Leveraging Deep Semantics for Sparse Recommender Systems (LDS-SRS)

Adel Alkhalil
Department of Software Engineering,
College of Computer Science and Engineering,
University of Ha'il, Ha'il, 81481, Saudi Arabia

***Abstract*—RS (Recommender Systems) provide personalized suggestions to the user(s) by filtering through vast amounts of similar data, including media content, e-commerce platforms, and social networks. Traditional recommendation system (RS) methods encounter significant challenges. Collaborative Filtering (CF) is hindered by the lack of sufficient user-product engagement data, while CBF (Content Based Filtering) depends extensively on feature extraction techniques in order to describe the items, which requires an understanding of both content contextual and semantic relevance of the information. To address the sparsity issue, various matrix factorization methods have been developed, often incorporating pre-processed auxiliary information. However, existing feature extraction techniques generally fail to capture both the semantic richness and topic-level insights of textual data. This paper introduces a novel hybrid recommendation system called Topic-Driven Semantic Hybridization for Sparse Recommender Systems (LDS-SRS). The model leverages the semantic features from item descriptions and incorporates topic-specific data to effectively tackle the challenges posed by data sparsity. By extracting embeddings that capture the deep semantics of textual content—such as reviews, summaries, comments, and narratives—and embedding them into Probabilistic Matrix Factorization (PMF), the framework significantly alleviates data sparsity. The LDS-SRS framework is also computationally efficient, offering low deployment time and complexity. Experimental evaluations conducted on publicly available datasets, such as AIV (Amazon Instant Video) and Movielens (1 Million & 10 Million), demonstrate the exceptional ability of the method to handle sparse user-to-item ratings, outperforming existing leading methods. The proposed system effectively addresses data sparsity by integrating embeddings that encapsulate the deep textual semantics content, including summaries, comment(s), and narratives, within PMF (Probabilistic Matrix Factorization). The LDS-SRS framework is also highly efficient, characterized by minimal deployment time and low computational complexity. Experimental evaluations conducted on publicly available MovieLens (1 Million and 10 Million) and AIV (Amazon Instant Video) benchmark datasets demonstrate the framework's exceptional ability to handle sparse user-item ratings, surpassing existing advanced methods.**

***Keywords*—LDA-2-Vec technique; content representation; topic-based modeling; probabilistic matrix decomposition**

I. Introduction

RSs are integral to e-commerce, offering tailored product suggestions for various items, including movies, books, clothing, and news. In recent years, their applications have expanded to areas such as social media platforms, websites, and articles. Leading Fortune 500 companies like Facebook, Netflix, and eBay have created proprietary RSs to predict and cater to customer preferences. The success of these organizations in the business market is significantly influenced by the effectiveness of their RSs. For instance, Netflix suggests movies based on individual user preferences, while Amazon and eBay recommend related products to customers, and social media platforms display relevant pages and advertisements to users. As a result, it is evident that recommendation systems play a crucial part in driving the financial growth of these companies [1], [2].

The principles underlying RSs are broadly categorized into CF, CB, and Hybrid Recommendation Methods [3], [4] combine multiple filtering techniques, such as Collaborative Filtering (CF), to derive user-item ratings based on historical metadata, analysing individual preferences and behaviours. It recommends new products by identifying similarities between users based on their past preferences, without relying on the specific details of the items' content [5], [6]. Conversely, Content-Based (CB) filtering analyzes item descriptions, utilizing their attributes and features to match them with user profiles. By analyzing items liked by the user, CB generates a similarity matrix to recommend the most relevant new items. This method heavily relies on item descriptions and user profiles for making accurate recommendations. This approach depends largely on item descriptions and user profiles to provide precise recommendations [7], [8]. Hybrid filtering, which integrates both CB and CF methods, utilizes user preferences along with content semantics to improve the effectiveness of recommender systems.

Collaborative Filtering (CF) is widely considered a powerful model for developing recommendation systems, as it predicts ratings by examining user-item rating matrices [1]. CF can be categorized into two main approaches: Memory Based and Model Based techniques [9], [10]. Memory-based methods concentrate on identifying similarities among users or products to find potential neighbors, using these similarities to predict ratings. However, this approach is often hindered by issues like limited data availability and the cold start problem [3], [11]. In contrast, Model-Based Collaborative Filtering utilizes trained techniques, including decision trees, clustering methods [12], [13], Bayesian models [15], latent factor models [14], and dimensionality reduction methods [16]. While these approaches can be highly effective, their implementation and upkeep demand substantial effort and computational resources, particularly due to the necessity of frequent parameter optimization [17]. Among these, Matrix Factorization (MF) stands out for its balance of scalability and precision [11]. When working with sparse or large-scale datasets, Matrix Factorisation (MF) may perform poorly, perhaps producing less accurate

predictions. MF breaks down the user product engagement grid to latent features that uncover underlying rating patterns [4].

The rising volume of online users interaction and available products has exacerbated sparse ratings problem. For Collaborative Filtering (CF)-based recommendation systems to function effectively, comprehensive user rating histories are crucial. However, upon the registration of new users, the absence of historical data leads to less accurate recommendations. Consequently, the sparsity of ratings and the lack of detailed semantics in item descriptions hinder the accuracy of predictions and recommendations [1], [18]. Trust-based collaborative filtering methods have been explored to mitigate these issues by leveraging user trust relationships to enhance recommendations [22]. To address these challenges, several methods have been introduced that integrate both user feedback data and supplementary details, including user preferences (e.g., likes, interests, and social media activity) and item specifics (e.g. reviews, summaries, and plots) [19] - [20].

To address the limitations of traditional CF and hybrid methods in handling sparse recommendation scenarios, we choose to propose the LDS-SRS framework. This method enhances PMF by embedding deep semantic and topic-driven representations, ensuring more accurate recommendations even in sparse datasets. Unlike neural network-based models, which demand high computational resources, LDS-SRS efficiently integrates on textual knowledge while maintaining low computational complexity.

