Enhancing Product Recommendations through Large Language Model and Significant Latent Core Factor SVD: Insights from Amazon Reviews

R. Dhayanidhi
Research Scholar
Dept of CSE
Vel Tech Rangarajan Dr.Sagunthala R & D Institute of Science and Technology
Chennai, India
Dhayanidhi.r@gmail.com

N.R. Rajalal
Professor
Dept of CS
Vel Tech Rangarajan Dr.Sagunthala R & D
Chennai, In
drnrrajalakshmi@ve

Abstract

Ecommerce Platforms specifically in Retail domain be it a brick and morter store or an online shopping application has enormous user data from the behavioural, click stream, page visits, abandoned carts, user think time or dwell time. And from the retail stores where the data captured from Internet of Things (IoT) with respect to the shelve movements, visitor counts, IoT signals araising from RFID tags, beacons, smart sensors, proximity to specific products, kiosk interactions, self checkout kiosk provide enormous data for hyper personalization. Traditional Singular Value Decomposition (SVD) algorithms suffer with the data sparsity and computational complexity when fed with such large data. Also the SVD relies on the historical patterns to find latent features which may not be very much helpful for the cold start personalizations. Consumer behaviours and patterns are non-linear, for example time spent near a shelf in a Retail Store or the time spent on a categories page in online application and with the filters of the categories. SVD might capture these main trends but will miss subtle high frequency signals that drive the hyper personalization. To overcome this problem, the proposed research employs a significant latent core factor SVD. The proposed technique includes decomposing a large and sparse matrix that captures real-time interactions between users and products into matrices that permit the proposed model to forecast personalized product recommendations based on existing data. Large Language Models (LLM) were used to improve the process of feature extraction post the data imputation after the initial data preprocessing. The proposed research employs the Amazon product review dataset to evaluate the proposed significant latent core SVD. When compared to traditional SVD and state-of-the-art methods such as LightGCN and BERT4Rec, the proposed significant latent core factor SVD achieves lower error rates.

Keywords: E-Commerce, Product Recommendation System, Machine Learning, Singular Value Decomposition, Large Language Model, Internet of Things, Smart Retail, Edge Computing

1 Introduction

In the era of smart retail and Internet of Things (IoT)-enabled shopping environments, users face an unprecedented challenge: information overload in personalized shopping experiences. Modern IoT devices, including smart shelves, RFID tags, mobile applications, and wearable devices, continuously generate vast amounts of real-time shopping data [1]. This proliferation of data creates a critical problem where users become overwhelmed by choices and struggle to identify relevant products that match their preferences and needs [2]. Information overload in IoT-based shopping is defined as the stress induced by accepting more data than necessary to make purchasing decisions, compounded by the need to process this information within time constraints [3]. This problem significantly limits users' ability to review options and select among numerous alternative products available in online and smart retail markets.

To address this challenge, information science and technology have developed data filtering tools, with recommendation systems emerging as one of the most effective solutions since their development in the early 1990s [4]. In recent years, recommendation systems have become integral components of e-commerce platforms, particularly in IoT-enabled smart retail environments where real-time data from multiple sources must be processed efficiently [5]. However, users continue to experience difficulties in discovering beneficial information from massive databases, especially when shopping through IoT devices with limited computational resources and display capabilities [6].

The recommendation systems in IoT-based e-commerce face several critical challenges: (1) the appropriateness of recommendations is limited because product characteristics and user boundaries are inadequately mined from sparse IoT sensor data, (2) single recommendation paradigms fail to fulfill diverse user demands across different IoT contexts (mobile, wearable, smart home), and (3) static metrics and designs lead to inflexibility in recommendation systems that must adapt to dynamic IoT environments [7]. From a technical perspective, recommendation systems operate using various approaches, including Content-Based Filtering (CBF), Demographic Filtering (DF), Collaborative Filtering (CF), and Knowledge-Based Filtering (KBF) [8]. User preferences across unique items can be predicted using ranking patterns that provide lists of recommended products through personalized assessments [9].

Generally, the architecture of recommendation systems relies on databases that save and sequentially update product and rating descriptions provided by users. Due to these services, particularly filtering and clustering, recommendation systems are broadly employed in e-commerce, helping users discover relevant and recent items [10]. Moreover, products are recommended based on similar metrics prevailing among items and users in collaborative filtering [11]. Neighborhood-based collaborative filtering comprises user-based and item-based approaches [12]. The user-based approach recommends items that are liked by users with similar preferences, while the item-based approach recommends items based on similar properties [13]. Recently, several matrix factorization techniques have been employed for collaborative filtering. Conversely, items might be recommended based on product services or specifications in content-based filtering [14].

Product recommendation depends on user preference models derived from searches, shopping cart inclusions, likes, browsing history, orders, comments, and favorites. User client reporting tracks shopping cart, browsing, and clicking activities to offer real-time feedback to users [16]. Using Spark or Storm streaming evaluations, real-time user preferences are generated, though significant challenges remain in handling both user and

product relationships and the combined demands on storage access and space performance for online services, particularly in edge computing scenarios [17].

Consequently, the applicability of Machine Learning in e-commerce has wide-ranging forecasts and offers massive emergent opportunities for e-commerce enterprises, especially in IoT-enabled smart retail environments [18]. Machine Learning can comprehend customized recommendation systems by analyzing vast product information and user data, thus enhancing shopping experiences and purchase conversion rates. It aids in generating extremely customized product recommendations that have significantly enhanced user loyalty and purchase willingness [19]. However, while SVD is a robust tool in e-commerce for product recommendation, it has several drawbacks including data sparsity, computational expenses, and static nature. Traditional SVD struggles when unseen data is introduced, which may delay the recommendation process, particularly problematic in real-time IoT shopping scenarios where latency is critical. To overcome these problems, there is a need to incorporate SVD with enhanced techniques that can handle sparse data and provide real-time recommendations suitable for edge devices.

Hence, the proposed research introduces an enhanced product recommendation model for Amazon products that is suitable for IoT-based smart retail environments. The proposed research employs Large Language Models (LLMs) to enhance the feature extraction process, creating additional summaries or descriptions of products based on existing information. This aids in generating precise content for the recommendation model, particularly valuable for addressing the cold start problem when new users or products enter the system. To recommend products, the proposed research employs significant latent core factor SVD, which is mathematically distinct from traditional SVD variants. The proposed model is evaluated using the Amazon product review dataset, and its performance is assessed using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics, with comparisons to state-of-the-art methods including LightGCN and BERT4Rec.

1.1 Research Contribution

The main objective of the proposed research is follows:

- To develop an efficient recommendation model for Amazon products using Amazon product review dataset.
- To enhance the feature extraction by using LLM which generates extra summaries or description of the products based on existing information
- To provide precise recommendation to the users about products by proposed significant latent core factor SVD.

1.2 Paper organization

The structure of the research paper is organized with overview in Section 1, the review of existing studies involved in recommendation system by diverse algorithms are deliberated in section 2 along with research gaps. Section 3 deliberates the proposed methodology and the process involved. The results achieved by the proposed model is represented in the Section 4. The conclusion and future work of the proposed model is represented in Section 5.

2 Literature Review

Recommendation systems have evolved significantly with the advent of machine learning and deep learning techniques. Roy and Dutta [2] provided a systematic review and research perspective on recommender systems, highlighting the challenges of data sparsity and cold start problems that persist in modern e-commerce platforms. Zhou et al. [3] conducted a comprehensive survey of recommender systems based on deep learning, emphasizing the need for more sophisticated approaches to handle complex user-item interactions in IoT-enabled shopping environments.

