# Alleviating Sparsity and Cold-Start through Deep Learning-based Hybrid Recommender System

Nirupma Singh[1], Dr. Abid Hussain[2]

[1]Research Scholar, School of Engineering & Tech., Career Point University, Kota, India.

[2]Associate Professor, School of Computer Applications, Career Point University, Kota, India.
[1]nirupmachauhan27@gmail.com, [2]abid.hussain@cpur.edu.in

**Article Info**
**Page Number: 1876 – 1894**
**Publication Issue:**
**Vol. 71 No. 3s2 (2022)**

***Abstract***
In the digital world, recommender system has become a prominent solution for handling information overload and satisfying the users over the millions of available products or services. In the large-scale recommender system, determining the precise preferences of the users is a fundamental challenging task. Hybrid filtering methods have provided suggestions for the target users based on the preferences of similar users and metadata information of the products. However, resolving the cold-start user constraint through collaborative filtering is a critical task in the e-commerce site. In addition, utilizing the explicit feedback alone is inadequate to extract the preferences of the users and handle the data sparsity and cold-start constraints in the recommender systems. Several existing recommendation models have adopted the deep learning models to handle the data sparsity and suggest the desired products to the users. Despite this, ensuring the quality of the recommendation in e-commerce sites for the cold-start users and sparse data is challenging. Hence, this work focuses on addressing the data Sparsity and Cold-start with the developed deep learning model based Hybrid based Recommender (SC-HR) system. The SC-HR approach models the deep autoencoder for the data representation and hybrid filtering to handle the data sparsity in e-commerce sites. It utilizes the textual with non-textual feedback of the users on the products as well as the metadata of the products to generate consistent representations for the user-product relationships. The user-product relationship expansion with consistency creates a significant impact on the personalized recommendations. The SC-HR approach utilizes implicit feedback such as the click-rate information to resolve the cold-start user constraint when there is a lack of feedback on the products. By applying the deep autoencoder, the SC-HR approach transforms the high-dimensional representation to low-dimensional representation with the potential information from the learning of the user and product features. Finally, the SC-HR approach determines the user preferences for the consistent and complete representation of the user-product relations and suggests personalized products. Thus, the experimental results demonstrate that the SC-HR approach yields 84.2% precision in the Amazon product recommendation system.

**Article History**
**Article Received:** 22 April 2022
**Revised:** 10 May 2022
**Accepted:** 15 June 2022
**Publication:** 19 July 2022

**Keywords:** Hybrid Filtering, Deep Autoencoder, Product Recommendation, E-commerce, Sparsity, Cold-start, Click-rate, and Personalized Recommendation.

## 1. Introduction

With extensive access to e-commerce sites and Web technologies, people often suffer from the massive amount of digital information over the Web. Over the past decades, the recommender system [1] has provided a significant commercial value for the information overload problem by assisting the users in finding their relevant information within a reasonable time from the abundance of the data. The main objective behind the recommender system is to utilize the past preferences of the users to predict the interests of the users in the future [2]. In the Internet industries, Amazon, YouTube, and Netflix have been widely utilized the recommendation models to provide personalized recommendations for their users based on the profile of the past activities of the users in the corresponding sites. Collaborative filtering, content filtering, and hybrid filtering methods are the three main categories of the recommendation model. With the increased pervasiveness among the user-item interactions, collaborative filtering-based recommendation approaches [3] have gained significant attention in addressing the data sparsity over the content-based methods. However, traditional collaborative filtering methods have been encountered difficulties in improving the precision of the recommendation. In addition, collaborative filtering based recommendation often suffers from obtaining the profile information in the e-commerce sites due to the comparatively sparse profile information of the users than the social networks. Hence, it is essential to precisely extract the user preferences from the feedback of the users and available user and item features [4, 5].

In e-commerce sites, users provide a few ratings or reviews about the products, creating implications for the recommender systems. The data sparsity [6, 7] is an ever-increasing challenge due to the available ratings on the partial amount of products over the millions of products. Also, recommender systems often meet the cold-start user and cold-start item constraints while applying the collaborative and content-based filtering models. To resolve these shortcomings, the existing recommender systems have applied hybrid filtering models to suggest personalized recommendations. Despite this, the recommender system still faces data sparsity and cold-start constraints especially, in e-commerce sites. The conventional researchers have presented the different developments in the recommendation models by applying the deep neural networks [8, 9]. In recommender systems, autoencoders [10] have been widely used by researchers to handle the data sparsity with the learned knowledge of the user presentation and item representation. However, it is difficult to alleviate the data sparsity and cold-start user constraints in the product recommendation model. Thus, this work focuses on developing the recommendation model with the hybrid filtering and the deep learning models from the analysis of the textual and non-textual feedback as well as the metadata of the products. Moreover, the SC-HR approach analyzes the user's click rate to handle the cold-start user constraint during the preference extraction. Hence, it effectively handles the data sparsity and cold-start constraints even when there is a lack of profile information in the e-commerce sites.

The major contributions of the Sparsity and Cold-start-aware Hybrid Recommender (SC-HR) methodology are presented as follows.

- This work designs the hybrid recommender system using the deep learning model to suggest the user preferred products even when there is an inadequate amount of knowledge regarding the users and products.

- Applying the content-based recommendation procedure along with the analysis of the implicit feedback as the click-rate information overcomes the shortcomings in the collaborative recommendation model for the e-commerce site and improves the quality of the recommendations.
- In addition to the hybrid modeling, the SC-HR approach employs the deep autoencoder model to transform the high-dimensional data into the low-dimensional data to resolve the sparsity constraint.
- Moreover, the combination of the features from the hybrid approaches along with the unsupervised deep learning model facilitates the prediction of the user preferences on the products.
- Thus, the experimental results illustrate that the SC-HR approach outperforms the existing hybrid recommendation model with the improved quality of the personalized product recommendation.

## 2. Related Works

This section reviews the conventional researches in the recommendation system for the data sparsity constraint. In addition, it surveys several deep learning-based recommendation approaches.

### 2.1.Recommendation Approaches in Addressing Sparsity Constraint

The multi-type auxiliary implicit feedback-based recommendation model [11] addresses the data sparsity by generating the target data with the help of the nearest neighbors and linear regression model. Learning the correlations between the target and auxiliary feedback generates new target feedback and computes the confidence of the generated data using the ranking method to suggest the recommendations. Mobile Sparse Additive GEnerative (Mobi-SAGE) model [12] considers the user interests and the user privacy preferences for the categories to suggest the mobile applications to the users. Moreover, the Mobi-SAGE model jointly learns the visual content and textual content associated with the mobile application to resolve the data sparsity constraint during the recommendation. The research work [13] measures the reliability for the recommendation in three different perspectives, such as the user-based, item-based, and rating-based reliability measures. It enhances the users' rating profiles based on the reliable ratings in user-based reliability measures.