This research seeks to address the sparsity problem by extracting the rich semantics from textual item descriptions. The proposed method utilizes an embedding model to analyze auxiliary item data, capturing semantic information enhanced with topic-related details. These embeddings are integrated into the document latent factors of Probabilistic Matrix Factorization (PMF). By employing PMF as the CF technique, experimental outcomes demonstrate the model's effectiveness. The main contributions of the research study include:

- The introduced framework, LDS-SRS, integrates the detailed semantics of textuaThe main contributions of the research study include:l item data with topic-specific insights.
- The derived embeddings serve as input for memory-based collaborative recommendation methods to enhance the precision of predicted ratings.
- A comparative analysis demonstrates that LDS-SRS outperforms existing paradigms on MovieLens (1 Million and 10 Million) and AIV (Amazon Instant Video) benchmark datasets

The structure of this paper is outlined as follows: Section II reviews the relevant literature, while Section III details the proposed model for the recommendation system. Section IV provides a comprehensive analysis of the results and performance metrics, and Section V wraps up the study with concluding remarks.

II. Literature Review

The functionality of recommendation systems (RS) is significantly affected by the cold start issue and the scarcity of user-item rating data. To enhance the accuracy of rating predictions and the quality of recommendations, research has focused on CB(Content Based) and CF(Collaborative Filtering) techniques. One major challenge, the cold start problem, is addressed by applying text analysis techniques to extract valuable information from both user and item data, allowing the system to effectively incorporate new users or items. Additionally, the CB approach [21] helps mitigate sparsity by utilizing item-specific features, making it easier to manage new items.

These techniques use low-rank factorization to approximate the user item interaction grid, improving the prediction of products. Matrix Factorization techniques break down the user item grid into lower-dimensional embeddings for both users & products. Additionally, some systems enhance collaborative recommenders by incorporating trust-based information [46]. The Topic-MF model [23] integrates biased matrix factorization techniques to address sparsity by combining rating data with topic-related information derived from user reviews. Since text reviews provide more semantic depth than ratings, they offer richer insights into user and item characteristics. Functional Matrix Factorization (FMF) uses interview-based tools to build user profiles [24], while the Latent Dirichlet Allocation (LDA) model is employed to extract topic-specific details from item descriptions.

ALS (Alternating Least Squares) [25] and SGD (Stochastic Gradient Descent) [4] are well-established optimization techniques frequently used for training Matrix Factorization models. In [26], two alternative multiplicative update methods for Non-negative Matrix Factorization (NMF) were proposed, each with unique update rules for the multiplicative factors. The method presented in [27] integrates Weighted Singular Value Decomposition (WSVD) with a linear regression model, where each latent factor is assigned a weight parameter. Furthermore, [28] introduced a cosine similarity-based Matrix Factorization (CosMF) model, aimed at addressing sparsity issues without the need for additional data processing. However, sparse rating data often hampers the effective training of user and product vectors because of the lack of supplementary data. This model substitutes dot products with cosine similarity for users and products with limited data, helping to mitigate the adverse effects of missing auxiliary data.

Probabilistic Matrix Factorization (PMF) [29] outperforms SVD in recommendation tasks. Recently, advanced generalizations of PMF, such as Generalized PMF [30], ConvMF [32], and Bayesian PMF [31], have been introduced. While the CB model [21] addresses sparsity by utilizing item features, it struggles with effectively accommodating new users. Hybrid RS models were introduced to integrate CF with user or product content information [20], [32], [33]. For example, in [34], item features were generated using a Stacked Denoising AutoEncoder (SDAE) trained on online item descriptions. These features were integrated into the timeSVD++ CF model, which uses a weighted bag-of-words approach but fails to capture semantic word similarities. Approaches for document representation, including Latent Dirichlet Allocation (LDA) and Stacked Denoising Autoencoders (SDAE), have been utilized to extract content features from supplementary sources like reviews, synopses, or abstracts [35], [36], [20], [37]. Wang et al. combined Probabilistic Matrix Factorization (PMF) and SDAE to enhance the accuracy of latent models for rating

predictions [20]. In contrast, Collaborative Deep Learning (CDL) adopted a more straightforward collaborative filtering approach, concentrating on generating top-N recommendations.

Convolutional Neural Networks (CNNs) are widely applied in Digital Image Processing (DIP) and Computer Vision (CV) tasks [38]. Additionally, CNNs have shown considerable promise in Natural Language Processing [39], [40], [41] and information retrieval applications. CNNs effectively capture contextual features from images or textual descriptions by employing subsampling, shared weights, and receptive fields [38]. CNNs can incorporate word embeddings, such as Word2Vec, to extract contextual word features. ConvMF [32] integrates CNN architecture, specifically designed for NLP, with PMF. Similarly, [42] utilized a pre-trained embedding model with CNNs, which were then integrated into PMF. While these models capture the contextual characteristics of item textual data, they often fail to incorporate deep semantics. Furthermore, these models may struggle with negative values in latent representations, potentially leading to the degradation of information during rating prediction [49]. In [51], a hybrid recommendation system was introduced that utilizes content embeddings to predict scores for cold start products. The $HRS-CE$ model generates word embeddings from item descriptions and uses them to build user profiles for recommending similar items. [52] enhanced Non-Negative Matrix Factorization (NMF) by incorporating contextual item information through a sophisticated embedding model that captures semantic meaning as well as additional contextual data [53], [54].

In contrast to traditional recommendation system (RS) models, the proposed LDS-SRS model adopts a hybrid approach that captures comprehensive content embeddings of products and integrates them with CF (collaborative filtering) methods for predicting missing ratings. While extracting content features, both the semantic and topic-related elements of item descriptions are recognized and incorporated into the hidden representations of the PMF(Probabilistic Matrix Factorization) model. The LDS-SRS model does not rely on neural networks for natural language processing tasks, nor does it require the iterative updating of item embeddings. As a result, it offers greater computational, temporal, and memory efficiency compared to other RS algorithms.