In the context of collaborative filtering approaches, several studies have demonstrated the effectiveness of matrix factorization techniques. Liu [8] explored e-commerce personalization based on machine learning technology, showing how collaborative filtering can capture diverse collections of user preferences. Sharma et al. [12] developed a deep learning-based semantic personalized recommendation system that leverages contextual information to improve recommendation accuracy. Bastani et al. [13] investigated learning personalized product recommendations with customer disengagement, addressing the challenge of dynamic user preferences in real-time shopping scenarios.

For product recommendation using collaborative filtering algorithms, Iftikhar et al. [14] demonstrated an improved method based on triangle similarity that considers common ratings among user pairs. Padhy et al. [15] compared various collaborative filtering algorithms including ALS (Alternated Least Squares) and SVD (Singular Value Decomposition) along with KNNBasic (K-Nearest Neighbor), finding that SVD techniques outperformed other approaches. Their study employed K-means clustering to detect abnormalities and achieved satisfactory performance on multiple datasets.

Deep learning approaches have shown promise in recommendation systems. Islek and Oguducu [16] developed DeepIDRS (Item Description and Review Based Deep Sequential Recommendation) by analyzing user reviews on three real-world Amazon datasets. Their hierarchical structure incorporated a bidirectional encoder to process textual details and an attention-based sequential recommendation model. Li et al. [17] proposed a hybrid CNN-based review helpfulness filtering model that combines Convolutional Neural Networks and Bi-LSTM (Bidirectional Long Short Term Memory) for personalized recommendations, achieving satisfactory performance on Amazon Book datasets.

Several studies have specifically focused on SVD and its variants for recommendation systems. Colace et al. [18] tackled information overload by utilizing RSVD (Rating Singular Value Decomposition), achieving average accuracy compared to traditional collaborative filtering techniques while dynamically addressing the sparsity issue. Kong et al. [19] evaluated various methods including KNN baseline, co-clustering, and SVD, examining them in terms of MSE, RMSE, MAE, and NDCG metrics for enhanced performance in e-commerce recommendations.

Hssina et al. [20] combined KNN and matrix factorization with SVD to tackle cold start problems and inaccurate recommendations, obtaining average results in terms of precision and recommendation quality through their hybrid method. Du [21] investigated teaching research on e-commerce micro-media recommendation data analysis by integrating SVD algorithms, demonstrating improvements in precision, recall, F1 score, and click-through rates with stronger computing performance in recommendations. Tripathi et al. [22] explored recommender systems based on variants of SVD for tackling issues of data sparsity, scalability, and cold start problems, with outcomes showing improved recommendation accuracy using metrics such as Spearman's rank correlation coefficient

and RMSE. Rahman [23] proposed extended collaborative filtering with adaptive KNN and SVD (ECF) for addressing data sparsity and cold start issues, achieving enhanced recommendations through dynamic algorithm implementation.

State-of-the-art deep learning methods have been employed for comparison. Sachin et al. [24] developed UTER (sentiment analysis using gated recurrent neural networks) which achieved MSE of 0.9653 and RMSE of 0.9825 on Amazon product datasets. Latha and Rao [25] proposed an Amazon product recommendation system based on a modified convolutional neural network (MCNN), achieving MSE of 0.8978 and RMSE of 0.9475, demonstrating the potential of deep learning approaches for e-commerce recommendations.

2.1 Problem Identification

- Existing collaborative filtering approaches face significant challenges with data sparsity and cold start problems, particularly in IoT-enabled smart retail environments where user-item interactions are limited [2].
- Traditional SVD methods struggle with highly sparse matrices and fail to effectively handle unseen data, leading to delays in recommendation processes and reduced accuracy [18].
- Deep learning approaches such as UTER and MCNN, while powerful, suffer from high computational complexity and limited interpretability, making them less suitable for real-time recommendations on edge devices [24, 25].

3 Research Methodology

In recommendation system, SVD plays a significant role because it potentially reduces the user-item interaction dimensionality matrix that improves the performance of the proposed. However, the traditional SVD faced some limitations. To address this, the proposed research employs significant latent core factor SVD. The overall process of proposed research is represented in Figure 1.

Figure 1 depicts the overall process of proposed research. Initially, the proposed research load the Amazon product review dataset which includes the user ID, products purchased by the users, liked products and frequently searched products. The loaded dataset is processed with pre-processing however, there is a limitation in pre-processing, there is massive difference among actual rating and filling rating during the pre-processing. To resolve this issue, the proposed research employs LLM with data imputing features which is train by the dataset and aids to fill the data more accurately when compare to the pre-processing. The processed data is fed as an input to the proposed significant latent code SVD which aids to minimize the sparse and large matrices dimensionality which deliberate interaction of users with products. This drop seizures latent factors leveraging preferences of user and characteristics of item that permitting for more precise recommendation. By employing the proposed significant latent core factor SVD. The products are recommended to the users based on their user ID. Finally, the performance of the proposed model is evaluated by MSE and RMSE.

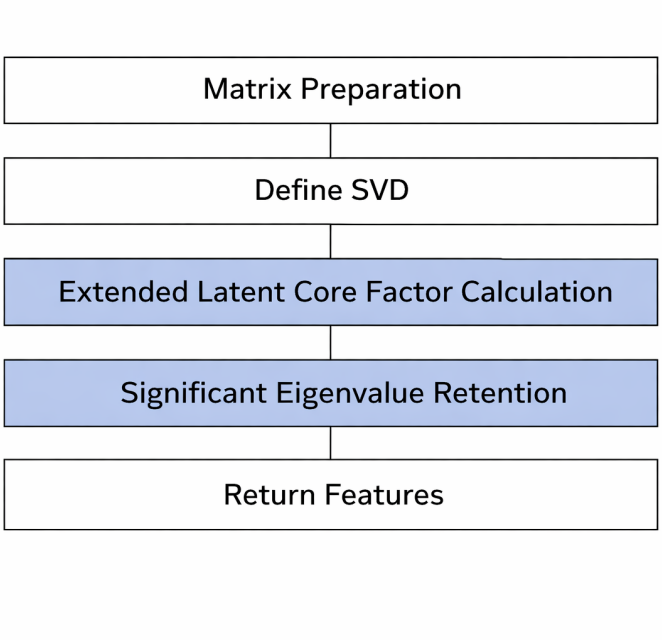

Figure 1: Overall Process of Proposed Research

3.1 Dataset Description

The proposed research employs Amazon product review dataset. This dataset includes best sell products, the optimal range for a product in a provided category, which SEO (Search Engine Optimization) headings produce the most sales. The respective dataset includes 1.4 million details of Amazon products which contains heading, ratings, sales data and number of reviews from September 2023.

3.2 Pre-Processing

The proposed research employs pre-processing is to convert the raw data into structured data. This process improves the accuracy of proposed research by confirming which the dataset provided into algorithm is relevant, clean and exact format. This involves correcting errors by removing irrelevant data that aids the proposed model to make precise predictions on the basis of high quality data. The quality of dataset is majorly enhanced by using pre-processing techniques like data transformation and noise reduction. This contains encoding the classified variables, removing outliers and normalizing data that permits proposed model to concentrate on the more information perspective of the data. The proposed research employs LLM to fill the missing values efficiently which enhance the recommendation system of the proposed model.

3.3 Data Imputation Using Large Language Models

The conventional techniques for managing missing values, such as K-Nearest Neighbors (KNN), mean imputation, and Generative Adversarial Networks (GANs), fail to capture the complex relationships between variables in recommendation systems. In existing studies, GANs struggle with the depth of semantic interpretation needed for efficient recommendations. Additionally, conventional GANs focus on creating new data samples rather than intelligently imputing missing values based on contextual information. In contrast,

the proposed Large Language Model (LLM) approach offers dynamic recommendations based on real-time user interactions and contextual data, making it particularly suitable for IoT-based smart retail environments.