Moreover, it applies the item-based reliability measure and generates the target rating profile with the highest reliable items, which enforces the similarity computation between the users and unknown or new items and suggests the desired items to the users. The noise correction-based approach [14] handles the highly sparse environment in the recommender system by predicting the unrated items using noise-free data. Moreover, it determines the noise ratings by classifying the users and items into the strong, average, and weak classes. An enhanced data sparsity reduction model [15] applies the bi-separated clustering algorithm to build the bi-clusters of the rating matrix for the users and items with the assistance of the rating classification. By employing the bi-mean imputation algorithm, it imputes the missing ratings with the mean values in the bi-clusters and predicts the cold items for the new rating matrix through the collaborative filtering-based recommendation. The neighborhood reduction model [16] addresses the data sparsity and new user cold start problems by removing the redundant users from a set of neighbors of a new user with the assistance of the Covering Reduction Collaborative Filtering (CRCF) algorithm. It ignores the redundant elements for each new user while predicting the rating score for the unrated items and suggests the items with the highest rating score for a new user.

Fusion collaborative model based recommendation approach [17] presents the multi-factor similarity model along with the global rating information to capture the linear and non-linear relationship between the users. By integrating the local relations of the users with the global rating optimization, it handles the extremely sparse data and improves the robustness and prediction accuracy in the recommendation model. Burger recommendation model [18] resolves the data sparsity by utilizing the knowledge from the user preferences on similar items particularly, the pizza. It employs the variational autoencoder collaborative filtering in the cross-domain to improve the quality of the recommendation based on the user-item interactions and product of experts architecture. The hybrid recommendation model [19] applies the profile expansion technique and addresses the cold start issue with the consideration of the demographic information of the users along with the rating information. Moreover, it expands the rating profiles of the users using the strategies of the Global Most-Rated (GMR) and Global User-Local Clustering (GUC) to improve the performance of the recommender system. The electronic product recommendation model [20] utilizes contextual information through the sentiment analysis of the textual reviews of the users on products. By incorporating the knowledge from the sentiment score, it addresses the sparsity constraint in the scenario of the inadequate rating values on the products. However, it fails to resolve the sparsity constraint when there is the existence of new users and new products in the recommender system.

### 2.2.Recommendation Approaches Using Deep Learning

The deep hybrid recommendation model [21] integrates the stacked denoising autoencoders with neural collaborative filtering to learn the features of both the users and items. By exploring the implicit feedback of the users and items, it improves the preference predictions through the Generalized Matrix Factorization (GMF++) and Multi-layer Perceptron (MLP++) learning models. The deep Collaborative Filtering (DeepCF) model [22] presents the Collaborative Filtering Network (CFNet) with the assistance of the vanilla MLP model. It employs representation learning and matching function learning to learn the low-rank user-item relations. Sequential Deep Matching (SDM) model [23] analyzes both the short and long-term behaviors of the users to design the large-scale recommender system. By applying the multi-head self-attention and gated fusion modules, it examines the short-term session behaviors and long-term preferences respectively to suggest the successive items for the users from their sequential behaviors.

The correlative Denoising Autoencoder (CoDAE) approach [24] explores the correlations among the multiple roles of the users to handle the sparse rating information in the social networks through robust representations. It separately utilizes three autoencoders with the regularization term for learning the three roles of user features involving the truster, trustee, and rater, respectively. The product recommendation model [25] employs the Capsule Networks (Caps) and Matrix Factorization (MF) to represent the document. It represents the textual description through Bi-directional Recurrent Neural Network (Bi-RNN) stacked with the Capsule network and thus, recommends the user preferred items with the help of the Probabilistic Matrix Factorization. Model-based Collaborate Filtering Algorithm based on Stacked AutoEncoder (MCFSAE) approach [26] handles the sparsity problem by transforming the rating matrix into the high-level low-dimensional representation equivalent to the number of ratings. Consequently, it predicts the unknown ratings for the users on the items by utilizing the high-level features in the softmax classification model.

## 3. Problem Statement

With the rapid emergence of numerous new products or services in the commercial market, recommender systems often confront data sparsity constraints. The user-based collaborative filtering model suffers from the new user cold-start and data sparsity constraints during the recommendation. In addition to the emergence of new products, suggesting personalized recommendations for the new users through predictions is impossible due to the inadequate amount of data for the cold-start users. In addition, the lack of utilizing the advantage of the beneficial relationships among the users and products leverages the inaccurate recommendations. Hybrid recommender systems fail to extract the potential representations only from the input knowledge over the high-dimensional data. Owing to the increased diversities among the users and the change of users' interests over time, determining the user preferences on a particular product is a challenging task. In addition, suggesting the desired products to the new users is arduous on the e-commerce site due to the lack of profile information of the user in the e-commerce sites. The lack of analyzing the implicit feedback in the e-commerce site fails to recommend the desired products to the cold-start users. Even though deep learning-based recommender systems extract the implicit contexts, the lack of analyzing the explicit contexts reduces the quality of the recommendations. Hence, it is essential to utilize the metadata to extract the user preferences for the recommendation even when there is sparse data.

## 4. Outline of the Proposed Model

This section describes the primary processes involved in the SC-HR approach, which are discussed as follows.

**Text Preprocessing and Relationship Extraction:** The SC-HR approach initially applies the natural language and text preprocessing procedure to preview text and the metadata description of the product. Moreover, it measures the similarity between the users and products from the users' ratings on the products and metadata of the products and generates the similarity score.

**Data Sparsity Reduction:** The SC-HR approach targets mitigating the sparsity level in the input user-product-rating data. To accomplish this, it utilizes the review information to determine the interests of the products and represents the input dataset with the help of the deep autoencoder.

- **Rating and Review-based Sparsity Handling:** To handle the sparsity in the recommender system, the SC-HR approach analyzes the sentiment polarity of the product reviews to reduce the sparsity level. It enriches the input user-product-rating data with the integration of the sentiment score of a particular user on the corresponding product to generate feedback in the form of the numerical score even when rating information is absent.

- **Deep Autoencoder-based Sparsity Handling:** In subsequence, the SC-HR approach represents the enriched data with the assistance of the deep autoencoder model to transform the low-level data representation into the high-level representation of the user interests in the products. By generating the normalized values of each user on the products from the knowledge of partial feedback information, it resolves the data sparsity in the recommender system.

