BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0035

An Improved Product Recommender System Using Collaborative Filtering and a Comparative Study of ML Algorithms

S. Amutha, Vikram Surya R.

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
E-mails: amutha.s@vit.ac.in vikramsurya.r2019@vitstudent.ac.in

Abstract**:** *One of the methods most frequently used to recommend films is collaborative filtering. We examine the potential of collaborative filtering in our paper's discussion of product suggestions. In addition to utilizing collaborative filtering in a new application, the proposed system will present a better technique that focuses especially on resolving the cold start issue. The suggested system will compute similarity using the Pearson Correlation Coefficient* (*PCC*)*. Collaborative filtering that uses PCC suffers from the cold start problem or a lack of information on new users to generate useful recommendations. The proposed system solves the issue of cold start by gauging each new user by certain arbitrary parameters and recommending based on the choices of other users in that demographic. The proposed system also solves the issue of users' reluctance to provide ratings by implementing a keyword-based perception system that will aid users in finding the right product for them.*

Keywords**:** *Recommender system, Collaborative Filtering* (*CF*)*, Pearson Correlation Coefficient* (*PCC*)*, Cold start problem, Data sparsity.*

1. Introduction

To tackle data overload, recommender systems have become essential for offering personalized suggestions in movies, music, books, news, and web pages. Over the past decade, researchers have developed multiple algorithms and systems utilized by online platforms like Amazon.com, Netflix.com, and IMDb.com. The randomness of vast data makes it challenging for users to choose based on their preferences. Addressing this, relevant recommendations play a crucial role [6]. Both service providers and users benefit mutually from such systems. Service providers gain recognition for their products, while users save time in finding items and services of interest.

Collaborative Filtering (CF) is widely used in recommender systems for its transparency and accuracy [2]. It's a personalized method based on user ratings for specific items derived from user feedback. Ratings can be positive engagements, where a user clicks on an article, or explicit behavior, where a user rates an item.

CF involves two recommendation approaches: model-based, using item/user attributes, and memory-based, calculating the similarity between the active user and others in the rating system [1, 5]. Users are classified as neighbors of the active user, and recommendations are made based on their preferences. In our research, we have examined cosine similarity and the Pearson Correlation Coefficient (PCC) [10].

Despite success, CF faces challenges like the cold start problem, data sparsity, and scalability. Our work aims to enhance system accuracy. CF evaluates similarities between users or items with similar interests, demanding substantial computational analysis. The project's primary goal is to ease the cold start issue and data sparsity related to users and items. Conversely, items with a significant number of interactions will receive well-suited recommendations.

Objective:

- The main objective of this project is to overcome the cold start problem;
- The data sparsity problem is also addressed here;
- To enhance CF;
- To reduce both time and cost.

In this research paper, we address the cold start problem by recommending products based on the choices of other users within specific demographics or age groups. This approach suggests products popular among users, meeting those criteria [8]. Additionally, we have introduced a keyword search option to tackle the issue of data sparsity [3]. The research paper is segmented into the following sections: Section 1. Introduction (previously covered); Section 2. Notations; Section 3. Literature review; Section 4. Problem formulation and methodology; Section 5. Metrics; Section 6. Conclusion.

2. Notations

The goal of CF techniques, as a type of recommender framework, is to give people recommendations for useful products. These strategies deal with a set of users, $U = \{u_1, u_2, \ldots, u_m\}$, and a set of products, $I = \{i_1, i_2, \ldots, i_n\}$ in the most common case. Each user, $u_i \in U$, has a related profile comprising the subset of products he/she has rated, $I_u \subseteq I$, and the corresponding rating for each product. Furthermore, the subset of clients that have rated a specific product, $U_i \subseteq U$, is identified. Here u_a stands for the active user or the user for whom a prediction is being made. The ratings typically fall into whole numbers within a specific range, with the set of possible ratings being $R = \{0, 1, 2, 3, 4, 5\}$. Utilizing client profiles, the rating matrix V represents the client ratings. Each element of V, $v_{ui} \in R \cup \varphi$, indicates the rating given by user $u \in U$ to product $i \in I$, where the value φ denotes that the user has not rated the item yet. The primary objective of the CF Algorithm is to forecast the estimation of v in these situations. Let $p_{ui} \in R \cup \varphi$ denote the prediction the algorithm makes for the rating of item $i \in I$ by client $u \in U$. In case the calculation cannot make this prediction we accept, $p_{ui} = \varphi$. Finally, we define the subset of client ratings, $v_{u.} = \{v_{ui} \in V \mid i \in I_u\}$, and the subset of product evaluations, $v_i = \{v_{ui} \in V \mid u \in U_i\}$. We denote the client mean rating as v_u, and the product mean rating as $v_{.i}$.