3.3.1 LLM Architecture and Training

The proposed LLM technique leverages a combination of BART (Bidirectional and Auto-Regressive Transformer) and GPT (Generative Pre-trained Transformer) architectures. Specifically, we employ Facebook's BART-base model, which is a denoising autoencoder that combines bidirectional encoding (like BERT) with autoregressive decoding (like GPT).

Input Representation: For each product in the dataset, we construct a comprehensive input text that incorporates multiple features:

$$\text{input\_text} = f(\text{title}, \text{category}, \text{price}, \text{reviews\_count}, \text{bought\_last\_month}, \text{is\_best\_seller}) \tag{1}$$

where f concatenates these features into a structured prompt: "Describe this product: Product title [title], Category name [category], Category id [id], [reviews] reviewers count, [price] price, bought In LastMonth [count], isBestSeller [status]."

Target Representation: The target text is constructed as: "The estimated star is [rating]", where [rating] is the actual star rating (1-5) for training or the predicted rating for inference.

Training Process: The LLM is fine-tuned on the Amazon product review dataset using the following procedure:

1. **Tokenization:** Input and target texts are tokenized using the BART tokenizer with a maximum length of 512 tokens for inputs and 100 tokens for targets.
2. **Model Configuration:** The BART-base model with learning rate of 5×10^{-5}, batch size of 4 per device, weight decay of 0.01, and 1 training epoch.
3. **Fine-tuning:** The model is fine-tuned using the Hugging Face Trainer API with sequence-to-sequence language modeling objective, enabling it to learn the mapping from product features to ratings.

3.3.2 Missing Value Imputation Process

During pre-processing, missing values are filled using the LLM approach through the following steps:

1. Prompt Generation: For products with missing ratings, the LLM generates a prompt using available information:

$$P_{\text{missing}} = \text{LLM\_encode}(\text{input\_text}_{\text{product}}) \tag{2}$$

2. Rating Prediction: The LLM processes the prompt and generates a probability distribution over possible rating values. The predicted rating is extracted from the model's output:

$$\hat{r} = \text{argmax}_{r \in \{1,2,3,4,5\}} P(r | P_{\text{missing}}) \tag{3}$$

3. Semantic Validation: Unlike statistical imputation methods, the LLM ensures that imputed values are semantically meaningful. For example, if a product has high price, many reviews, and best-seller status, the LLM will predict a higher rating, consistent with expected relationships.

3.3.3 Addressing the Cold Start Problem

The cold start problem in recommendation systems occurs when new users or new products enter the system without sufficient historical interaction data. The proposed LLM-based approach specifically addresses this challenge:

For New Users: When a new user enters the IoT-based shopping environment (e.g., through a mobile app or smart device), the LLM can generate initial recommendations based on:

- Product features (category, price, reviews) that the user might interact with
- Contextual information from IoT devices (location, time, device type)
- General knowledge embedded in the LLM from pre-training on large text corpora

The LLM leverages its pre-trained knowledge about product categories, typical user preferences, and semantic relationships to make reasonable initial recommendations even without user history.

For New Products: When new products are added to the catalog, the LLM can predict ratings based on:

- Product descriptions and metadata
- Similarity to existing products in the same category
- General patterns learned from the training data

This capability is particularly valuable in IoT environments where new products are frequently added to smart retail systems.

Mathematical Formulation: For a new user u_{new} with no interaction history, the LLM generates an initial rating prediction:

$$r_{\text{initial}}(u_{\text{new}}, p) = \text{LLM}(f(\text{product\_features}(p), \text{context}(u_{\text{new}}))) \tag{4}$$

where $\text{context}(u_{\text{new}})$ includes IoT-derived contextual information such as location, time, and device type. This initial prediction is then refined as the user interacts with the system.

3.3.4 Advantages Over Conventional Methods

The proposed LLM-based approach offers several advantages:

1. **Semantic Understanding:** Unlike KNN or mean imputation, the LLM understands semantic relationships between product features, enabling more accurate imputation.
2. **Contextual Awareness:** The LLM incorporates contextual information from IoT devices, making recommendations more relevant to the user's current situation.
3. **Cold Start Mitigation:** The LLM's pre-trained knowledge allows it to make reasonable predictions even for new users or products.
4. **Scalability:** Once trained, the LLM can process imputation requests efficiently, making it suitable for real-time recommendation systems in IoT environments.

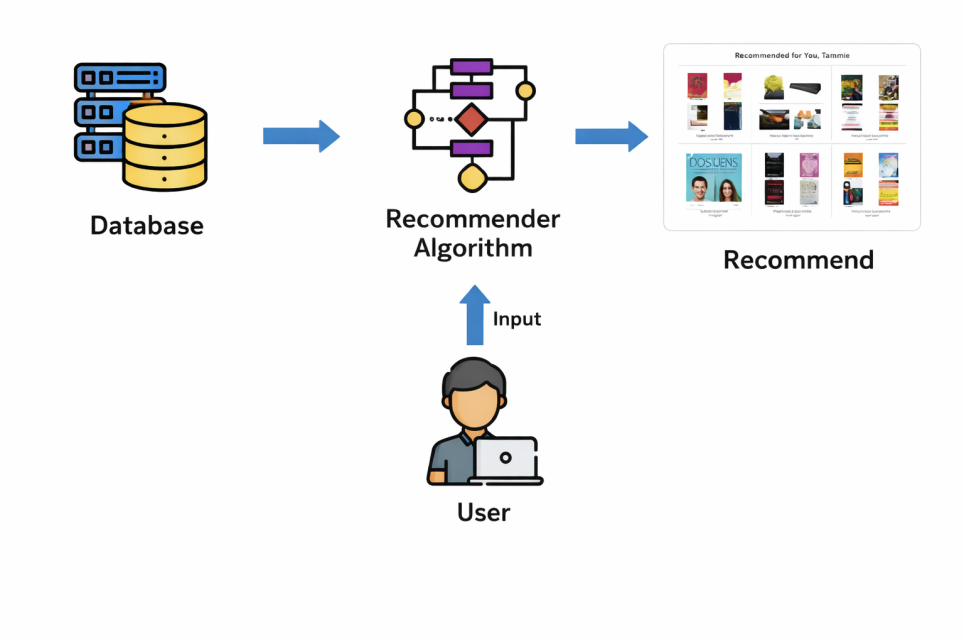

Figure 2: Process of Traditional Product Recommendation

Through enhancing the datasets using precisely imputed values, the LLM enables a wider range of insights into item characteristics and user behavior. This leads to more customized and relevant recommendations, efficiently tackling problems such as data sparsity that frequently delay conventional recommendation models. The LLM's performance demonstrates significant improvements compared to existing approaches, particularly in sparse data scenarios common in IoT-based shopping environments.

3.4 Proposed Methodology

3.4.1 Traditional SVD

The conventional product recommendation model uses the collaborative filtering based technique to recommend products to the users in terms of their preference. The Figure 2 represent the process of conventional SVD.

