2, pp. 443–486, Apr. 2019, doi: 10.1007/s11257-018-9217-6.

[26] P. K. Singh, P. K. D. Pramanik, and P. Choudhury, “An improved similarity calculation method for collaborative filtering- based recommendation, considering neighbor’s liking and disliking of categorical attributes of items,” *Journal of Information and Optimization Sciences*, vol. 40, no. 2, pp. 397–412, Feb. 2019, doi: 10.1080/02522667.2019.1580881.

[27] A. Althbiti and X. Ma, “Machine Learning,” in *Encyclopedia of Big Data*, L. A. Schintler and C. L. McNeely, Eds. Cham: Springer International Publishing, 2020, pp. 1–5.