3. Literature review

CF is one of the most famous strategies utilized in building intelligent recommender systems, allowing them to improve recommendations as more information about the items is collected. However, there has been a significant reduction nowadays due to drawbacks such as the cold start problem and data sparsity.

Related works and drawbacks of existing model. In CF, we recommend items or products based on how similar clients liked the item. For example, let us consider two users: A and B. They both share similar interests in movies. User A liked the movie "Avengers: Endgame". While user B has not watched the movie yet, the framework suggests this movie to B because it has identified their similar tastes. Besides user likeness, recommender systems can also utilize collaborative filtering based on item similarity. One of the most commonly used measures of similarity is cosine similarity. This method computes the cosine of the angle between two vectors. To determine the cosine similarity between items, we use the following formula:

$$\cos\theta = \frac{\mathbf{A}.\mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}}. \tag{1}$$

For our model, vector **A**, i.e., the user, remains constant. Then, we loop through all the products, considering each product as vector **B** to calculate the cosine similarity of each product to our profile. We can also conceptualize these vectors in a 3D space. Essentially, the dot product of two vectors divided by the product of each vector's magnitude is known as Cosine similarity. We divide the dot product by the magnitude as we are solely measuring the angle difference. Since the cosine of an angle of 0 degrees equals one, the cosine similarity increases as the values get closer to one. The product that most closely matches the user profile will be suggested using cosine similarity. The item-based CF approach does not take into account user ratings when using the cosine similarity metric. Adjusted cosine similarity overcomes this flaw by deducting the average rating of each respective user from each pair of ratings, and is defined as

$$\mathrm{sim}_{(m,n)}^{\mathrm{Acos}} = \frac{\sum_{i\in I}(r_{m,i}-\overline{r}_i)}{\sqrt{\sum_{i\in I}(r_{m,i}-\overline{r}_i)^2}\sqrt{\sum_{i\in I}(r_{n,i}-\overline{r}_i)^2}}, \tag{2}$$

where I represent the items commonly rated by users m and n, and $r_{m,i}$ and $r_{n,i}$ are the ratings given to the item i by m and n, respectively. Finally, r_m and r_n represent the average ratings given by m and n, respectively. There are numerous ways to measure comparability. One measure extensively used in data mining, particularly in CF, is Pearson's Correlation Coefficient (PCC). The PCC Algorithm is a type of memory-based CF Algorithm. PCC reflects the degree of linear connection between two factors, showing the extent to which, the factors are linked, ranging from +1 to –1. A correlation of +1 implies a perfect positive linear connection between factors or, in other words, two users with very similar tastes, while a negative relationship indicates that the users have dissimilar tastes. Instead of considering the distance between feature vectors as a method to assess similarity, we can consider the relationship between the critics' scores. The cross-analysis of the products is performed by the active user m and another user n using the PCC. It then computes the statistical correlation between the two active users. By using the PCC, the similarity between the two users/products is defined through the formula given below:

(3) $$\mathrm{sim}^{\mathrm{PCC}}_{(m,n)}=\frac{\sum_{i\in I}(r_{m,i}-\bar{r}_m)(r_{n,i}-\bar{r}_n)}{\sqrt{\sum_{i\in I}(r_{m,i}-\bar{r}_i)^2}\sqrt{\sum_{i\in I}(r_{n,i}-\bar{r}_i)^2}},$$

where I is the items commonly rated by users m and n, $r_{m,i}$ and $r_{n,i}$ are ratings given to the item i by m and n respectively. Finally, r_m and r_n are the average ratings given by m and n, respectively.

In the existing model, such as CF, we encounter drawbacks such as:

- Cannot handle new products

The dot product of the model's vectors represents the expectation for a specific client/product pair. Consequently, if an item is not observed during testing, the system cannot incorporate it and is unable to utilize it to challenge the model. This scenario is commonly known as the cold-start problem. Fig. 3 illustrates this issue while suggesting a new dress. In [19] authors have proposed a user-based CF recommendation mechanism on Hadoop, utilizing a CF mechanism to segment users into groups.