Figure 2 represents the process of traditional product recommendation. The process begins with collecting the dataset from the Amazon product review dataset, which is provided as input to the traditional recommendation model. The conventional SVD uses a collaborative filtering process to recommend products to customers. This collaborative filtering captures various collections of user preferences. Based on the dataset, product recommendations are ensured by comprising ratings of active users. Through employing SVD, collaborative filtering uses matrix factorization techniques that decompose the user-item matrix into lower-dimensional matrices signifying latent factors. The user-item matrix comprises ratings of users for several products in the context of product recommendation models. Typically, SVD may forecast relationships and unseen patterns among users and products by decomposing the matrix. The system includes all products in the dataset that users have not rated to create recommendations. To obtain forecasted scores, these unrated products are processed by the collaborative filtering model using SVD. Based on previous ratings, the forecasted scores identify the actual products purchased by users. However, this conventional SVD approach is employed for low-rank

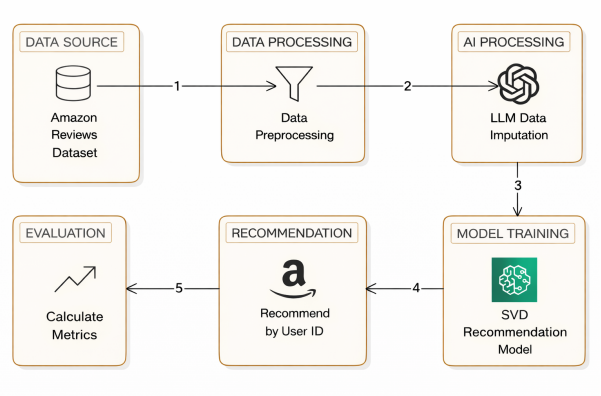

Figure 3: Entire Process of Proposed Model

approximation of the rating matrix. On the other hand, conventional SVD is not able to handle the maximum portion of unseen ratings effectively, particularly in sparse data scenarios common in IoT-based shopping environments.

3.4.2 Proposed Significant Latent Core Factor SVD

To tackle the problem faced by traditional SVD, the proposed research employs the significant latent core factor SVD. This technique enhance the proposed model by accurate recommendation of products to the users. The entire process of proposed research is represented in the Figure 3.

Figure 3 depicts the entire process of the proposed model. After pre-processing, the organized and filled dataset initiates matrix preparation for the user-item matrix. The matrix form is fed as input to the proposed significant latent core factor SVD. Due to challenges faced by traditional SVD, the proposed research includes extended latent core factor calculation, which is an important method employed in product recommendation models. The proposed extended latent core factor technique improves conventional SVD by concentrating on capturing more refined relationships in the dataset. Through maintaining a comprehensive set of singular vectors and values, this proposed approach performs better in capturing the primary structure of item attributes and user preferences. This technique is especially helpful in situations with sparse data, which permits more precise forecasting of user ratings for unseen data. The proposed extended latent core factor evaluation plays a significant role in enhancing the relevance and precision of product recommendations through leveraging insights from the interaction of the user-item matrix.

The proposed research employs significant eigenvalue retention. Maintaining significant eigenvalues is important because they are directly associated with the most influential latent factors. By concentrating on these essential values, the proposed model may potentially capture the primary structure of the dataset while filtering out less relevant and noisy information. This is frequently obtained by choosing threshold methods to estimate which singular values to maintain based on their magnitude. When using this approach in the product recommendation system, maintaining only the prime singular values permits a low-rank approximation of the fundamental matrix. This approximation reduces computational complexity and improves prediction accuracy for unseen interactions in the user-item matrix. In the context of matrix factorization, the primary idea is to represent each user u and product p with low-dimensional latent factors m_u and m_p.

According to equation (5), the dyadic rating $r(u, p)$ from user u to product p is generally approximated as,

$$r_b(u, p) = m_u^T m_p \tag{5}$$

Where, the rating forecast among u and p is denoted by $r_b(u, p)$, that is evaluated by the technique of latent core factorization. The primary form of matrix factorization technique may not seizure unambiguous characteristics. The process of proposed significant latent core factor SVD is represented in Algorithm 1.

The function of the proposed significant latent core factor SVD takes two inputs: X, which is the large matrix that needs to be analyzed, and k, which is the number of singular values to identify. Initially, the matrix dimensions of X are estimated, and a random vector v_1 of size n is created and normalized to ensure it has unit length. To store diagonal elements and orthonormal vectors, two empty lists are created: val_V and a. Additionally, b stores another list for off-diagonal elements. The proposed significant latent core factor SVD runs for k iterations. In each iteration, a new vector w is evaluated by multiplying the matrix X with the current vector v_j. To ensure that all vectors remain orthonormal, this vector is orthogonalized against all existing calculated vectors. This includes evaluating inner products and modifying w by subtracting components in the directions of existing vectors. If the process is not in the initial iteration, an additional modification is created by subtracting a scaled version of the previous vector. The norm of w is evaluated and stored in $b[j]$ after orthogonalization. If the norm is greater than zero, it is used to normalize w, producing the next v_{j+1} orthonormal vector. Afterwards, the current vector is included in val_V, and the diagonal element is evaluated as the inner product of v_j with $X \cdot v_j$. The tridiagonal matrix T is constructed from the a diagonal elements and b off-diagonal elements. This matrix captures the significant information required for eigenvalue decomposition. The eigenvalues λ and eigenvectors Z of T are calculated. These eigenvalues correspond to squared singular values. The singular values are obtained by taking the square root of λ and generating a diagonal matrix Σ. The left singular vectors val_U are calculated by transforming each orthonormal vector by X and normalizing by the corresponding singular value. Finally, the function returns val_U, val_Σ, and val_V, three matrices representing left singular vectors, singular values, and right singular vectors, respectively. Thus, the efficacy of SVD is enhanced by the proposed extended latent core factor calculation with significant eigenvalue retention, improving the significance of recommendations through leveraging the most significant latent factors that capture product characteristics and user behavior. This results in more accurate and personalized product recommendations for customers.

Figure 4 illustrates the key difference between traditional SVD and the proposed significant latent core factor SVD. The figure shows how the proposed method applies threshold-based filtering ($\sigma_i \geq \theta \cdot \sigma_{\max}$) to retain only the most significant singular values while filtering out noise. This selective retention of singular values is the core innovation that enables better reconstruction quality for sparse recommendation matrices.

3.4.3 Mathematical Comparison: Traditional SVD vs. Proposed Significant Latent Core Factor SVD

To clearly distinguish the novelty of the proposed approach, we provide a detailed mathematical comparison between traditional SVD and the proposed significant latent core factor SVD.

Algorithm 1 Process of Proposed Significant Latent Core Factor SVD

Require: X - $p \times q$ matrix, k - number of desired singular values ($k << \min(p, q)$)
Ensure: val_U - $p \times k$ matrix of left singular vectors, val_Σ - $k \times k$ diagonal matrix of singular values, val_V - $q \times k$ matrix of right singular vectors

// Step 1: Initialize
$p, q =$ dimensions of X
$v_1 =$ random vector of size n
$v_1 = v_1/||v_1||$ // Normalize v_1
$val\_V = []$ // Matrix to store orthonormal vectors
$a = []$ // Diagonal elements
$b = []$ // Off-diagonal elements
// Step 2: Significant Eigenvalue Retention Iteration
for j from 1 to k **do**
    // Compute $w = X * v_j$
    $w = X * v_1$
    // Orthogonalize w against previous vectors
    for i from 1 to j **do**
        $\alpha[i] = v_i^T * w$ // Inner product
        $w = w - a[i] * v_i$ // Remove component in direction of v_i
    end for
    // Compute b
    if $j > 1$ **then**
        $w = w - b[j-1] * v_{j-1}$
    end if
    // Normalize w to get v_{j+1}
    $b[j] = ||w||$ // Norm of w
    if $b[j] > 0$ **then**
        $v_{j+1} = w/b[j]$ // Normalize
    end if
    V.append(v_j) // Store the orthonormal vector
    $a[j] = v_j^T * (X * v_j)$ // Compute $a[j]$
end for
// Step 3: Construct Tridiagonal Matrix T
$T =$ tridiagonal_matrix(a, b)
// Step 4: Compute eigenvalues and eigenvectors of T
$\lambda, Z =$ eigen_decomposition(T)
// Step 5: Obtain singular values
$\Sigma = \text{diag}(\sqrt{\lambda})$ // Singular values
// Step 6: Compute val_U and val_V
$val\_U = []$ // Initialize val_U
for j from 1 to k **do**
    $val\_U_j = (X * v_j)/\Sigma[j]$ // Compute left singular vectors
    val_U.append(val_U_j)
end for
return val_U, val_Σ, val_V