III. Proposed Methodology

A recommender system is presented that combines deep semantic insights with topic-specific information to predict item ratings. The model combines "local" features extracted from dense vectors with "global" representations based on topic information, aligning them within a unified space. These contextual embeddings are subsequently utilized to initialize the item representation factors within the framework of PMF.

This section outlines the both the mathematical artifacts/formulation and system's design. Fig. 1 shows the suggested paradigm architecture. Experiments are conducted using the MovieLens dataset, where each film is treated as a product, with its corresponding plot acting as the product description.

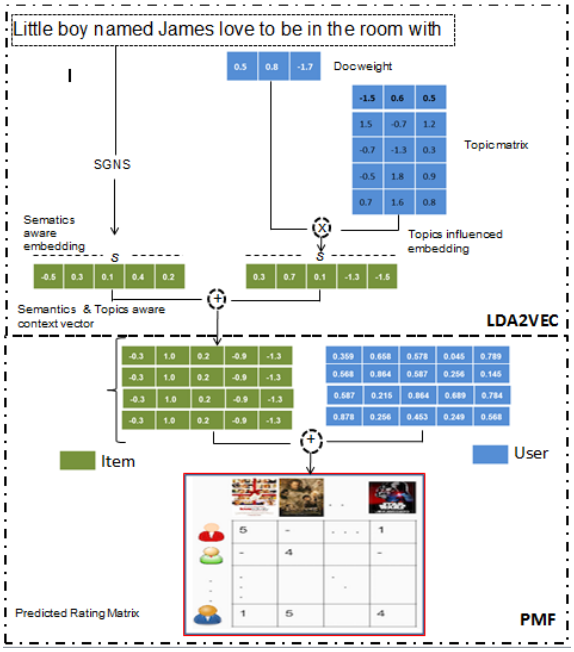

Fig. 1. Proposed framework architecture.

A. Mathematical Modeling

Let $U = [u_1, u_2, u_3, \ldots, u_m]$ denote the set of users, and $I = [i_1, i_2, i_3, \ldots, i_n]$ denote the set of products. The user product rating grid is represented as a two-dimensional array, as follows:

$$R_{ui} \in [0.5, 1, 2, 3, 4, 5]^{m \times n}$$

Here m: Total Users No.
n: Items No.
R_{ui}: Ratings of the Item(i) by the User's(u).

The proposed LDS-SRS approach aims to estimate ratings $\hat{R}_{ui}$ for items (i) that are unrated in the rating matrix, based on the existing ratings. For collaborative filtering, probabilistic matrix factorization (PMF) is applied. PMF decomposes the matrix $R^{m \times n}$ into three separate components: $U \in R^{s \times m}$, $I \in R^{s \times n}$, and $\Omega \in R^{s \times s}$. Thus,

$$RU \times \Omega \times I \tag{1}$$

In this scenario, $R^{m \times n}$ refers to the $(m \times n)$ matrix of user-item ratings, where U represents the latent factors associated with users and I represents those associated with items. Additionally, Ω is a diagonal matrix that represents the hidden patterns within R, while s represents the count of latent factors contained in R.

In order to address the problem of data sparsity, LDS-SRS integrates embeddings containing semantic and topic-related information from text into the probabilistic matrix factorization (PMF) model. The aim is to learn latent representations for users and products, $U \in R^{s \times m}$ & $I \in R^{s \times n}$, in a way that their multiplication $(U^T I)$ closely matches the user-item rating grid R [29]. These latent models are refined by minimizing a

cost function ℓ, which includes the squared difference between actual and predicted ratings, along with regularization terms.

$$\ell = \sum_{u}^{m}\sum_{i}^{n} A_{ui}(r_{ui} - u_u^T i_i)^2 + \alpha \sum_{u}^{m} \|u_u\|^2 + \beta \sum_{i}^{n} \|i_i\|^2 \quad (2)$$

In this context, A_{ui} is defined as a function that returns 1 if user *u* has rated item *i* and 0 otherwise. The performance of the model is evaluated by computing the Root Mean Square Error (RMSE) on the test dataset. A Gaussian noise-based probabilistic linear framework is utilized. The probability distribution based on the observed ratings is expressed as follows:

$$\rho(R|U, I, \sigma^2) = \prod_{u}^{m}\prod_{i}^{n} \eta(r_{ui}|u_u^T i_i, \sigma^2)^{A_{ij}} \quad (3)$$

In this context, $\eta(x|\mu, \sigma^2)$ denotes the probability density function of a Gaussian distribution characterized by a mean value of μ and a variance of σ^2. A spherical Gaussian prior, centered at zero, is applied to the latent vector of the user, as per the equation below:

$$\rho(U|0, \sigma_U^2) = \prod_{u}^{m} \eta(u_u|0, \sigma_U^2 A)$$

In contrast to the user latent method, our approach suggests deriving the probabilistic representation of the item's latent vector from a features paradigm based on the D_i (product description), which includes a Gaussian noise component represented by a gamma variable.

$$i_i = \textit{Embeddings}(D_i) + \gamma_i$$

$$\gamma_i \sim \eta(0, \sigma_I^2 A)$$

The item latent model's distribution is then expressed as:

$$\rho(I|D_i, \sigma_I^2) = \prod_{i}^{n} \eta(i_i|\textit{Embeddings}(D_i), \sigma_I^2 A)$$

Here, D_i represents the description document linked to item *(i)*. The context vector derived from the dense embedding characterizes the Gaussian distribution for the item, with the variance being determined by the Gaussian noise. This variance is essential for initializing the item latent representation in the PMF framework.