- Data sparsity

Most recommender systems only allow users to rate a tiny subset of the accessible items, leaving the majority of the rating network's cells empty. Finding similarities between different customers or items in such situations is exceedingly difficult.

4. Problem formulation and methodology

In the proposed model, we aim to address the drawbacks of the existing model, such as the cold start problem and data sparsity illustrated in Fig. 1.

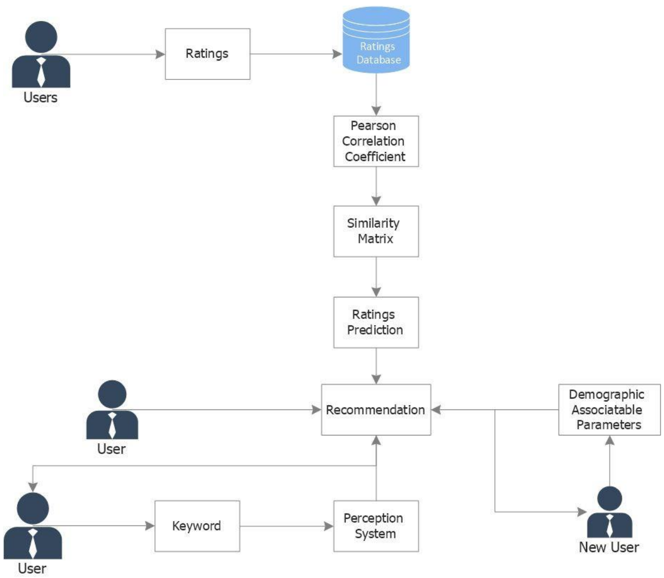

Fig. 1. System architecture

When new users join the website, the database lacks any purchase records related to the new user profile, resulting in the cold-start problem. Consequently, the recommender system fails to provide product recommendations to the user. To resolve this issue, the system recommends the most popular products based on the user's gender and age group. Here, the system filters the products in the dataset according to the user's provided gender and age, recommending the most popular products within this group.

In an effort to resolve the cold start problem a hybrid recommendation method is proposed, based on the profile expansion technique. Here, we utilize user demographic information to expand the user pool within the neighborhood. The introduction of a few new features enhances user-rating profiles. Addressing issues related to the cold start, the proposed rating profile expansion technique improves system performance. This is achieved by creating a denser user-item rating matrix that includes additional ratings [11].

Applying the bipartite network projection, we implement the Community Detection Algorithm in this scenario. The Louvain Algorithm is utilized to perform community detection on the projected One-mode network, updating the original item set with new items and subsequently recommending appropriate items to the user group. The algorithm's implementation in the recommendation system assures greater community confidence due to its improved accuracy, strong stability, and fast execution times [18].

A brand-new CF ranking model is suggested that combines a pairwise ranking-oriented method of Bayesian personalized ranking with a rating-oriented approach of Probabilistic matrix factorization to address theproblem of sparse data. Both implicit and explicit data are used in this model [12].

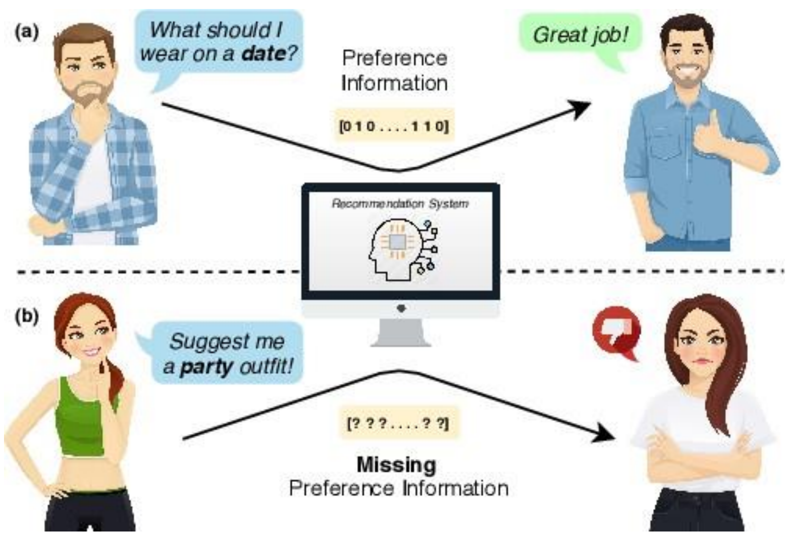

Fig. 2. Illustration of the cold-start issue in recommending dress for new users: (a) represents a profiled user, whose preferences are already presentin the system; (b) represents a new user with no preference records or history [15]