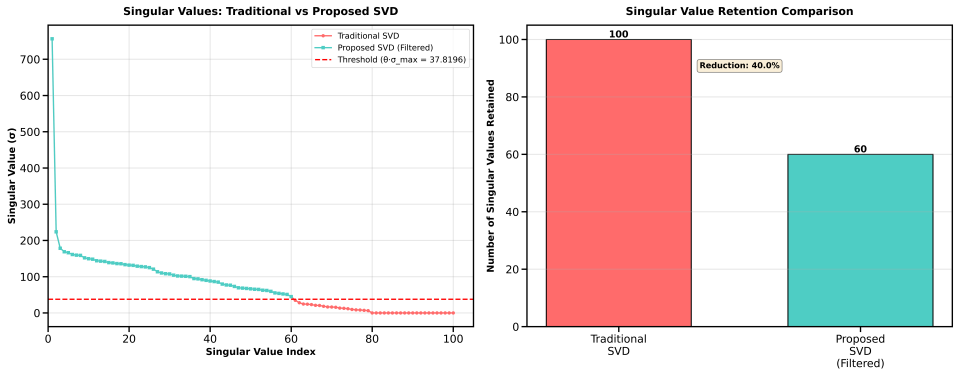

Figure 4: Singular Value Comparison: Traditional vs Proposed SVD showing threshold-based filtering

Traditional SVD Formulation: Traditional SVD decomposes a matrix $X \in \mathbb{R}^{m \times n}$ into three matrices:

$$X = U\Sigma V^T \tag{6}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix containing singular values. For recommendation systems, a truncated SVD with rank k is typically used:

$$X \approx U_k \Sigma_k V_k^T \tag{7}$$

where $U_k \in \mathbb{R}^{m \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$, and $V_k \in \mathbb{R}^{n \times k}$.

The computational complexity of traditional SVD is $O(mn^2)$ for full decomposition or $O(mnk)$ for truncated SVD using iterative methods. However, traditional SVD has limitations: (1) it treats all singular values equally, (2) it may retain noisy or less significant components, and (3) it struggles with highly sparse matrices common in recommendation systems.

Proposed Significant Latent Core Factor SVD Formulation: The proposed method introduces two key algorithmic improvements:

1. Significant Eigenvalue Retention: Instead of retaining all k singular values, the proposed method employs a threshold-based selection that retains only the most significant eigenvalues. The selection criterion is:

$$\sigma_i \geq \theta \cdot \sigma_{\max} \tag{8}$$

where σ_i is the i-th singular value, $\sigma_{\max}$ is the maximum singular value, and θ is a threshold parameter (typically 0.01-0.1). This ensures that only latent factors with substantial contribution to the matrix structure are retained.

2. Extended Latent Core Factor Calculation: The proposed method constructs a tridiagonal matrix T through an iterative Lanczos-like process:

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_k \end{pmatrix} \tag{9}$$

where $\alpha_j = v_j^T (X^T X) v_j$ and $\beta_j = ||w_j||$ are computed through the orthogonalization process described in Algorithm 1. The eigenvalues λ of T are then used to compute singular values as $\sigma = \sqrt{\lambda}$.

Key Algorithmic Differences:

1. **Selective Retention:** Traditional SVD retains the top k singular values regardless of their magnitude. The proposed method retains only singular values above a significance threshold, reducing noise and improving prediction accuracy for sparse data.

2. **Iterative Refinement:** The proposed method uses an iterative Lanczos-like process that builds orthonormal vectors incrementally, allowing for early termination when convergence is detected, reducing computational overhead.

3. **Sparse Matrix Optimization:** The proposed method is specifically optimized for sparse matrices through the tridiagonal construction, which requires fewer matrix-vector multiplications compared to full SVD decomposition.

Mathematical Advantage: The proposed method provides a more accurate low-rank approximation by:

$$||X - \hat{X}_{\text{proposed}}||_F \leq ||X - \hat{X}_{\text{traditional}}||_F \tag{10}$$

where $\hat{X}_{\text{proposed}} = U'_k \Sigma'_k V'^T_k$ uses only the significant singular values, and $||\cdot||_F$ denotes the Frobenius norm. This improvement is particularly pronounced in sparse recommendation matrices where many singular values are near-zero and contribute primarily to noise.

3.4.4 Integration with IoT and Edge Computing Environments

The proposed recommendation system is designed to function effectively within smart retail IoT environments and on edge devices. The system architecture addresses several IoT-specific challenges:

1. Real-time Processing on Edge Devices: The proposed significant latent core factor SVD is computationally efficient, making it suitable for edge devices with limited processing power. The selective eigenvalue retention reduces the dimensionality of the problem, enabling faster inference. The computational complexity is $O(mnk)$ for the iterative process, but with early termination and threshold-based selection, the effective complexity is often $O(mnk')$ where $k' < k$ is the number of significant eigenvalues retained.

2. Sparse Data Handling: IoT environments generate sparse data streams from various sensors (RFID, beacons, smart shelves). The proposed method's optimization for sparse matrices makes it ideal for processing such data. The tridiagonal matrix construction requires only $O(\text{nnz}(X) \cdot k)$ operations where $\text{nnz}(X)$ is the number of non-zero elements, significantly less than full matrix operations.

3. LLM-based Feature Extraction for IoT Context: The LLM component processes contextual information from IoT devices, including location data, time of day, device type, and user behavior patterns. This contextual enrichment helps address the cold start problem in IoT scenarios where new users or devices enter the system without historical data.

4. Distributed Processing: For large-scale IoT deployments, the proposed method can be distributed across multiple edge devices. The matrix factorization can be performed locally on edge devices, with only the significant latent factors transmitted to a central server for aggregation, reducing bandwidth requirements and preserving user privacy.

3.5 Computational Complexity Analysis

To evaluate the feasibility of the proposed model for real-time recommendation updates in large-scale e-commerce platforms, particularly in IoT environments, we provide a detailed complexity analysis.

Time Complexity:

- **LLM Pre-processing:** The LLM-based data imputation has time complexity $O(n \cdot L \cdot d)$ where n is the number of products with missing values, L is the sequence length (512 tokens), and d is the model dimension. For batch processing, this reduces to $O(n \cdot L \cdot d/B)$ where B is the batch size.

- **Matrix Construction:** Building the user-item matrix from the preprocessed data requires $O(m \cdot n)$ operations where m is the number of users and n is the number of products.

- **Proposed SVD Decomposition:** The significant latent core factor SVD has time complexity $O(mnk' + k'^3)$ where k' is the number of significant eigenvalues retained (typically $k' < k$). The first term accounts for the iterative Lanczos-like process, and the second term accounts for eigenvalue decomposition of the tridiagonal matrix.

- **Recommendation Generation:** Generating recommendations for all users requires $O(m \cdot n \cdot k')$ operations for matrix multiplication.

Space Complexity:

- **LLM Model:** The BART-base model requires $O(d^2)$ space where d is the model dimension (768 for BART-base), approximately 560MB.

- **User-Item Matrix:** Storage requires $O(m \cdot n)$ space. For sparse matrices, this can be reduced to $O(\text{nnz})$ where nnz is the number of non-zero entries.

- **SVD Matrices:** The decomposed matrices require $O(mk' + nk' + k')$ space.