**Resolving Cold-Start Item Constraint:** In the recommender system, the SC-HR approach handles the sparsity generated by the new users in addition to the sparsity created by the existing users or customers. Instead of only analyzing the explicit feedback of the ratings and reviews of the users on products, it utilizes the metadata of the products such as the product description, brand, title, and category to determine the similar products. Thus, the SC-HR approach computes the interest of the user on the new product, which ensures the addressing of the cold-start item constraint.

**Resolving Cold-Start User Constraint:** Instead of recommending the desired products to the existing users in the e-commerce site, the SC-HR approach suggests the desired products to the new users. By utilizing the click-rate score on the products by the new users, it determines the similar products and suggests them to the new users based on the product features. Thus, the SC-HR approach addresses the cold-start user constraint in the recommender system.

**Preference Extraction and Recommendation:** Finally, the SC-HR approach extracts the preferences of all the users on the existing and new products and suggests the desired products to the existing and new users.

Figure 1 illustrates the process of alleviating the data sparsity and cold-start constraints in the product recommender system.

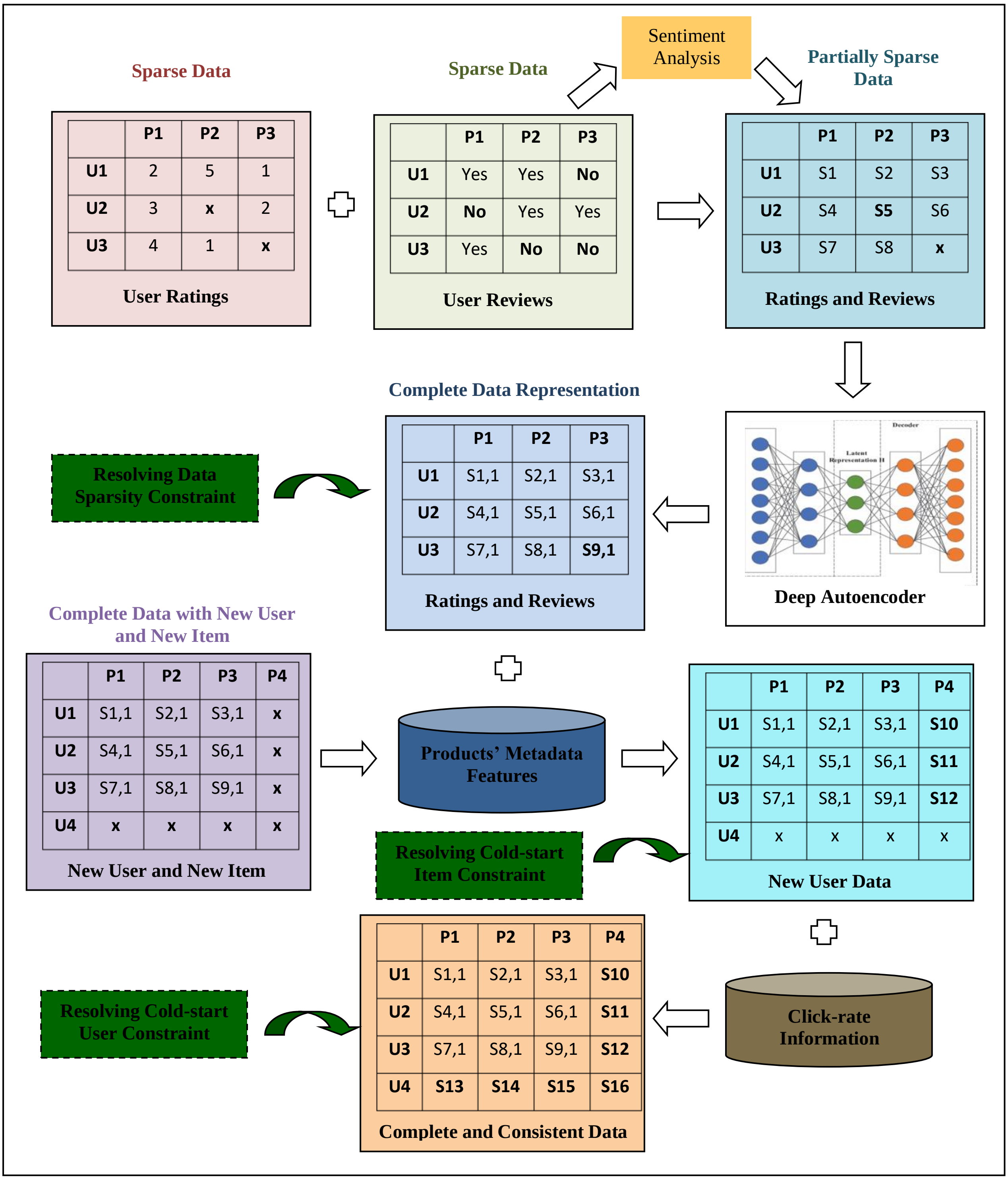


**Figure 1: Step-wise Data Transformation in the SC-HR Approach**

## 5. The Proposed SC-HR Methodology

The proposed hybrid recommender system is targeting to resolve the data sparsity constraint in the recommendation model using the deep learning algorithm in addition to the advantage of the hybrid recommendation model. As shown in Figure 2, the SC-HR approach utilizes information such as textual feedback such as reviews and non-textual feedback such as ratings or voters in the recommendation sites, as well as the content information of the products such as attributes to design the hybrid recommendation model. Even though the existing hybrid recommender systems utilize the content information for addressing the data sparsity in the collaborative filtering model, it confronts the lack of knowledge in the user-Product relationship patterns. In consequence, the learning model or decision-making model misguides the preference extraction or recommendation process even when there are the preferences of the users on the content-based relevant products. Hence, the SC-HR approach applies the unsupervised deep learning model such as the deep autoencoder model to transform the high-dimensional data into low-dimensional data with a high level of abstraction. With the help of the autoencoder, the SC-HR approach generates the potential representations for the input data, which plays a significant role in resolving the data sparsity issue and facilitates the recognition of the relevant product preferences alone. Moreover, the SC-HR approach performs the concept of the feature combination in the hybrid recommender system model, which jointly represents the user and product features from both the collaborative and content-based filtering models and jointly learns the features using the deep learning model suggesting the recommendations. Thus, the SC-HR approach precisely improves the quality of the recommendation even when there are new users and new products that create the cold-start problem.