A novel top-N interactive recommender device is advocated in another model, based on deep reinforcement learning. The interactions between the agent and users are mimicked by the recurrent neural community in our version, which views the processes of recommendation as Markov selection techniques. This model can be applied on cold startand data sparsity scenarios to improve performance [13].

	userID	productID	ratings	timestamp
0	AKM1MP6P0OYPR	132793040	5	1365811200
1	A2CX7LUOHB2NDG	321732944	5	1341100800
2	A1GI0U4ZRJA8WN	439886341	1	1334707200
3	A1QGNMC6O1VW39	511189877	5	1397433600
4	A3J3BRHTDRFJ2G	511189877	2	1397433600

Fig. 3. Ratings of products by users

We can fix the cold start and data sparsity issues with Linked open data as well. The Matrix factorization model with Linked open data is used to address the problem of data sparsity, while the recommender system with Linked open data model can be used to address the problem of cold start. We update the matrix factorization model to address datasparsity by utilizing the information about new entities found in the Linked open data knowledge base “DBpedia” to solve a cold start problem [14].

4.1. Advantages

1. Cold start problem resolution. The model effectively addresses the cold start problem by recommending popular products based on user demographics and expanding user profiles using demographic information. This helps new users get relevant recommendations even when there is no historical data.

2. Data sparsity mitigation. The use of hybrid recommendation methods, Linked open data, and collaborative filtering helps alleviate data sparsity issues. These techniques enhance the user-item rating matrix by introducing new features or leveraging external knowledge bases, making recommendations more reliable, even with limited user ratings.

3. Community detection. The application of the Louvain Algorithm for community detection can improve recommendation accuracy by grouping similar users and items. This can lead to more personalized recommendations, enhancing the user experience.

4. Combination of techniques. The model combines different recommendation techniques, such as Bayesian personalized ranking, Probabilistic matrix factorization, and CF, which can lead to more robust and accurate recommendations compared to using a single method.

5. Keyword search feature. The inclusion of a keyword search feature provides users with more control over the recommendations. Users can input their preferences directly, making the system more user-centric.

6. Visualization of product clusters. The visualization of product clusters based on product descriptions can enhance user understanding and engagement. It helps users discover products related to their interests more effectively.

4.2. Disadvantages

1. Complexity. The proposed model appears to be quite complex, involving multiple recommendation techniques, data sources, and algorithms. This complexity can make it challenging to implement and maintain, requiring significant development and computational resources.

2. Data integration. Leveraging Linked open data and external knowledge bases like DBpedia can be beneficial, but it also requires continuous data integration and updates. Keeping these external sources up to date and aligned with the recommendation system can be resource-intensive.

3. Scalability. The model's performance might suffer when dealing with a large number of users and items. Managing the scalability of the hybrid recommendation method, collaborative filtering, and community detection algorithms can be a significant challenge.

4. Keyword search limitations. While the keyword search feature is user-friendly, it may not always provide the most accurate or personalized recommendations. It relies heavily on the quality of product descriptions and may not capture nuanced user preferences.

5. Algorithmic complexity. The algorithm used for predicting user ratings involves finding similar users and computing weighted averages, which can be computationally expensive for a large user-item matrix.

6. Algorithm evaluation. The model's effectiveness in addressing the cold start problem and data sparsity issues should be rigorously evaluated through experimentation. The actual performance in real-world scenarios may vary, and the proposed methods might not always provide optimal results.

In conclusion, while the proposed model offers several advantages in terms of addressing cold start and data sparsity issues, it also presents challenges related to complexity, scalability, and the need for ongoing data management. Careful implementation and evaluation are necessary to determine its suitability for specific use cases and ensure that it delivers the expected benefits to users and businesses.

4.3. Experimental setup model

1. Introduction of Collaborative Filtering for Product Recommendations. In this paper, we explore the application of CF, a commonly used method for recommending films, in the context of product recommendations. Our adaptation represents a novel approach to utilizing CF techniques.