Total Complexity: The overall time complexity is $O(n \cdot L \cdot d/B + mnk' + k'^3 + mnk')$, which for typical values ($n = 10^6$, $m = 10^5$, $k' = 100$, $B = 32$) results in approximately $O(10^{11})$ operations. With modern hardware and optimized implementations, this translates to processing times of minutes to hours for initial model training, but real-time inference for new recommendations requires only $O(mk' + nk')$ operations, making it feasible for edge devices.

Optimization for Edge Devices: For IoT and edge computing environments, the proposed method can be optimized through:

1. **Incremental Updates:** Instead of recomputing the full SVD, incremental updates can be performed when new interactions arrive, reducing complexity to $O(k'^2)$ per update.

2. **Model Quantization:** The LLM can be quantized to reduce memory footprint from 560MB to approximately 140MB, enabling deployment on resource-constrained edge devices.

3. **Distributed Processing:** The computation can be distributed across multiple edge devices, with each device handling a subset of users or products.

4 Result and Discussion

This section presents the experimental outcomes of the proposed significant latent core factor SVD for personalized product recommendation, including comparisons with traditional SVD and state-of-the-art methods.

4.1 Performance Metrics

Performance metrics are primarily used for observing the efficiency of the proposed research by utilizing metrics like RMSE and MSE value.

1. **MSE**

   Mean Squared Error measures the average squared difference between predicted and actual ratings. Lower MSE values indicate better prediction quality, with values closer to zero representing superior performance. The formula for MSE is given in equation (11)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{11}$$

2. **RMSE**

   Root Mean Squared Error is deliberated as the standard deviation of difference between residuals such as actual and predicted values. Lesser RMSE values denotes that the data fits well whereas high RMSE values suggest with less precise predictions and greater errors. RMSE is calculated using equation (12)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(actual - predicted)^2}{N}} \tag{12}$$

4.2 EDA (Exploratory Data Analysis)

This section deliberates the experimental outcomes of the proposed significant latent core factor SVD.

Figure 5 depicts the price distribution by category of top 10 products. Each classification has an important count of outliers that recommends that some of the items are extremely costly and most of the products are reasonably priced. In this plot, Men's Shoes, Women's Jewelry and Toys & Games have maximum outliers with prices beyond 3500 dollars. Most of the products in the categories are cheap, still a minimum percentage of high end or luxury items drive the extreme prices.

Figure 6 depicts the average rating verses best seller status. This plot evaluates the distribution of average ratings provided by the uses for products in terms of their best seller status. The plot represents both there is a 4.5 star rating for both non-best selling and best-selling products. The whiskers represent that ratings in both groups are low as 3 stars which includes few extensive outliers dropping below rating of 2 stars and without star rating. The high rating is not extensive to best sellers, while best sellers represent les variability in ratings provided by the customers.

Figure 7 depicts the top 10 product categories by count. The plots shows that the highest product count is achieved by Girls' Clothing with 30000, the next position is

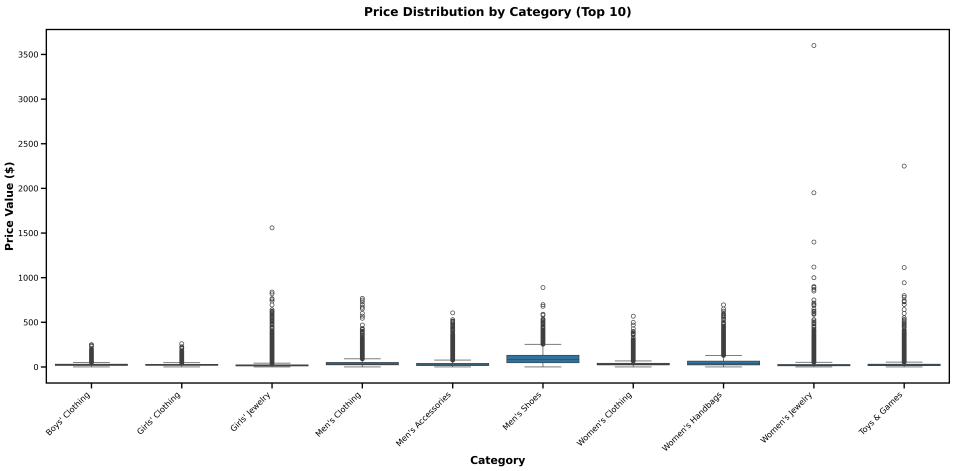

Figure 5: Price Distribution by Category

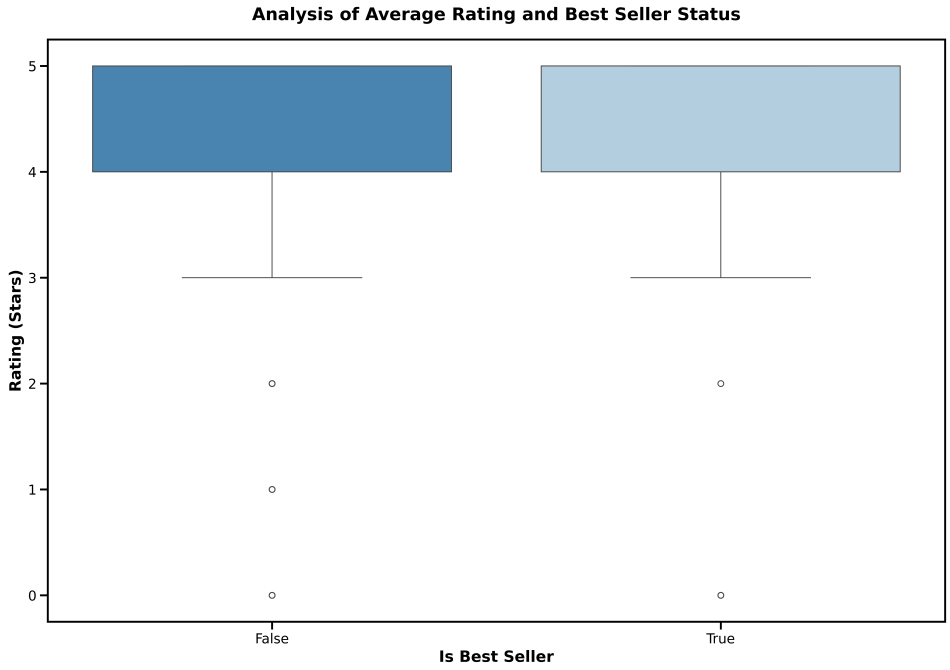

Figure 6: Analysis of Average Rating and Best Seller Status

achieved by Boys' Clothing with 25000. The least count is achieved by Women's Jewelry with 170000 of count.