In the proposed paradigm, an embedding is created specifically for movie plots. This embedding effectively captures the local semantic features of each plot, while also incorporating broader, topic-related information. Let G represent the movie plot (or item description), consisting of l words, denoted as $w_1, w_2, w_3, \ldots, w_l$.

$$G \xleftarrow[\text{Plot Description}]{} \{w_1, w_2, w_3, ...w_l\}$$

A collection C is constructed from the textual descriptions of movie plots, encompassing all the movies (denoted as G), and can be expressed as:

$$C \xleftarrow[\text{Corpus Generation}]{} \{G_1, G_2, G_3, ...G_n\}$$

In which $C \in \{w_1, w_2, w_3, ...w_l, w_{l+1}, ...w_t\}$, where t denotes the total word count in the entire corpus C.

The Word2Vec (w2v) model employs the Skipgram Negative Sampling (SGNS) method on the corpus C to produce dense word embeddings with s dimensions. This paradigm captures local semantic relationships within a plot by computing the probabilistic weights of words in relation to a reference word. It *locally* predicts the surrounding words based on the given pivot word in the context of a movie plot.

$$P(P_{TW}|P_{PW})$$

In this context, P_{TW} represents the likelihood of the intended word, while P_{PW} denotes the likelihood of the pivot word within the model. The Word2Vec model generates enriched real-valued embeddings that capture both semantic and syntactic relationships within the plots. These compact representations offer increased flexibility but are less interpretable. In contrast, Latent Dirichlet Allocation (LDA) is applied to the dataset to extract topic-specific information from the text. LDA generates a sparse probability vector, providing better interpretability and a clearer understanding of the data. Operating on a *global* scale, LDA constructs a unified plot vector that summarizes topic-related features, which are later used for word prediction within the plot.

$$P(P_{TW}|P_{Topics})$$

Here, P_{Topics} represents the probability of the topic information. The enhanced embedding model combines the dense representations produced by Word2Vec (w2v) with the interpretability provided by Latent Dirichlet Allocation (LDA). This method combines the w2v embeddings of a plot with sparse vectors produced by LDA to approximate a multinomial distribution across underlying word categories. The resulting representation is subsequently passed through a probabilistic model to assign distinct topics to a specific set of movies or items.

$$P(P_{TW}|P_{PW} + P_{Topics}) \quad (4)$$

Eq. 5 is composed of two components. The first component, $\pounds_{SGNS}$, adheres to the standard word2vec (w2v) methodology, aiming to maximize the likelihood of target words and negative samples in relation to the surrounding word representation. The computation is performed using the context vector C_v, the target word vector P_{TW}, the pivot word vector P_{PW}, and the vector for negative (irrelevant) words NS_v. The second component, $\pounds_{LDA}$, introduces a Dirichlet likelihood term that is linked to the document weights. The complete structure of the suggested recommendation model is depicted in Fig. 2. The overall objective function is the combination of the SGNS loss component, augmented by the Dirichlet distribution term that is applied to the document weights.

$$\pounds = \pounds_{SGNS} + \pounds_{LDA} \quad (5)$$

$$\pounds_{SGNS} = \log(C_v|P_{TW}) + \log(-C_v|NS_v) \quad (6)$$

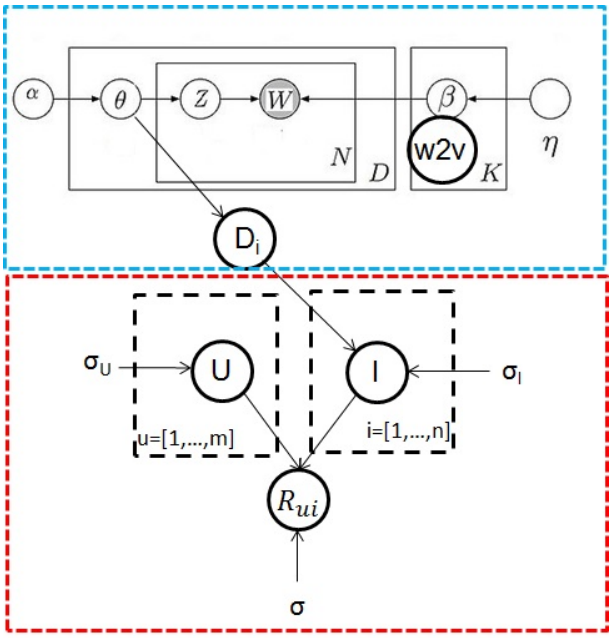

Fig. 2. Proposed paradigm with LDA2VEC embedded in W2V for generating ratings.

Ultimately, by following the outlined procedure, the embedding model processes movie plots and generates a corresponding latent vector for each one. This vector encapsulates the deep semantic context of the plot, while also capturing topic-specific information unique to each movie.

$$LV_i = \textit{Embeddings}(D_i)$$

Here, D_i refers to the description of the *i*th film, while LV_i denotes the latent vector associated with the *i*th film.

B. Optimization

In order to optimize the latent representations for users and products, a posteriori estimation is carried out as follows:

$$\max_{U,I} \rho(U, I|R, \boldsymbol{D}, \sigma^2, \sigma_U^2, \sigma_I^2) = \max_{U,I}[\rho(R|U, I, \sigma^2) \rho(U|\sigma_U^2)\rho(I|D, \sigma_I^2)] \tag{7}$$

By applying the negative logarithm to Equation 7, we obtain:

$$\ln(U, I) = \sum_{u=1}^{m}\sum_{i=1}^{n} \frac{A_{ui}}{2}(r_{ui} - u_u^T i_i)^2 + \frac{\alpha}{2}\sum_{u=1}^{m}\|u_u\|^2 + \frac{\beta}{2}\sum_{i=1}^{n}\|i_i - \textit{Embeddings}(D_i)\|^2 \tag{8}$$

Gradient descent is employed to update the latent representations for users (U) & items (I), iteratively optimizing each model while holding the other variables constant. The main objective is to identify the local minima of the OF (Objective Function) by differentiating Eq. 8 with respect to u_u and i_i in a closed-form expression. Consequently, the latent representations for both the user and the item are computed as follows:

$$u_u = \frac{IR_u}{II_u I^T + \alpha I_s} \tag{9}$$

$$i_i = \frac{UR_i + \beta \textit{Embeddings}(D_i)}{UI_i U^T + \beta I_s} \tag{10}$$

In this context, I_u refers to a diagonal matrix with diagonal elements $I_{ui}, i = 1, \ldots, n$, while R_u represents a vector containing $(r_{ui})_{i=1}^n$ for the user u. Similarly, for the item *(i)*, I_i and R_i are specified in the same way as I_u and R_u.