2. Addressing the Cold Start Problem. We focus on resolving the Cold start problem, which occurs when new users join the platform, and there is no purchase history to base recommendations on. To tackle this issue, we propose recommending popular products based on the user's gender and age group, effectively providing recommendations even for new users.

3. Profile Expansion Technique. We introduce a profile expansion technique that leverages user demographic information to enhance user pools in the neighborhood. By creating a denser user-item rating matrix, this method improves recommendation accuracy, especially when dealing with cold start issues.

4. Hybrid Recommendation Model. We propose a hybrid recommendation model that combines a pairwise ranking-oriented method with a rating-oriented approach to address the problem of sparse data. This model effectively uses both implicit and explicit data to make recommendations.

5. Keyword-Based Product Recommendations. To combat data sparsity and enhance user experience, we introduce a keyword search feature. Users can input keywords, and our system employs CF methods to display clusters of products based on descriptions, effectively helping users find relevant products based on their search queries.

6. Visual Representation of Product Clusters. Our paper includes visual representations, such as Fig. 4, which illustrate how product clusters are formed based on product descriptions, providing a clear visual understanding of the recommendation process.

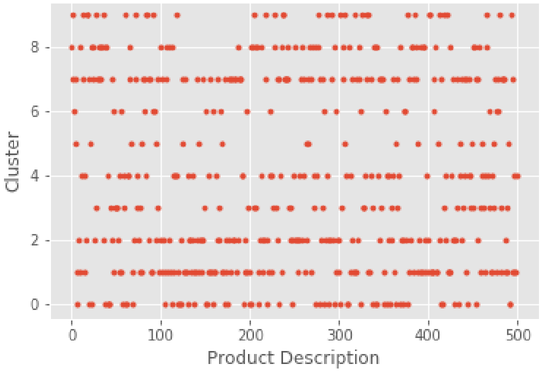

Fig. 4. Visualization of product cluster based on product description

4.4. Algorithm

In the proposed model, the algorithm used to predict the user rating is given below.

1. We have an $n \times m$ framework comprising the ratings of n clients and m products. Each element of the matrix (i, j) represents how client i rated product j as given in Fig. 4.

Since we are dealing with product ratings, each evaluation is expected to be a wholenumber from 1-5 (reflecting one-star evaluations to five-star evaluations) if client i has evaluated product j, and 0 if the client has not evaluated that specific product.

2. To suggest a set of products that each client has not yet used, we will effectively employ a methodology similar to weighted k-nearest neighbors.

3. For each product j that client i has not used yet, we find the set of clients "u" who are similar to client i and have used product j. For each similar client "u", we utilize the formula to determine the correlation with various products in the matrix.

Finally, we rank the products by their weighted average ratings. These average ratings serve as an estimate of what the client might rate each product. Products with higher average ratings are more likely to be favored by the client; thus, the system will recommend the products with the highest average rankings to the client.

Using collaborative filtering introduces another problem called data sparsity. This issue arises when users provide ratings for fewer products or when a company is in its early stages, resulting in a lower amount of user ratings. To address this challenge, we can introduce a keyword search feature that leverages various collaborative filtering methods to recommend products. With this feature, customers can input any keyword into the search bar. Subsequently, the collaborative filtering method generates different clusters containing various products. Once a cluster is identified based on the user's input word, the recommendation system presents items from the product clusters, using the product descriptions. In case a search word appears in multiple clusters, the algorithm selects the cluster where the word appears most frequently. Fig. 4 provides a visualization of the product clusters based on product descriptions.

In summary, the contributions encompass addressing common challenges in recommendation systems, including the cold start problem and data sparsity, while proposing innovative solutions, algorithms, and hybrid models to enhance the effectiveness of collaborative filtering for product recommendations. Additionally, the introduction of a keyword-based search feature further enriches the recommendation system's capabilities.

In this research, several models and techniques are employed to enhance the effectiveness of product recommendation systems. Collaborative filtering serves as the foundational approach for generating personalized recommendations. The paper introduces a profile expansion technique to enrich user profiles with demographic information and improve recommendation accuracy. Additionally, a hybrid recommendation model combines Bayesian personalized ranking and Probabilistic matrix factorization to address data sparsity issues. Finally, a keyword-based recommendation feature is introduced, enabling users to find products based on their input keywords. Together, these models and techniques collectively contribute to a robust and versatile product recommendation system.