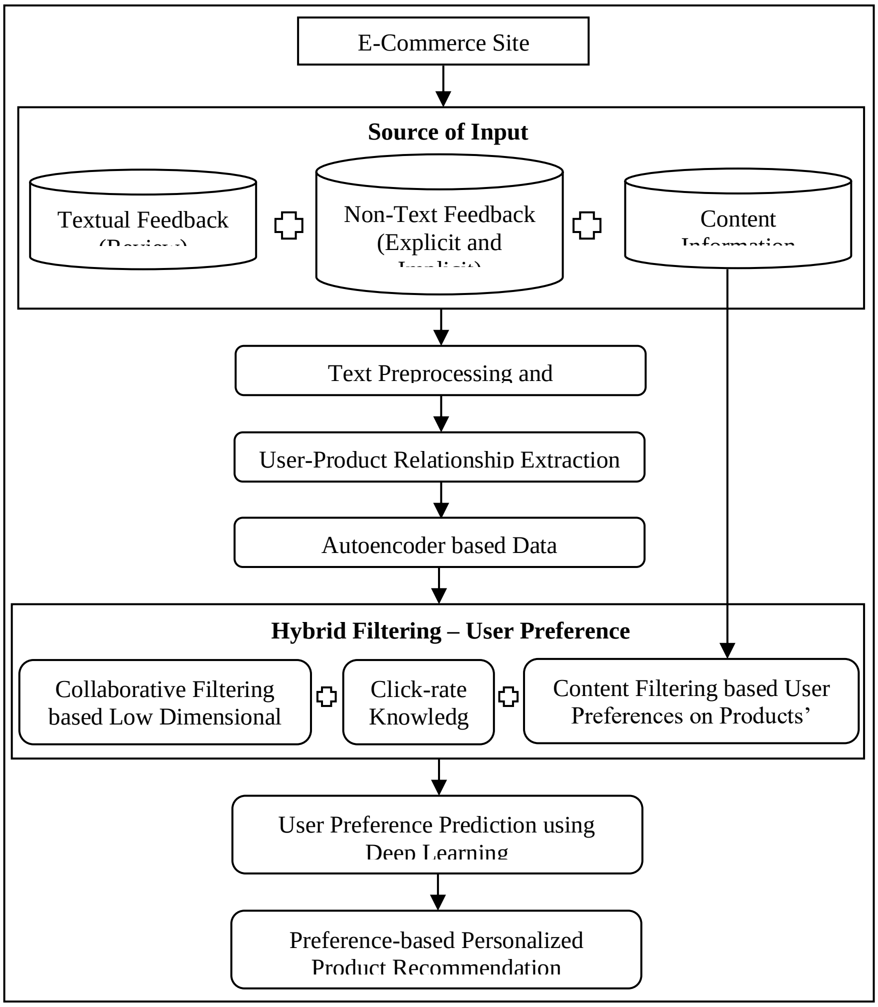


**Figure 2: Deep Learning-based Hybrid Recommender System**

### 5.1. Preprocessing the User and Product Content

Initially, the SC-HR approach applies the preprocessing steps on the textual and non-textual feedback provided by the users on the products on the e-commerce site. In the field of data mining, managing data inconsistencies through data preprocessing becomes a significant task. To handle the inconsistencies and improve the effectiveness of the data mining tasks, the SC-HR approach performs the data cleansing that removes the noisy data from the raw input dataset. The SC-HR approach applies textual preprocessing to review data involving noise removal, tokenization, stemming, lemmatization, Part-of-Speech (POS) tagging, and parsing. In essence, noise removal includes the dates, numerical values, special characters, URLs, and stop-words removal. Moreover, the SC-HR approach extracts the user-Product relationship from the users' ratings, reviews, and products' metadata. It applies the similarity models between the users and products for ratings and product features. Equation (1) computes the cosine similarity between the users 'i' and 'j' for the products rated by both the users. $\mathrm{R_{i,p}}$ and $\mathrm{R_{j,p}}$ denotes the rating value of $i^{th}$ and $j^{th}$ user on product 'p', respectively.

$$\mathrm{Sim(i,j)} = \frac{\sum_{p=1}^{P}\left(R_{i,p} \times R_{j,p}\right)}{\sqrt{\sum_{p=1}^{P}\left(R_{i,p}\right)^2 \times \sum_{p=1}^{P}\left(R_{j,p}\right)^2}} \quad (1)$$

Moreover, the SC-HR approach computes the similarity between the products based on the meta attributes to find the relevant products for each user. The product similarity is based on the weighted average of meta attributes of the product, such as the brand, type, price, and features, which is calculated using equation (2).

$$\mathrm{Sim(P1,P2)} = \frac{\sum_{p_a=1}^{A}\left(S(P_1,P_2)_{p_a}\right)^{i} \times w_{p_a}}{\sum_{p_a=1}^{A} w_{p_a}} \quad (2)$$

By applying Equation (2), the SC-HR approach computes the similarity score between the products (P1, P2) for the product attributes ($p_a$). In equation (2), $w_{p_a}$ denotes the weight of $a^{th}$ attributes, and 'A' denotes the total number of product attributes. $\left(S(P_1,P_2)_{p_a}\right)^{i}$ represents the similarity score between the products P1 and P2 in the perspective of $a^{th}$ attribute over the set of desire or purchased products of $i^{th}$ user. After computing the similarity values between the users and products, the SC-HR approach represents the score of each user on each product with the combination of similarity scores.

### 5.2. Unsupervised Learning-based Feature Extraction

The SC-HR approach extracts the features to recognize the preferences of the users on the products through unsupervised learning. It applies the deep autoencoder and hybrid filtering model to extract the potential features influencing user preferences. To effectively extract the user desires, it handles the data sparsity constraint through sentiment analysis and the deep autoencoder-based representation.