IV. Outcomes and Performance Assessment

This section provides an evaluation and discussion of the proposed model's performance. It starts with an in-depth description of the dataset and its processing, followed by a thorough explanation of the experimental setup. The results are then presented, analyzed, and visually displayed, with comparisons drawn to leading methods in the field.

A. Dataset Pre-processing

The Movie-Lens 1 Million corpus [43] contains 1,000,209 evaluations $R_{ui} \in \{0.5, 1, 2, 3, 4, 5\}$ for approximately 3,900 movies, rated by 6,040 users. The MovieLens 10 Million dataset includes 10,000,054 evaluations across 10,681 movies (items), contributed by 71,567 users. Additionally, the Amazon Instant Video (AIV) dataset features evaluations for 15,149 items provided by 29,757 users, with scores $R_{ui} \in \{1, 2, 3, 4, 5\}$ ranging from 1 to 5.

Table I provides a summary of the statistics for the datasets used.

TABLE I. Dataset Characteristics

Dataset Name	Total Ratings	User Count	Item Count	Rating Scale	Density (%)
AIV	1,351,88	29,757	15,149	[1 - 5]	0.030
1M	1,000,209	6,040	3,900	[0.5 - 5]	1.431
10M	1,000,005,4	71,567	10,681	[1 - 5]	4.641

To predict ratings for items, additional information such as movie plots is required. However, not all movies in the MovieLens open-source datasets have corresponding plots. To address this, the original datasets were expanded to ensure that the experiments could be carried out effectively. Movie plots were obtained using the OMDB API[1]. A Python script was written to query the OMDB database using each movie's ID, from which the corresponding plot was automatically retrieved. The gathered data underwent several preprocessing steps, including text cleaning and removal of stopwords. Stopwords, which are frequent yet low-information words (e.g. "a", "an", "the", "that"), were eliminated to enhance the quality of text analysis. The following preprocessing steps were performed on the movie plot data: 1) Plot lengths were restricted to a maximum of 200 words. 2) The Word2Vec (w2v) model was

[1]http://www.omdbapi.com

utilized with a sliding window of size 1 to capture contextual relationships among words in the dataset. 3) Subscripts were removed from the text as a precautionary measure. 4) The tokens were generated for the text corpus. 5) LDA2VEC methods were implemented and applied to the tokens of the corpus text using a pre-trained w2v model based on the same dataset [44]. 6) The vector dimensionality was configured to 100. Furthermore, movies without plot summaries were excluded from the dataset to maintain accuracy and reliability in the results.

Harnessing the detailed semantic insights encapsulated in the latent vectors produced by our embedding model, we utilized this information to initialize the item latent factors within the PMF. This step plays a crucial role in integrating the LDA2VEC model with the PMF, enabling an effective fusion of item descriptions and ratings data.

B. Evaluation Metrics

The LDS-SRS model's performance is assessed by employing a ten fold cross validation approach. The overall prediction metric is calculated by averaging the results across all 10 iterations. The training dataset contains a User-Product grid that includes known feedback, while the testing dataset consists of user-product pairs for which the ratings must be predicted. These predictions are generated by multiplying the matrices *U* and *I*. The RMSE, a commonly employed cost function in traditional rating prediction models, is computed as follows:

$$RMSE = \sqrt{\frac{\sum_{u,i}^{U,I}(R_{ui} - \hat{R})^2}{Numberof Ratings}}$$

Along with the previously discussed metrics, Precision and Recall were also employed to assess the effectiveness of the proposed paradigm. Precision represents the fraction of accurate recommendations among the overall recommendations generated, while Recall@K signifies the fraction of correct recommendations among all relevant items. To evaluate Precision and Recall, items were categorized into a pair of groups based on their assigned ratings: non-relevant (ratings 1-3) and pertinent (ratings 4-5). The items in the datasets were then categorized into those that were predicted by the model and those that were not. Precision and Recall for this approach are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

Here,

TP: True +ive (An item is accurately identified as relevant)

FP: False +ive (An item is incorrectly identified as relevant when it is not)

FN: False -ive (An item selected as negative but is actually positive)

The RMSE of the suggested model demonstrates a decrease in inaccuracies on the test set. RMSE is a widely used metric in recommendation systems as it quantifies the deviation of predicted ratings from actual user ratings, making it a reliable measure of prediction accuracy. Additionally, Precision and Recall are employed to assess the model's ability to recommend relevant items, with Precision reflecting the proportion of correctly recommended items and Recall indicating the proportion of relevant items successfully retrieved. These validation measures are essential in evaluating the effectiveness of LDS-SRS in mitigating sparsity while maintaining high-quality recommendations.

C. Outcomes & Evaluation

The effectiveness of the LDS-SRS model was assessed using datasets from *MovieLens 1M*, *10M*, and AIV. Multiple experiments were carried out to examine the model's convergence, with RMSE being the main metric for validation in each case. The model's performance was evaluated under various conditions, where *s* represents the size of the user-product latent factors. Table II shows the RMSE values for the suggested paradigm with $K = 50/100$ for the both the Movielens and AIV datasets. Fig. 3 demonstrates the model's convergence throughout the iterations for $K = 50/100$. The smallest value of RMSE was obtained when the value of $K = 100$, suggesting that the model performs better in capturing detailed and precise information for both user and item embeddings. For other values of K, the RMSE increased, indicating a decline in the accuracy of movie representations. The plots provide a clear visualization of the convergence trends for both the training and testing datasets.