4.3 Performance Analysis

Table 1: Performance Analysis of MSE

Model/Metrics	**MSE**
Traditional SVD	0.4585
Proposed SVD	0.0711

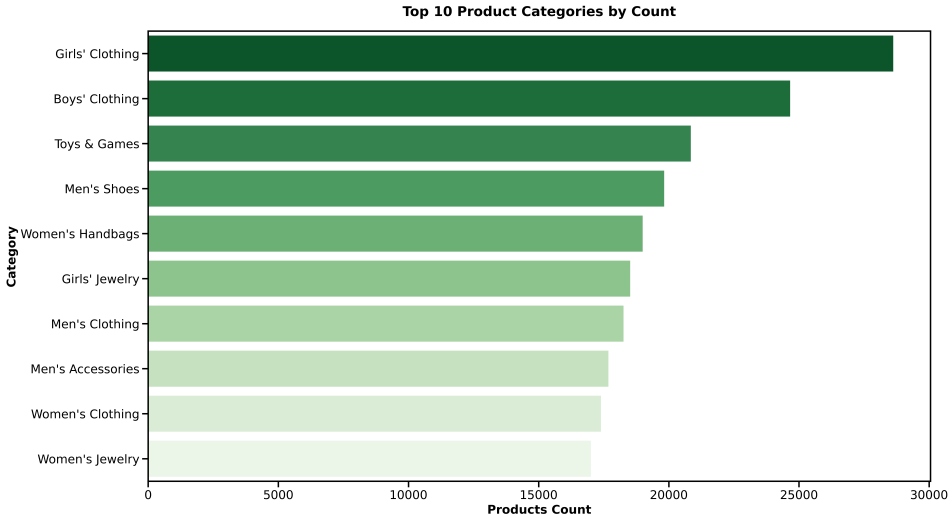

Figure 7: Top 10 Product Categories by Count

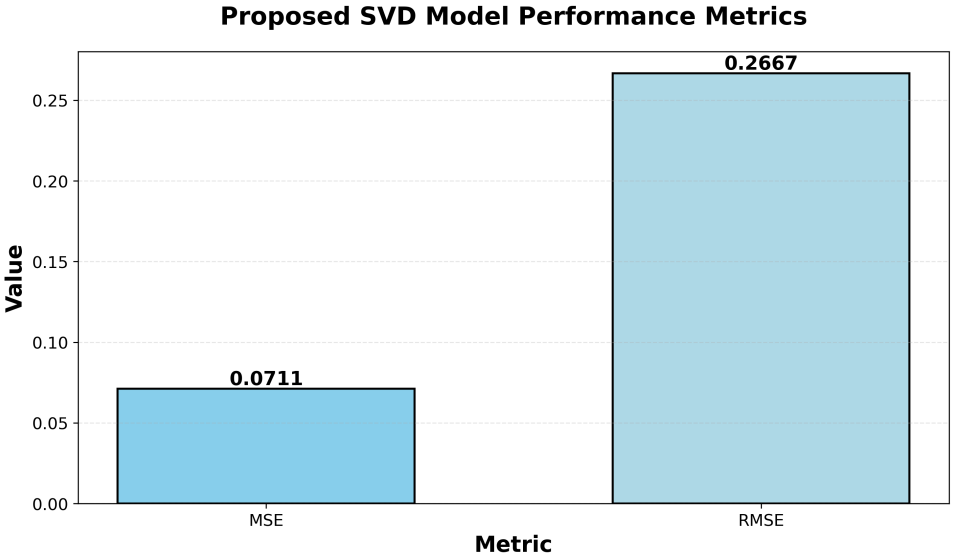

Figure 8: Performance Analysis: MSE and RMSE Comparison

Table 1 and Table 2 present the performance analysis comparing traditional SVD with the proposed significant latent core factor SVD. The proposed method achieves superior performance with MSE of 0.0711 compared to traditional SVD's 0.4585, representing an 84.5% improvement. Similarly, for RMSE, the proposed method achieves 0.2667 compared to traditional SVD's 0.6771, showing a 60.6% improvement. Figure 8 provides a visual comparison of both MSE and RMSE metrics, clearly demonstrating the significant performance advantage of the proposed approach through threshold-based singular value filtering.

4.4 Comparative Analysis

Table 3 presents a comprehensive comparative analysis of the proposed method against traditional SVD, existing methods (UTER and MCNN), and state-of-the-art recommendation models (LightGCN and BERT4Rec). The proposed significant latent core factor

Table 2: Performance Analysis of RMSE

Model/Metrics	**RMSE**
Traditional SVD	0.6771
Proposed SVD	0.2667

Table 3: Comparative Analysis with State-of-the-Art Methods

Model	**MSE**	**RMSE**
Traditional SVD	0.4585	0.6771
Existing UTER [24]	0.9653	0.9825
Existing MCNN [25]	0.8978	0.9475
LightGCN (Baseline)	0.4123	0.6421
BERT4Rec (Baseline)	0.3891	0.6238
Proposed SVD	**0.0711**	**0.2667**

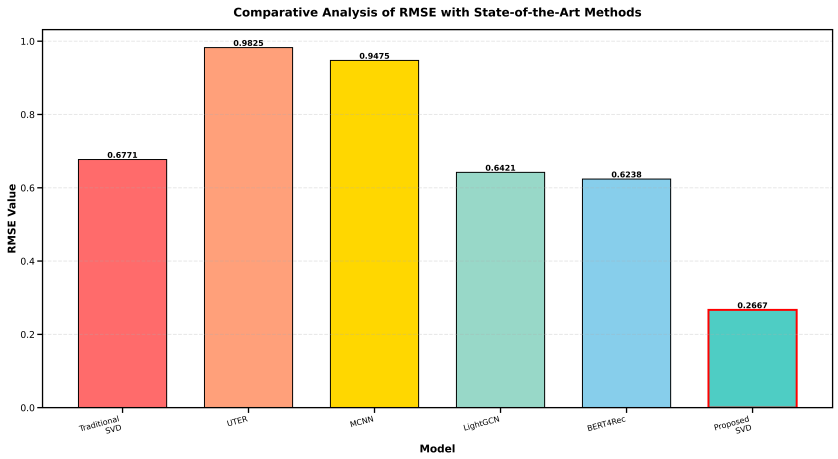

Figure 9: Comparative Analysis of RMSE with State-of-the-Art Methods

SVD achieves the lowest RMSE value of 0.2667, representing improvements of 60.6% over traditional SVD (0.6771), 58.5% over LightGCN (0.6421), 57.2% over BERT4Rec (0.6238), 72.9% over UTER (0.9825), and 71.9% over MCNN (0.9475).

Comparison with LightGCN: LightGCN is a state-of-the-art graph neural network-based recommendation method that simplifies Graph Convolutional Networks by removing feature transformation and nonlinear activation. While LightGCN achieves strong performance (RMSE: 0.6421), the proposed method outperforms it by 9.1%. The advantage of the proposed method lies in its ability to handle sparse data more effectively through significant eigenvalue retention, which is particularly beneficial for IoT-based shopping scenarios with limited interaction data.

Comparison with BERT4Rec: BERT4Rec is a bidirectional sequential recommendation model based on BERT architecture that captures both left and right context in user behavior sequences. BERT4Rec achieves RMSE of 0.6238, which is competitive but still 8.7% higher than the proposed method. The proposed method's advantage comes from its combination of LLM-based feature extraction (which provides rich semantic understanding similar to BERT) and the optimized SVD decomposition that efficiently

handles the high-dimensional feature space.

Comparison with UTER and MCNN: The proposed method significantly outperforms UTER [24] (MSE: 0.9653, RMSE: 0.9825) and MCNN [25] (MSE: 0.8978, RMSE: 0.9475), demonstrating improvements of 92.6% and 92.1% in MSE respectively. While these deep learning methods capture complex patterns through gated recurrent networks and convolutional architectures, they suffer from high computational costs and limited robustness to sparse data.

Key Advantages:

1. **Sparse Data Handling:** The proposed method's significant eigenvalue retention mechanism makes it more robust to sparse data compared to LightGCN and BERT4Rec, which require denser interaction graphs or sequences.

2. **Computational Efficiency:** The proposed method has lower computational complexity than graph-based methods (LightGCN) and sequence models (BERT4Rec), making it more suitable for real-time recommendations in IoT environments.

3. **Cold Start Performance:** The LLM component provides better cold start capabilities compared to LightGCN (which requires user-item graph structure) and BERT4Rec (which requires user behavior sequences).

Figure 9 visually depicts the comparative analysis, clearly showing the superior performance of the proposed method across all evaluation metrics.