### 5.2.1. User-Product Relation Extraction using Autoencoder

The SC-HR approach initially utilizes the textual feedback of the review on the product to handle the data sparsity for the existing users who fail to provide the ratings. The sentiment labels of the review assist in generating the rating score for only the unrated products by the previous users in the e-commerce site, which does not support the rating score computation for the sparse data influenced by the new users and new products. In essence, the SC-HR approach enriches the user-Product relation matrix with the knowledge of the reviews provided by the users on a particular product. By analyzing the sentiment of the review, it computes the rating score or preference score for the users to address the data sparsity over the user-Product interactions partially. Consequently, the SC-HR approach infers the ratings for all the products, which is more beneficial for the unrated products by the users when the users only submit the reviews instead of submitting the ratings for the products. Analyzing the sentiments or opinion from the reviews boost up the user preference extraction, that facilitates the personalized recommendation.

$$\widehat{\mathrm{R}} = \begin{cases} \left(\dfrac{\mathrm{R_{i,p}} + \left(\mathrm{R_{i,p}} \times \mathrm{S_{i,p}}\right)}{2}\right), \text{if } \mathrm{R_{i,p}} \neq \mathrm{NaN} \\ 2 \times e^{\mathrm{S_{i,p}}}, \text{if } [\mathrm{R_{i,p}} = \mathrm{NaN}]\&\&[\mathrm{i} \neq \mathrm{i_{New}}]\&\&[\mathrm{P} \neq \mathrm{P_{New}}] \end{cases} \quad (3)$$

Equation (3) computes the new rating score ($\widehat{\mathrm{R}}$) for the users using the sentiment score of the review provided by the $i^{th}$ user on product 'P' ($\mathrm{S_{i,p}}$). Thus, the SC-HR approach effectively updates the rating score in the user-Product relations using equation (3).

In addition to the review-based data sparsity in the user-Product relations, the SC-HR approach handles the data sparsity occurred by the new users and new products using the deep autoencoder. The proposed recommendation model applies the deep autoencoder to represent the sparse data for the preference extraction. The deep autoencoder is an unsupervised learning model that performs the data compression and reconstruction of the compressed or reduced data for the identical representation of the input data. It applies the backpropagation with the target of the lossless generation of the data for the input data. The main objective of the deep autoencoder is to generate the low dimensional representation of data with respect to the reconstruction error between the original user-Product relations and the reconstructed user-product relations. To resolve the data sparsity in the recommendation model, autoencoders have been widely used to reduce the data dimensions, noise removal, and feature extraction. The SC-HR approach employs the deep autoencoder to transform the original high-dimensional and inconsistent input to the low-dimensional and consistent user-Product relation representation without the missing rating information. In the e-commerce sites, the users are reluctant to provide feedback on products, and the new users and new products fail to provide and have feedback information, respectively, which causes the data sparsity while representing the user-Product relations. In the SC-HR approach, the encoder transforms the high-dimensional input data $X=\{X_1, X_2,..., X_n\}$ into a low dimensional hidden representation of the user-product relations. 'X' refers to the relational matrix of the users and products in the perspective of preference score from the knowledge of the ratings, reviews, and metadata.

Hence, the SC-HR approach generates the low-dimensional reconstructed data with the assistance of the deep autoencoder.

### 5.3. Preference Extraction and Recommendation

To precisely extract the preferences, the SC-HR approach performs the hybrid filtering that combines the low-dimensional representation of the user-Product relations with the knowledge of the metadata of the products. It applies the collaborative filtering model through the deep autoencoder-based low-dimensional representation and applies the content-based filtering to enhance the user-Product relation representation further. By mapping the users and products with the metadata of the products, the SC-HR approach augments the representation of the user-Product matrix with the inherent preferences of the users on the products. The main objective behind the hybrid filtering-based recommender system is to extract the preferences of the users particularly, for the new products using the content-based filtering even when the collaborative filtering model results in the low-dimensional representation using the deep autoencoder. The hybrid filtering model addresses the shortcomings in the domain knowledge analysis in the collaborative filtering and the similar users' preferences analysis in the content-based filtering models.

The SC-HR approach computes the desired score for each user on the product using equation (4), which is based on the content-filtering knowledge along with the click-rate on the product ($C_i^P$). Instead of utilizing the users' profile information, it exploits the click rate to compute the desired score of the product for the new user. Moreover, the SC-HR approach utilizes the rating information from similar users to handle the missing ratings, which the deep autoencoder model further reconstructs. Hence, the deep autoencoder-based representation and click-rate consideration belong to the collaborative filtering model. In equation (4), the SC-HR approach computes the preference score for the ith user on the Pth product with the scenario of new product and new user in the e-commerce site with reference to equation (2).

$$D(i,P) = \begin{cases} \sum_{P \in M} \left[ \frac{\frac{\sum_{p_a=1}^{A} \left(S(P_{New},P)_{p_a}\right)^{i} \times w_{p_a}}{\sum_{p_a=1}^{A} w_{p_a}}}{M-1} \right], \text{if } P = P_{New} \\ \widehat{C_i^P}, \text{if } i = i_{New} \end{cases} \quad (4)$$

To compute the click-rate score ($\widehat{C_i^P}$) in equation (4), the SC-HR approach applies equation (5) with the analysis of the click-rate score of the new user on all the products. The new user only clicks a few of the products within a short period. Even the e-commerce site contains numerous products that are relevant to the new user. Hence, the SC-HR approach generates the preference score on all the products based on the knowledge of products' metadata in the clicked products of the new user using equation (5).

$$\widehat{C_i^P} = \begin{cases} \alpha \times C_i^P, \text{if } P = P(C_i^P \geq 1) \\ \sum_{P,K \in M} \left[ \frac{\frac{\sum_{pa=1}^{A} \left(S(P_{C(i)}, P)_{pa}\right)^i \times w_{pa}}{\sum_{pa=1}^{A} w_{pa}}}{K} \right], \text{if } P = P(C_i^P = 0) \end{cases} \quad (5)$$

In equation (5), 'K' denotes the number of clicked products by the i$^{th}$ user in the e-commerce site. 'α' refers to the normalized parameter for the click-rate score, assigned as 0.01. Moreover, the SC-HR approach estimates the preference of each user on the products with the consistent and complete user-Product relation dataset structured from the hybrid filtering-based preference extraction. By examining the latent relationships between the users and products, the SC-HR approach significantly improves the quality of the preference prediction with the assistance of the deep learning model. From the results of the predicted preference score on the products, the SC-HR approach recommends the personalized set of products for each user in the e-commerce site.

## 6. Experimental Evaluation

To exemplify the performance of the recommendation model, the experimental framework evaluates the SC-HR approach with the Sparsity-aware Contextual Recommender (S-CR) model [20] on the recommendation test dataset.