TABLE II. EFFECTIVENESS OF THE LDS-SRS FOR K@50/100

Paradigm	**Dataset (DS)**	**Assessment (RMSE)**	
		K@50	K@100
Suggested	1M	0.857	0.851
	10M	0.789	0.782
	Amazon Instant Video	1.101	1.083

Table III compares the prediction Effectiveness of the suggested paradigm for various vector sizes generated by the content embedding model. The findings show that the model achieves the best performance when the vector size is configured to 100. On the other hand, increasing the vector size beyond 100 does not enhance performance, as the model effectively captures the required semantic and topic information with a vector size of 100. Further increases in vector size result in a drop in performance, as larger vectors become sparse, leading to an uneven distribution of topic information. Fig. 4 shows the model's convergence for different content vector sizes over iterations. In addition, Fig. 5 illustrates the improvement in item recommendations, with both Precision and Recall steadily increasing. The Precision and Recall at 10 values for our paradigm are computed and compared with those of other leading models from existing literature.

Table IV presents the prediction errors of the proposed model alongside several other recommender systems [33], [20], and [32]. In [33], Wang et al. incorporated LDA into PMF, achieving RMSE values of 0.8969 on the '1M' collection, 0.8275 on the '10M collection', and 1.549 on the 'AIV' collection. In [20], SDAE was employed to learn item features within PMF, resulting in RMSE values of 0.8879 for '1M',

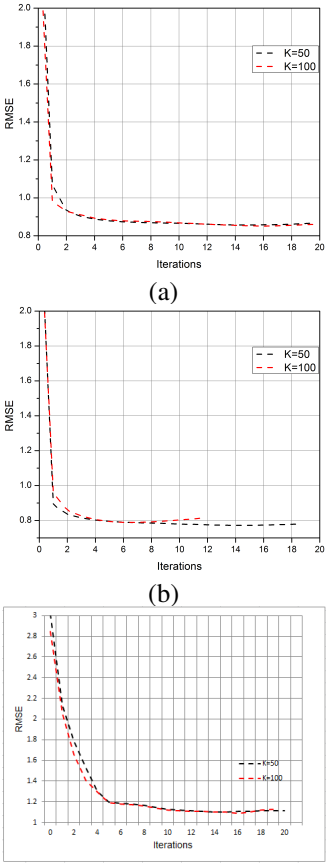

Fig. 3. LDS-SRS Performance with $K = 50$ and $K = 100$.

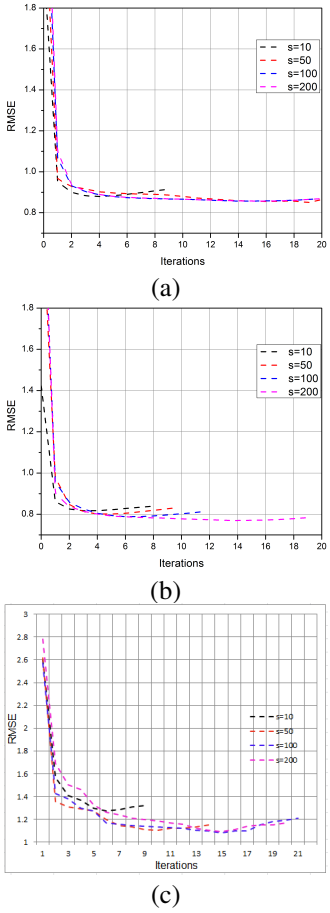

Fig. 4. LDS-SRS performance with different embedding sizes.

TABLE III. EVALUATION OF THE PROPOSED MODEL WITH VARYING VECTOR SIZES

Paradigm	Dataset (Ds)	Assessment (RMSE)			
		Size=10	Size=50	Size=100	Size=200
Suggested	1M	0.879	0.868	**0.856**	0.857
	10M	0.818	0.801	**0.782**	0.780
	Amazon Instant Video	1.12	1.119	**1.083**	1.094

0.8186 for '10M', and 1.3594 for 'AIV'. More recently, [32] combined contextual item information with PMF, enhancing RMSE to 0.853 for '1M', 0.795 for '10M', and 1.133 for 'AIV'. In [42], CNN was integrated with PMF to account for item statistics and Gaussian noise, yielding RMSE values of 0.847 for '1M', 0.784 for '10M', and 1.101 for 'AIV'. [45] introduced imputed data from similar neighbors into SVD, resulting in an RMSE of 0.850 for '1M'. [48] employed data clustering based on user/item similarity for recommender systems, achieving an RMSE of 1.0878 for '1M'. Lastly, [50] classified items and users into three distinct categories and influenced them with a Bhattacharya coefficient, obtaining an RMSE of 1.7000 for '1M'.

Table IV clearly demonstrates that the proposed model surpasses other recommender systems on the MovieLens and AIV datasets. Specifically, it achieves RMSE values of 0.846, 0.779, and 1.083 for the '1 Million', '10 Million', and 'AIV (Amazon Instant Video)' datasets, respectively, within a sparse user to item matrix, highlighting its superior performance.

Its runtime for rating predictions is significantly faster compared to other models.

TABLE IV. SUGGESTED PARADIGM ANALYSIS WITH HYBRID MODELS

Paradigm	ML-1 M	ML-10 M	AIV
CTR [33]	0.8969	0.8275	1.549
CDL [20]	0.8879	0.8186	1.3594
CMF [32]	0.853	0.795	1.133
CMF+ [32]	0.854	0.793	1.1279
RCMF [42]	0.847	0.784	1.101
RN [47]	0.863	0.807	1.105
ISVD [45]	0.850	-	-
GAGE [48]	1.0878	-	-
NCBR [50]	1.7000	-	-
Suggested	**0.846**	**0.779**	**1.083**

D. Complexity Analysis

The suggested paradigm demonstrates considerably lower computational complexity in comparison to other models. Unlike approaches like [32], which necessitate retraining the entire network and updating weights at each iteration, the proposed model operates without relying on neural networks.