5. Metrics

- **Mean Absolute Error (MAE).** The mean absolute error has been utilized to gauge the contrast between the forecast of the calculation and the genuine rating. It is figured over all the ratings accessible in the assessment dataset, utilizing the formula:

(4) $$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}\left|d_i - \hat{d}_i\right|,$$

where n is the number of data points, d_i is the value of data/ input provided, $\hat{d}_i$ is the average/mean of the data/input provided.

Notwithstanding its constraints, while assessing frameworks zeroed in on suggesting just a specific numberof things, however because of the effortlessness of its count and its measurable properties this measurement is considered as one of the most well-known while assessingrecommender frameworks.

- **Root Mean Squared Error (RMSE)** is similar to the MAE, here it places more prominent accentuation on bigger blunders by using the next equation. RMSE of a model is a metric that measures how much the signal and the noise is clarified by the model.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \hat{d}_i)^2}, \quad (5)$$

where n is the number of data points, d_i is the value of data/ input provided, $\hat{d}_i$ is the average/mean of the data/input provided.

The analysis of RMSE and MAE with various models is tabulated in Table 1 and illustrated in Fig. 5. These measurements have been utilized because errors in these metrics have a significant impact on user decision-making. For instance, on a 5-point scale, a 1-point error might not be noticed by the customer (considering products rated only 4 or 5 points as good suggestions), while a 4-point error on a product could indicate that the item is unacceptable.

Table 1. RMSE and MAE of different models

Model/Accuracy	Popularity recommeder model	KNN with means	SVD	Parameter tuned SVD
MSE	2.2198	1.1719	1.1078	0.853
RMSE	1.4899	1.0826	1.0525	0.9236

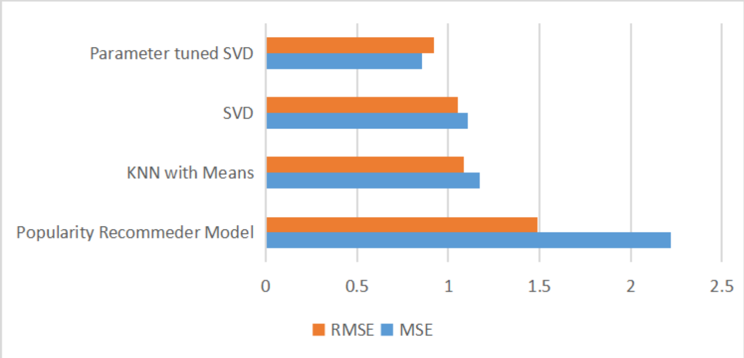

Fig. 5. Graphical representation of RMSE and MAE

6. Conclusion

In this paper, we have initially evaluated the drawbacks of current similarity techniques in CF. Subsequently, we propose an enhanced similarity estimation method based on the Pearson correlation coefficient to overcome the current limitations of similarity measurement strategies in CF. Various algorithms have been introduced in this paper to address the issues of CF, such as the cold start problem and data sparsity. To address the cold start problem, we assess each new user based on certain arbitrary parameters and recommend products based on the choices of other users in the same demographic. Additionally, we have included a keyword search feature to recommend products, resolving the issue of data sparsity. For future research directions, we intend to utilize deep neural networks as a predictive model instead of the Pearson correlation coefficient and cosine similarity methods, aiming to enhance the predictive capability of our approach. Moreover, we are exploring the use of other natural language processing tools to develop user attribute matrices and compare their performance with the current framework.

***Acknowledgments*:** We as a team would like to pay our sincere gratitude to our obliging, cooperative, and kind-hearted teachers, and our University, Vellore Institute of Technology – Chennai, for providing us with a nurturing environment to bring out the best in us.