### 6.1. Experimental Setup

The experimental framework implements the proposed recommendation model on Ubuntu 16.04 64-bit machine with a 3GHz Intel CPU and 16GB memory. It implements the deep learning-based hybrid recommendation model using the Python programming language. In particular, to conduct the experiments of the text processing and the deep learning models, the experimental framework employs the python libraries with the python version 3.6.8. The experimental model uses the publicly available Amazon Product recommendation dataset [27] that comprises the ratings, textual reviews, and metadata to test the hybrid recommendation model. The experimental model utilizes the Software products as the test dataset from the Amazon product review dataset. The Amazon Software product consists of three files related to the software product, such as the core data, ratings, and metadata. In the test dataset, the core data contains 12,805 core data in which each user and product has at least five reviews. Whereas the rating file comprises 4,59,436 instances, and product metadata consists of 26,815 instances. From the 12,805 data, the experimental model tests the 1316 unique users on 343 unique products with new users and new products. Product metadata incorporates the product description, price, brand, category, package type, size, and so on. Also, the experimental model generates the synthetic click-rate data for the users on the products to implement the proposed algorithm.

#### 6.1.1. Evaluation Metrics

To validate the quality of the recommendation methodology, the experimental framework employs the decision support methods of precision and recall metrics and error-based methods of MAE and RMSE metrics. The experimental model tests the relevant items to be recommended for each user from the perspective of the timestamp of the product by a particular user or their similar users in the recommender system.

**Precision:** It is the ratio between the number of correctly suggested products based on the timestamp and the total number of suggested products for a particular user.

**Recall:** It is the ratio between the number of correctly suggested products based on the timestamp and the total number of products to be suggested for a particular user.

To measure the error rate in the product recommendation, the experimental model utilizes the MAE and RMSE metrics.

**Mean Absolute Error:** It measures the deviation of the predicted values ($y_i$') by the recommendation model from the actual values ($y_i$). 'n' refers to the total number of ratings of the $i^{th}$ user on the $j^{th}$ set of products.

$$\text{MAE}=\frac{1}{n}\left[\sum_{i,j}\left|y_{i,j}-y'_{i,j}\right|\right]$$

**Root Mean Square Error:** It measures the error between the actual rating values and predicted rating values.

$$\text{RMSE}=\sqrt{\frac{1}{n}\sum_{i,j}\left(y_{i,j}-y'_{i,j}\right)^2}$$

### 6.2. Experimental Results

The evaluation model compares the performance of the proposed recommendation approach with the existing recommendation approach with the help of different performance metrics for the different scenarios.

#### 6.2.1. Performance of MAE

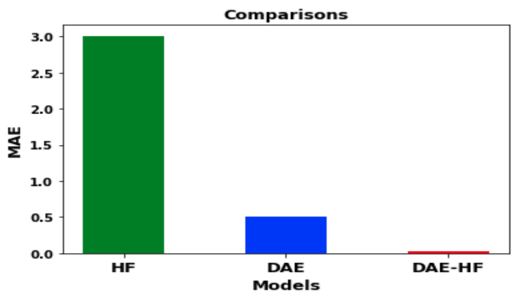


**Figure 3: Comparative Performance of MAE in the SC-HR**

Figure 3 depicts the MAE of the Hybrid Filtering (HF), Deep AutoEncoder (DAE), and the proposed SC-HR approach with DAE-HF of recommendation model for the Amazon software product dataset. As shown in Figure 3, only the hybrid filtering-based recommendation model achieves a higher MAE value as 3.03 when there is the existence of new users and new products. In essence, when the number of 1316 unique users including 17 new users and 343 unique products including 8 new products, the hybrid filtering-based proposed SC-HR model obtains the MAE value as 0.02 only whereas, the DAE model for the same test dataset yields 0.5 of MAE. As a result, the combination of the DAE and HF in the proposed SC-HR approach comparatively reduces the MAE and leverages the recommender model to suggest the relevant products to the users.

### 6.2.2. Sparsity Level

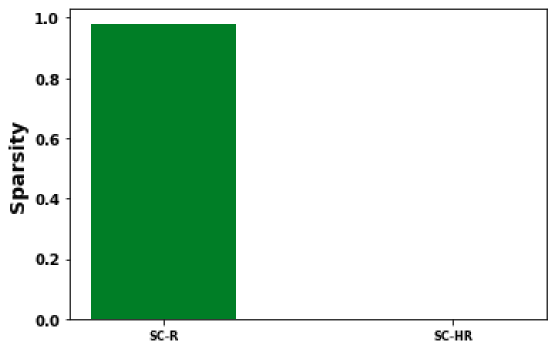


**Figure 4: Comparative Performance of Resolved Sparsity Constraint**

Figure 4 illustrates the level of sparsity constraint resolved by the proposed and existing models for Amazon's software product dataset with 1316 users and 343 products. The sparsity level measures the percentage of available ratings in the dataset for combining the users and products. The sparsity level linearly decreases with the enrichment from the raw input data to the deep autoencoder-based data representation. By contextually resolving the sparsity from the perspectives of the users' ratings and reviews, the proposed SC-HR approach alleviates the sparsity level from 98% to 0% and facilitates the accurate recommendation. For 1316 users and 343 products, even though the rating and review-based enriched data focus on handling the sparsity constraint in the product recommendation model, the sparsity level in the existing S-CR data is 98% due to the lack of handling the sparsity for the new users and new products. In addition, it fails to support the handling of the sparsity when the user has only one rating and review on the product. The proposed deep autoencoder-based data representation retains the user-product data with minimum sparsity compared to the existing dataset. The proposed SC-HR approach updates the rating information in the input dataset with the assistance of the sentiment score of the user on the product reviews. Moreover, to resolve the data sparsity in the user-product relationships, the proposed approach models the deep autoencoder for high-level data representation and reduces the sparsity level.

### 6.2.3. Precision and Recall

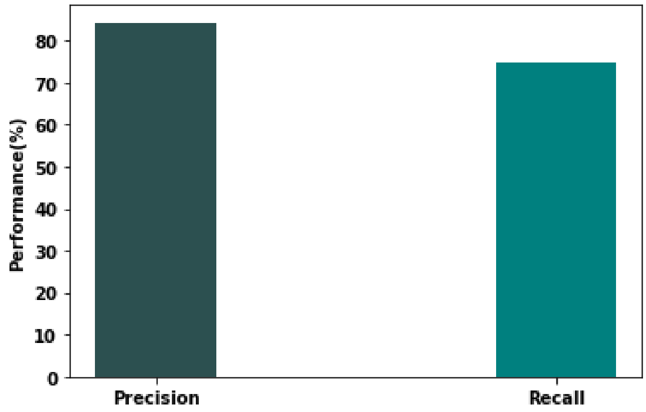


**Figure 5: Performance of Precision and Recall**

Figure 5 shows the precision and recall of the proposed SC-HR approach. The proposed SC-HR approach obtains 84.2% precision by analyzing the ratings and reviews of the users and metadata of the products to extract the preferences of the users. Moreover, the proposed SC-HR approach applies the deep autoencoder to transform into the high-level representation of the sparse input data. The recall performance of the proposed SC-HR is depicted in Figure 5. The proposed SC-HR approach yields 74.9% of recall by greatly utilizing the knowledge from the feedback of the users and metadata of the products. By testing the previously preferred products by the user, the proposed approach generates the recommendations that are preferred in the future product list and yields accurate recommendations. Instead of suggesting the rated products by a particular user, it recommends the most relevant products in the perspective of users and product relevance through hybrid filtering. Thus, the proposed SC-HR approach accurately recommends the products to the users by applying the deep learning-based hybrid filtering and handling the cold-start user and cold-start product constraints in the e-commerce dataset.