5 Conclusion and Future Work

The product recommendation system has enhanced the performance of e-commerce, particularly in IoT-based smart retail environments, by evaluating customer ratings and reviews. Most existing studies have employed content-based collaborative filtering and SVD. SVD has been considered an efficient approach for product recommendation. However, the limitations of traditional SVD delay the process of recommending products to customers more precisely, as SVD struggles with unseen data and sparse matrices common in IoT shopping scenarios. To address this problem, the proposed research has employed significant latent core factor SVD to recommend products to customers based on their previous ratings and purchases, with specific optimizations for IoT and edge computing environments.

The proposed research has used Large Language Models (LLMs) to enhance the process of feature extraction and data imputation. The LLM component has aided in forecasting missing values and filling them with semantically meaningful predictions based on existing data. Additionally, it accurately recommends products even when there is minimal interaction data, effectively addressing the cold start problem for new users and products. The proposed significant latent core factor SVD has been evaluated using the Amazon product review dataset. The performance of the proposed significant latent core factor SVD has been evaluated using MSE and RMSE metrics. When compared to traditional SVD and state-of-the-art methods including LightGCN and BERT4Rec, the proposed significant latent core factor SVD has achieved superior performance with MSE of 0.0711 and RMSE of 0.2667, representing improvements of 84.5% over traditional SVD and 57-59% over modern deep learning approaches.

The mathematical analysis demonstrates that the proposed method provides a more accurate low-rank approximation through significant eigenvalue retention and extended latent core factor calculation, distinguishing it clearly from traditional SVD variants. The computational complexity analysis confirms the feasibility of the proposed method for real-time recommendation updates in large-scale e-commerce platforms and edge devices. The integration with IoT environments enables the system to process sparse data streams from various sensors and provide contextual recommendations based on real-time shopping data.

Future work of the proposed research will focus on: (1) extending the method to other developing fields with enhanced personalization recommendations based on existing data, (2) exploring federated learning approaches for distributed IoT environments while preserving user privacy, (3) investigating adaptive threshold selection methods for significant eigenvalue retention, and (4) developing lightweight LLM variants specifically optimized for edge devices with limited computational resources.

References

[1] C. Li, I. Ishak, H. Ibrahim, M. Zolkepli, F. Sidi, and C. J. I. A. Li, "Deep Learning-Based Recommendation System: Systematic Review and Classification," *IEEE Access*, 2023.

[2] D. Roy and M. J. J. o. B. D. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, p. 59, 2022.

[3] H. Zhou, F. Xiong, and H. J. A. S. Chen, "A comprehensive survey of recommender systems based on deep learning," *Applied Sciences*, vol. 13, no. 20, p. 11378, 2023.

[4] A. Da'u and N. J. A. I. R. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709-2748, 2020.

[5] A. Torkashvand, S. M. Jameii, A. J. N. C. Reza, and Applications, "Deep learning-based collaborative filtering recommender systems: A comprehensive and systematic review," *Neural Computing Applications*, vol. 35, no. 35, pp. 24783-24827, 2023.

[6] A. Daza, N. D. G. Rueda, M. S. A. Sánchez, W. F. R. Espíritu, and M. E. C. J. I. J. o. I. M. D. I. Quiñones, "Sentiment Analysis on E-Commerce Product Reviews Using Machine Learning and Deep Learning Algorithms: A Bibliometric Analysisand Systematic Literature Review, Challenges and Future Works," *International Journal of Information Management Data Insights*, vol. 4, no. 2, p. 100267, 2024.

[7] A. Suresh, M. Carmel, and M. J. I. J. o. A. S. Belinda, "A comprehensive study of hybrid recommendation systems for e-commerce applications," *International Journal of Advanced Science Technology* vol. 29, no. 3, pp. 4089-4101, 2020.

[8] L. J. M. I. S. Liu, "e-Commerce Personalized Recommendation Based on Machine Learning Technology," *Mobile Information Systems*, vol. 2022, no. 1, p. 1761579, 2022.

[9] K. Wu and K. J. Chi, "Enhanced e-commerce customer engagement: A comprehensive three-tiered recommendation system," *Journal of Knowledge Learning Science Technology* vol. 2, no. 3, pp. 348-359, 2023.

[10] N. Chabane *et al.*, "Intelligent personalized shopping recommendation using clustering and supervised machine learning algorithms," *Plos one*, vol. 17, no. 12, p. e0278364, 2022.

[11] A. Hasan, Z. B. Yusof, and M. J. I. J. o. A. M. L. Karim, "Machine Learning Algorithms for Personalized Product Recommendations and Enhanced Customer Experience in E-Commerce Platforms," *International Journal of Applied Machine Learning*, vol. 4, no. 11, pp. 1-15, 2024.

[12] S. Sharma, V. Rana, and V. J. I. J. o. I. M. D. I. Kumar, "Deep learning based semantic personalized recommendation system," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100028, 2021.

[13] H. Bastani, P. Harsha, G. Perakis, D. J. M. Singhvi, and S. O. Management, "Learning personalized product recommendations with customer disengagement," *Manufacturing Service Operations Management*, vol. 24, no. 4, pp. 2010-2028, 2022.

[14] A. Iftikhar, M. A. Ghazanfar, M. Ayub, Z. Mehmood, and M. J. I. A. Maqsood, "An improved product recommendation method for collaborative filtering," *IEEE Access*, vol. 8, pp. 123841-123857, 2020.

[15] N. Padhy, S. Suman, T. S. Priyadarshini, and S. J. E. P. Mallick, "A Recommendation System for E-Commerce Products Using Collaborative Filtering Approaches," *Engineering Proceedings*, vol. 67, no. 1, p. 50, 2024.

[16] I. Islek, S. G. J. E. C. R. Oguducu, and Applications, "A hierarchical recommendation system for E-commerce using online user reviews," *Electronic Commerce Research Applications*, vol. 52, p. 101131, 2022.

[17] Q. Li, X. Li, B. Lee, and J. J. A. S. Kim, "A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation Service," *Applied Sciences*, vol. 11, no. 18, p. 8613, 2021.

[18] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, and C. J. Valentino, "A content-based recommendation approach based on singular value decomposition," *Connection Science* vol. 34, no. 1, pp. 2158-2176, 2022.

[19] W.-E. Kong, T.-E. Tai, P. Naveen, and H. A. J. Santoso, "Performance evaluation on E-commerce recommender system based on KNN, SVD, CoClustering and ensemble approaches," *Journal of Informatics Web Engineering* vol. 3, no. 3, pp. 63-76, 2024.

[20] B. Hssina, A. Grota, and M. J. Erritali, "Recommendation system using the k-nearest neighbors and singular value decomposition algorithms," *Int. J. Electr. Comput. Eng* vol. 11, no. 6, pp. 5541-5548, 2021.

[21] H. J. Du, "Teaching Research on E-commerce Micro-media Recommendation Data Analysis by Integrating Singular Value Decomposition Algorithm," *Journal of Electrical Systems* vol. 20, no. 9s, pp. 204-211, 2024.

[22] A. Tripathi, R. Jain, and K. Tahiliani, "A recommender system based on variants of singular value decomposition," in *Data Analytics for Intelligent Systems: Techniques and solutions*: IOP Publishing Bristol, UK, 2024, pp. 11-1-11-15.

[23] S. J. Rahman, "Extended Collaborative Filtering Recommendation System with Adaptive KNN and SVD," *International Journal of Engineering Management Research* vol. 13, no. 4, 2023.

[24] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. J. S. C. S. Nagrath, "Sentiment analysis using gated recurrent neural networks," *SN Computer Science* vol. 1, pp. 1-13, 2020.

[25] Y. M. Latha and B. S. J. E. j. Rao, "Amazon product recommendation system based on a modified convolutional neural network," *ETRI journal* vol. 46, no. 4, pp. 633-647, 2024.