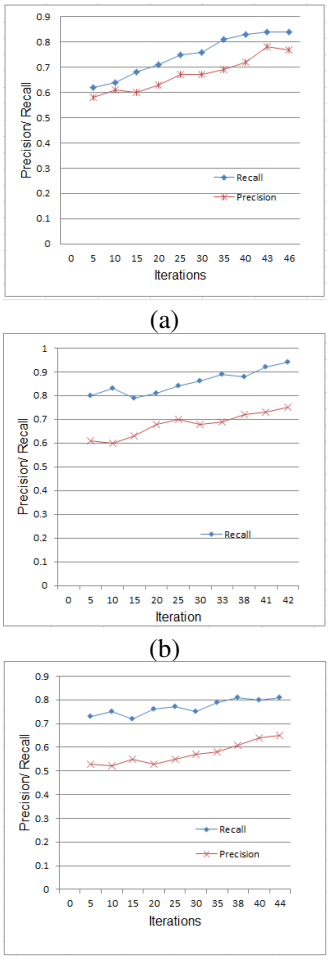

Fig. 5. Precision and recall for the proposed model.

This absence of neural network dependence leads to greater efficiency in its performance.

As shown in Table IV, our proposed LDS-SRS framework achieves a lower RMSE compared to CTR (0.8969 vs. 0.846), CDL (0.8879 vs. 0.846), and CMF (0.853 vs. 0.846) on the MovieLens 1M dataset. The key advantage of LDS-SRS lies in its ability to embed semantic and topic-driven representations within PMF, enhancing recommendation accuracy even in sparse user-item matrices. Unlike deep hybrid models that require high computational overhead, LDS-SRS efficiently integrates contextual knowledge while maintaining a lower computational cost.

In the outlined approach, movie plot vectors are computed a single time using a sophisticated content embedding model, with a computational complexity of $O(UIr + r^3)$, where $r = \min(U, I)$. These vectors are employed to set the item-specific latent factors within the PMF framework and are not subject to recalculation in subsequent iterations. The iterative updates for the latent factors U and I are performed with a complexity of $O(A^2\hat{R} + A^3U + A^3I)$, where $\hat{R}$ corresponds to the recorded ratings within the predicted feedback matrix, which are updated at every epoch. Consequently, the overall computational cost per iteration for the LDS-SRS model is $O(A^2\hat{R} + A^3U + A^3I) + UIr + r^3$, making it substantially more efficient compared to traditional recommender systems.

The proposed method exhibits outstanding efficiency with respect to time complexity. It employs a model for embedding content that integrates item semantics with topic information using LDA2VEC, which facilitates faster convergence of the collaborative filtering (CF) technique and improves overall accuracy. In contrast to neural network-based models that necessitate recalculating the content model in each iteration, this approach eliminates redundant computations, thus enhancing its efficiency.

E. Discussion

The results of this study demonstrate the effectiveness of LDS-SRS in mitigating the impact of data sparsity in recommender systems. By integrating deep semantic and topic-aware embeddings into PMF, our method enhances user-item interactions, leading to better rating predictions. Compared to existing hybrid recommendation approaches, our framework achieves higher accuracy while maintaining computational efficiency. This is particularly relevant for real-world applications where user data is often incomplete or sparse. Additionally, these findings align with prior research in matrix factorization-based recommendation (e.g. [20], [32], [42]), reinforcing the importance of leveraging textual semantics in recommendation tasks. Future work could further improve this approach by incorporating adaptive topic modeling techniques for dynamic recommendation scenarios.

V. Conclusion

This enhanced proposed approach and embedding model are incorporated into PMF to address the typical challenge of sparsity in recommender systems (RS). LDS-SRS predicts user ratings by utilizing item descriptions, with item representations generated through the embedding model that captures both semantic and topic data. These document representations are subsequently integrated into underlying factors of PMF, enhancing the accuracy of rating predictions. Our model was trained and evaluated using datasets from MovieLens and AIV, with movie plots serving as item descriptions. It can be adapted to other domains, such as e-commerce platforms like Amazon, Alibaba, and eBay, where product descriptions or user reviews can be processed using the enhanced LDA2VEC model and integrated into PMF for recommendation generation. It can be extended to other areas, including e-commerce platforms like Amazon, Alibaba, and eBay, where product descriptions or user reviews can be processed using the enhanced LDA2VEC model and integrated into PMF for recommendation generation. The integration of deep semantics and topic information significantly enhances the model's capability to provide more accurate rating predictions.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,"*IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734-749, Jun. 2005.

[2] P.G. Campos, F. Dez and I. Cantador, "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 67-119, Feb. 2005.

[3] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook", *In Recommender systems handbook,* New York, NY, USA: Springer, pp. 1-35, 2011.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer,* vol. 42, no. 8, Aug. 2009.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms", *In Proceedings of the 10th international conference on World Wide Web,* pp. 285-295, Apr 2001.

[6] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence.,* vol. 2009, pp. 4, Jan. 2009.

[7] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.,* pp. 501-508, 2006.

[8] M. Balabanovi, and Y. Shoham, "Fab: content-based, collaborative recommendation", *Communications of the ACM,* vol. 40, no. 3, pp.66-72, 1997.

[9] G. Chen, F. Wang, and C. Zhang, Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization, Information Processing and Management: an International Journal, 3, 368-379 (2009).

[10] K. Christidis and G. Mentzas, A topic-based recommender system for electronic marketplace platforms, Expert Systems with Applications, 11, 4370-4379 (2013).

[11] Aghdam, M.H., Analoui, M., Kabiri, P.: A novel non-negative matrix factorization method for recommender systems. Appl. Math. Inf. Sci. 9(5), 2721 (2015)

[12] T. Hofmann and J. Puzicha, Latent class models for collaborative filtering, Proceedings of the 16th international joint conference on Artificial intelligence, San Francisco, CA, USA, 688-693 (1999).