### 6.2.4. Comparison of RMSE

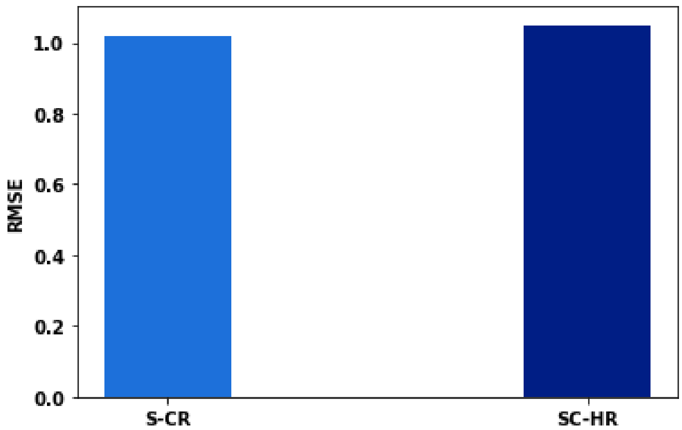


**Figure 6: Comparative Performance of RMSE**

Figure 6 depicts the RMSE of the proposed SC-HR as well as the existing S-CR approaches. As illustrated in Figure 6, the proposed SC-HR approach obtains a negligibly higher RMSE value than the existing S-CR approach. In essence, for the sparsity level is 98%, the proposed SC-HR approach yields the RMSE value as 1.05, whereas the existing S-CR approach obtains 1.02 of RMSE due to the utilization of metadata for the relevant products. Even though the S-CR approach outperforms the proposed SC-HR approach in terms of RMSE value, the SC-HR approach suggests the relevant products to the new users even when there are new products. In contrast, the existing S-CR approach only utilizes the ratings and reviews of the users, which fails to recognize the user interests in the products for the new users and new products. In addition, the validation of the existing model with the collaborative filtering handles the sparsity constraint and predicts the ratings for the recommendation whereas, the proposed SC-HR approach validates the predicted values for the hybrid filtering results. Due to the lack of actual rating values for the new users and new products, the proposed SC-HR approach obtains a comparatively high RMSE value than the existing S-CR approach.

### 6.2.5. Comparison of MAE

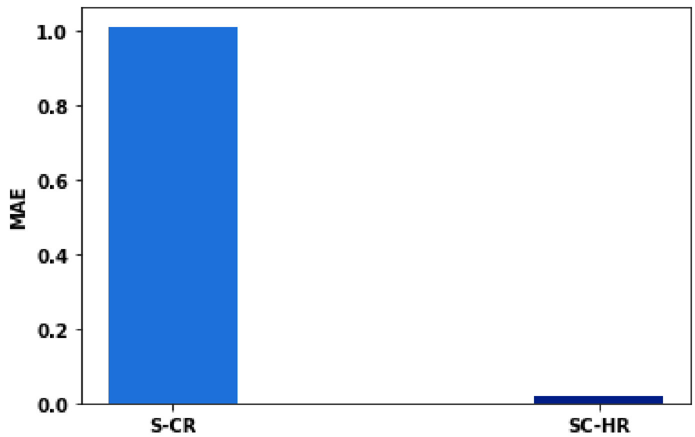


**Figure 7: Comparative Performance of MAE**

The MAE performance of the proposed SC-HR and existing S-CR approaches are illustrated in Figure 7 while testing the recommendation model for the Amazon product dataset with 98% of data sparsity. The recommendation model without handling the sparsity and cold-start constraints, misguide the suggestion of irrelevant products to the users or inaccurate rating prediction due to the lack of adequate knowledge for the user interests' on the products. The MAE value of the proposed SC-HR approach is minimum as 0.02 even when there is a higher sparsity level. In contrast, the existing S-CR approach increases the MAE value to 1.01 due to the lack of addressing the cold-start constraints. In essence, the existing S-CR approach fails to handle the sparsity when there is a lack of rating and review information for the users on the products. Even though the existing S-CR approach performs the sentiment analysis on the reviews to resolve the data sparsity constraint, it fails to transform the complete user-rating-product data, which leads to inaccurate recommendations. Moreover, the SC-HR approach enhances the representation of the sparse user-product dataset with the help of the deep autoencoder model, which leverages the accurate preference extraction.

## 7. Conclusion

This paper presented the hybrid recommendation model using a deep learning algorithm to address the data sparsity and cold-start problems in the recommender systems. In addition to the hybrid filtering, the SC-HR approach has considered the implicit feedback and the explicit feedback of the users on the e-commerce sites. By applying the deep autoencoder model, it has transformed the high-dimensional representation of the user-product relationship into the low-dimensional representation. The content-based filtering has resolved the difficulty in extracting the user preferences on the new products. Moreover, it has utilized the implicit feedback of the click-rate information in the e-commerce site to suggest personalized products to the new user. Thus, the SC-HR approach has predicted the user preferences using the deep learning algorithm and ensured the personalized recommendation. The experimental results have illustrated that the SC-HR approach has yielded 84.2% of precision and obtained 0.99 minimal MAE value than the existing S-CR approach.

## References

[1]. Mohamed, Marwa Hussien, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim, "Recommender systems challenges and solutions survey", In 2019 IEEE International Conference on Innovative Trends in Computer Engineering (ITCE), pp.149-155, 2019.

[2]. Fayyaz, Zeshan, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities", Applied sciences, Vol. 10, No. 21, p. 7748, 2020.

[3]. Juan, Wang, Lan Yue-xin, and Wu Chun-ying, "Survey of recommendation based on collaborative filtering", In Journal of Physics: Conference Series, Vol. 1314, No. 1, p. 012078, IOP Publishing, 2019.