References

1. Maleki Shoja, B., N. Tabrizi. Customer Reviews Analysis with Deep Neural Networks for e-Commerce Recommender Systems. – In: IEEE Access, Vol. **7**, 2019, pp. 119121-119130. DOI: 10.1109/ACCESS.2019.2937518.
2. Iftikhar, A., M. A. Ghazanfar, M. Ayub, Z. Mehmood, M. Maqsood. An Improved Product Recommendation Method for Collaborative Filtering. – IEEE Access, Vol. **8**, 2020, pp. 123841-123857. DOI: 10.1109/ACCESS.2020.3005953.
3. Won, H., Y. Lee, J. Shim, H. Ahn. A Hybrid Collaborative Filtering Model Using Customer Search Keyword Data for Product Recommendation. – In: Proc. of 18th IEEE International Conference on Machine Learning and Applications (ICMLA'19), Boca Raton, FL, USA, 2019, pp. 1523- 1526. DOI: 10.1109/ICMLA.2019.00251.
4. Shapira, B. Recommender Systems Handbook. USA, Springer, 2015.
5. Xue, H.-J., X. Dai, J. Zhang, S. Huang, J. Chen. Deep Matrix Factorization Models for Recommender Systems. – In: Proc. of 26th International Joint Conference on Artificial Intelligence (IJCAI'17), 2017, pp. 3203-3209. DOI: 10.24963/ijcai.2017/447.
6. Cacheda, F., V. Carneiro, D. Fernandez, V. Formoso. Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems. TWEB. 5. 2, 2011.
7. Sony, H., G. Poonam. Survey Paper on Recommendation System Using Data Mining Techniques. – International Journal of Engineering and Computer Science, 2016. DOI: 10.18535/ijecs/v5i5.60.
8. Yong, S., A. Yasuhito. Mining Relationship Between User Purposes and Product Features Towards Purpose-Oriented Recommendation. 2017, pp. 9-21. DOI: 10.1007/978-3-319-61845-6-2.
9. Fatoni, C., E. Utami, F. Wibowo. Online Store Product Recommendation System Uses Apriori Method. – Journal of Physics: Conference Series, 2018. 1140. 012034. 10.1088/1742-6596/1140/1/012034.
10. Priyanka, M. J., A. Shetty, D. Vora. Web Usage Mining Using Pearson's Correlation Coefficient. – International Journal of Engineering Research and Applications (IJERA), Vol. **3**, March-April 2013, Issue 2, pp. 676-679. ISSN: 2248-9622. **www.ijera.com**
11. Tahmasebi, F., M. Meghdadi, S. Ahmadian, et al. A Hybrid Recommendation System Based on Profile Expansion Technique to Alleviate Cold Start Problem. – Multimed Tools Appl., Vol. **80**, 2021, pp. 2339-2354.
12. Feng, J., Z. Xia, X. Feng, J. Peng. RBPR: A Hybrid Model for the New User Cold Start Problem in Recommender Systems. – Knowledge-Based Systems, Vol. **214**, 2021, 106732. ISSN: 0950- 7051.
13. Huang, L., M. Fu, F. Li, H. Qu, Y. Liu, W. Chen. A Deep Reinforcement Learning Based Long-Term Recommender System. – Knowledge-Based Systems, Vol. **213**, 2021, 106706. ISSN: 0950-7051.
14. Natarajan, S., S. Vairavasundaram, S. Natarajan, 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**, 2020, 113248. ISSN: 0957-4174.
15. Verma, D., K. Gulati, R. R. Shah. Addressing the Cold-Start Problem in Outfit Recommendation Using Visual Preference Modelling. – In: Proc. of 6th IEEE International Conference on Multimedia Big Data (BigMM'20), 2020, pp. 251-256. DOI: 10.1109/BigMM50055.2020.00043.

16. A Combined Approach for Collaborative Filtering Based Recommender Systems with Matrix Factorisation and Outlier Detection. – Journal of Business Analytics, Vol. **4**, July 2021, No 19, 1-14. DOI:10.1080/2573234X.2021.1947752.
17. B a n s a l, P., S. S. G i l l. Optimal High Pass FIR Filter Based on Adaptive Systematic Cuckoo Search Algorithm. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 4, pp. 167-177.
18. S h a m b o u r, Q. Y., N. M. T u r a b, O. Y. A d w a n. An Effective e-Commerce Recommender System Based on Trust and Semantic Information. – Cybernetics and Information Technologies, Vol. **21**, 2021, No 1, pp. 103-118.
19. P r e m a l a t h a, M., V. V i s w a n a t h a n. Course Sequence Recommendation with Course Difficulty Index Using Subset Sum Approximation Algorithms. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 3, pp. 25-44.

Received: 02.05.2023; Second Version:26.09.2023; Third Verion: 06.11.2023; Accepted: 17.11.2023