[13] L.H. Ungar and D.P. Foster, Clustering methods for collaborative filtering, Proceedings of the Workshop on Recommendation Systems, Menlo Park, CA, 1-16 (1998).

[14] J. Canny, Collaborative filtering with privacy via factor analysis, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 238-245 (2002).

[15] J.S. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, San Francisco, CA, USA, 43-52 (1998).

[16] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Information Retrieval, 2, 133-151 (2001).

[17] G.R. Xue et al., Scalable collaborative filtering using cluster-based smoothing, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 114-121 (2005).

[18] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, , 22(1):5–53, Jan. 2004.

[19] S. Li, J. Kawale, and Y. Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In , *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 811–820, New York, NY, USA, 2015. ACM

[20] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*, KDD '15, pages 1235–1244, New York, NY, USA, 2015. ACM.

[21] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends". In *F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender systems handbook,* pp. 73105. 2011

[22] G. Guo, J. Zhang and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start", *In Knowledge-Based Systems,* vol. 57, pp. 57-68, 2014.

[23] Y. Bao, H. Fang and J. Zhang, "TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation", In*AAAI*, Vol. 14, pp. 2-8, July. 2014.

[24] K. Zhou, S. H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation", In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 315-324, Jul. 2011.

[25] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize", *Lecture Notes in Computer Science,* vol. 5034, pp. 337-348, Jun. 2008.

[26] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing 13 (Proc. NIPS*2000)*. MIT Press, 2001.

[27] Hung-Hsuan Chen, Weighted-SVD: Matrix Factorization with Weights on the Latent Factors, *arXiv:1710.00482*, 2017.

[28] Wen, H., Ding, G., Liu, C., Wang, J.: Matrix factorization meets cosine similarity: addressing sparsity problem in collaborative filtering recommender system. In: Chen, L., Jia, Y., Sellis, T., Liu, G. (eds.) APWeb 2014. LNCS, vol. 8709, pp. 306–317

[29] A. Mnih, and R. R. Salakhutdinov, "Probabilistic matrix factorization", In *Advances in neural information processing systems,* pp. 1257-1264, 2008.

[30] H. Shan, and A. Banerjee, "Generalized probabilistic matrix factorizations for collaborative filtering", In *Data Mining (ICDM), 2010 IEEE 10th International Conference on,* pp. 1025-1030, Dec. 2010.

[31] R. Salakhutdinov, and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo", In *Proceedings of the 25th international conference on Machine learning,* pp. 880-887, Jul. 2008.

[32] Kim, D, Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 233–240. ACM (2016).

[33] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* KDD '11, pages 448–456. ACM Press, August 2011.

[34] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tang, Collaborative Filtering and Deep Learning Based Recommendation System For Cold Start Items, *Expert Systems With Applications*, KDD '15, pages 1235–1244, New York, NY, USA, 2015. ACM.

[35] G. Ling, M. R. Lyu, and I. King. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14*, pages 105–112, New York, NY, USA, 2014. ACM.

[36] J. McAuley and J. Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. *In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13*, pages 165–172, New York, NY, USA, 2013. ACM

[37] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. *In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ' 11, pages 448–456, ACM Press, August 2011.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing,* pages 306–351. IEEE Press, 2001.

[39] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR),* 12:2493–2537, Nov. 2011.

[40] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), June 2014.

[41] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Empirical Methods in Natural Language Processing (EMNLP),* pages 1746–1751, 2014.

[42] Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu. 2017. Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items. Information Sciences (2017).

[43] Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4, Article 19 (December 2015), 19 pages.

[44] C, Moody.: 2016. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec, arXiv:1605.02019v1 [cs.CL] 6 May 2016

[45] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data," *Knowledge-Based Systems, 2018*

[46] Farah Saleem, Naima Iltaf, Hammad Afzal and Mobeena Shahzad. "Using Trust in Collaborative Filtering for Recommendations." *In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 214-222. IEEE, 2019.

[47] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo. *RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems* ACM International Conference on Information and Knowledge Management October 22 to 26, 2018, Torino, Italy. ACM, New York, NY, USA, 10 pages.

[48] T. Mohammadpoura, A. M. Bidgolia, R Enayatifarb and H S Javadi "Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm" *Genomics, https://doi.org/10.1016/j.ygeno.2019.01.001* 2019

[49] Anwar, Fahad, Naima Iltaf, Hammad Afzal, and Haider Abbas. "A Deep Learning Framework to Predict Rating for Cold Start Item Using Item Metadata." *In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 313-319. IEEE, 2019.

[50] Bag, S., Kumar, S., Awasthi, A., and Tiwari, M. K. (2019). "A noise correction-based approach to support a recommender system in a highly sparse rating environment." *Decision Support Systems, 118, 46–57. doi:10.1016/j.dss.2019.01.001* (2019)

[51] F Anwaar, N Iltaf, H Afzal, and R Nawaz (2018). "HRS-CE: a hybrid framework to integrate content embeddings in recommender systems for cold start items" *Journal of Computational Science 2018)*. doi:10.1016/j.jocs.2018.09.008

[52] Z Khan, N Iltaf, H Afzal and H Abbas "Enriching Non-negative Matrix Factorization with Contextual Embeddings for Recommender Systems" *Neurocomputing (2019)* doi: https://doi.org/10.1016/j.neucom.2019.09.080

[53] Xiangyong Liu, Guojun Wang, and Md Zakirul Alam Bhuiyan, "The Strength of Dithering in Recommender System," *Proc. of the IEEE 14th International Symposium on Pervasive Systems, Algorithms, and Networks (IEEE I-SPAN 2017)*, Exeter, UK, Jun 23-27, 2017

[54] Abdullah Al Omar, Rabeya Bosri, Mohammad Shahriar Rahman, Nasima Begum and Md Zakirul Alam Bhuiyan, "Towards Privacy-preserving Recommender System with Blockchains," *Proc. of the 5th International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications (DependSys 2019)*, Guangzhou, China, November 12-15, 2019