[4]. Haruna, Khalid, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction", Applied Sciences, Vol. 7, No. 12, p. 1211, 2017.

[5]. Kulkarni, Saurabh, and Sunil F. Rodd, "Context Aware Recommendation Systems: A review of the state of the art techniques", Computer Science Review, Vol. 37, p. 100255, 2020.

[6]. Pawan Kumar Tiwari, Mukesh Kumar Yadav, R. K. G. A. . (2022). Design Simulation and Review of Solar PV Power Forecasting Using Computing Techniques. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(5), 18–27. https://doi.org/10.17762/ijrmee.v9i5.370

[7]. Idrissi, Nouhaila, and Ahmed Zellou, "A systematic literature review of sparsity issues in recommender systems", Social Network Analysis and Mining, Vol. 10, No. 1, pp. 1-23, 2020.

[8]. Singh, Monika, "Scalability and sparsity issues in recommender datasets: a survey", Knowledge and Information Systems, Vol. 62, No. 1, pp. 1-43, 2020.

[9]. Batmaz, Zeynep, Ali Yurekli, Alper Bilge, and Cihan Kaleli, "A review on deep learning for recommender systems: challenges and remedies", Artificial Intelligence Review, Vol. 52, No. 1, pp. 1-37, 2019.

[10]. Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay, "Deep learning based recommender system: A survey and new perspectives", ACM Computing Surveys (CSUR), Vol. 52, No. 1, pp. 1-38, 2019.
[11]. Zhang, Guijuan, Yang Liu, and Xiaoning Jin, "A survey of autoencoder-based recommender systems", Frontiers of Computer Science, Vol. 14, No. 2, pp. 430-450, 2020.
[12]. Kadhim, R. R., and M. Y. Kamil. "Evaluation of Machine Learning Models for Breast Cancer Diagnosis Via Histogram of Oriented Gradients Method and Histopathology Images". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 36-42, doi:10.17762/ijritcc.v10i4.5532.
[13]. Guo Guibing, Huihuai Qiu, Zhenhua Tan, Yuan Liu, Jing Ma, and Xingwei Wang, "Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems", Knowledge-Based Systems, Vol. 138, pp. 202-207, 2017.
[14]. Yin, Hongzhi, Weiqing Wang, Liang Chen, Xingzhong Du, Quoc Viet Hung Nguyen, and Zi Huang, "Mobi-SAGE-RS: A sparse additive generative model-based mobile application recommender system", Knowledge-Based Systems, Vol.157, pp.68-80, 2018.
[15]. Ahmadian, Sajad, Mohsen Afsharchi, and Majid Meghdadi, "A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems", Multimedia tools and applications, Vol. 78, No. 13, pp. 17763-17798, 2019.
[16]. Bag, Sujoy, Susanta Kumar, Anjali Awasthi, and Manoj Kumar Tiwari, "A noise correction-based approach to support a recommender system in a highly sparse rating environment", Decision Support Systems, Vol. 118, pp.46-57, 2019.
[17]. Roko, Abubakar, Abba Almu, Aminu Mohammed, and Ibrahim Saidu, "An Enhanced Data Sparsity Reduction Method for Effective Collaborative Filtering Recommendations", International Journal of Education and Management Engineering, Vol. 10, No. 1, Vol. 27, 2020.
[18]. Gupta, D. J. . (2022). A Study on Various Cloud Computing Technologies, Implementation Process, Categories and Application Use in Organisation. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(1), 09–12. https://doi.org/10.17762/ijfrcsce.v8i1.2064
[19]. Zhang, Zhipeng, Yao Zhang, and Yonggong Ren, "Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering", Information Retrieval Journal, Vol. 23, No. 4, pp. 449-472, 2020.
[20]. Feng, Chenjiao, Jiye Liang, Peng Song, and Zhiqiang Wang, "A fusion collaborative filtering method for sparse data in recommender systems", Information Sciences, Vol. 521, pp. 365-379, 2020.
[21]. Milenkoski, Martin, Diego Antognini, and Claudiu Musat, "Recommending Burgers based on Pizza Preferences: Addressing Data Sparsity with a Product of Experts", arXiv preprint arXiv:2104.12822, 2021.
[22]. Tahmasebi, Faryad, Majid Meghdadi, Sajad Ahmadian, and Khashayar Valiallahi, "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem", Multimedia Tools and Applications, Vol. 80, No. 2, pp. 2339-2354, 2021.

[23]. Osman, N.A., S.A.M. Noah, and M. Darwich, “Contextual sentiment based recommender system to provide recommendation in the electronic products domain”, International Journal of Machine Learning and Computing, Vol. 9, No. 4, pp. 425-431, 2019.
[24]. Liu, Yu, Shuai Wang, M. Shahrukh Khan, and Jieyu He, “A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering”, Big Data Mining and Analytics, Vol. 1, No. 3, pp. 211-221, 2018.
[25]. V. N. Patil and D. R. Ingle, “A Novel Approach for ABO Blood Group Prediction using Fingerprint through Optimized Convolutional Neural Network”, Int J Intell Syst Appl Eng, vol. 10, no. 1, pp. 60–68, Mar. 2022.
[26]. Deng, Zhi-Hong, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and S. Yu Philip, “Deepcf: A unified framework of representation learning and matching function learning in recommender system”, In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, pp. 61-68, 2019.
[27]. Lv, Fuyu, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng, “SDM: Sequential deep matching model for online large-scale recommender system”, In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2635-2643, 2019.
[28]. Pan, Yiteng, Fazhi He, and Haiping Yu, “A correlative denoising autoencoder to model social influence for top-N recommender system”, Frontiers of Computer science, Vol. 14, No. 3, pp. 1-13, 2020.
[29]. Katarya, Rahul, and Yamini Arora, “Capsmf: a novel product recommender system using deep learning based text analysis model”, Multimedia Tools and Applications, Vol. 79, No. 47, pp. 35927-35948, 2020.
[30]. Yu, Miao, Tianqi Quan, Qinglong Peng, Xu Yu, and Lei Liu, “A model-based collaborate filtering algorithm based on stacked Auto Encoder”, Neural Computing and Applications, pp. 1-9, 2021.
[31]. Singh, S. . (2022). Unconditionally G ?odel Degeneracy for Quasi-Meager, Smooth Moduli. International Journal on Recent Trends in Life Science and Mathematics, 9(1), 28–36. https://doi.org/10.17762/ijlsm.v9i1.139
[32]. Amazon Product Dataset, Available Online at: https://nijianmo.github.io/amazon/index.html, Accessed on August, 2021.