International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-2, March 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.2.4
Article ID IRP-1727, Pages 27-33
www.ijircst.org

Design and Implementation of a Hybrid E-Commerce Recommendation Engine Using Matrix Factorization and Semantic Content Analysis

Aaftab Alam[1],*Yusuf Jamal[2], Tausheer Alam Shah[3], Syed Mohd Ashir Ali[4], and Ubaid Rehman[5]

[1]Assistant Professor, Department of Computer Science & Engineering, Integral University, Lucknow, India
[2, 3, 4, 5] BTech Scholar, Department of Computer Science & Engineering, Integral University, Lucknow, India

*Correspondence should be addressed to Yusuf Jamal; yjamal@student.iul.ac.in

Received: 22 January 2026; Revised: 12 February 2026; Accepted: 28 February 2026

ABSTRACT - Online and mobile application shopping has become common place for consumers. They have access to numerous e-Commerce sites that offer consumers an even larger selection of consumer products than what can be found in traditional offline retail and an entirely different way to shop compared to traditional offline shopping. Consumers can use their personal preferences (size and color) to filter a large selection of items down to the products they want and then compare them with multiple sites before deciding what product to buy. This research models the user to item relationships through a confidence weighted version of Alternating Least Squares (ALS) trained using implicit user feedback. In parallel to this, I will encode the metadata of items using TF-IDF and then reduce these representations using a truncated Singular Value Decomposition (SVD) to obtain a representation of semantic similarity between items. The final ranking of items will be based on the weighted fusion of the collaboration and content scores, where the model training occurs offline and the inference from this model will occur through an in-memory REST service for low latency response time.

KEYWORDS- Cold Start Problem, Collaborative Filtering, Content-Based Filtering, Hybrid Filtering, Matrix Factorization, Recommender Systems, TF-IDFs.

I. INTRODUCTION

A. The Paradox of Choice in Digital Retail

The last ten years of modern society have significantly changed the way retailers do business today. While in the past, physical storefronts were the primary sales vehicle, the past 10 years has moved away from these locations to large digital marketplaces like Amazon, Flipkart and Alibaba, all of which give consumers easy access to potentially hundreds of millions of unique products. While this vast availability is a significant achievement of modern logistics, it has introduced a psychological barrier for the consumer known as "Information Overload."

In his foundational research, Barry Schwartz posits in The Paradox of Choice that while a degree of choice provides liberation, an excessive amount becomes paralyzing for the individual [1]. A user entering an e-Commerce platform without intelligent guidance is often met with a "wall of noise." In the absence of sophisticated filtering mechanisms [2], the cognitive burden required to reach a purchase decision becomes prohibitive, often resulting in high rates of cart abandonment or user churn.

Consequently, Recommendation Systems (RS) have shifted from being auxiliary features to becoming business-critical infrastructure [3]. They function as a "digital shop assistant," architecting a personalized experience for every individual user. Current industry data underscores this importance, indicating that approximately 35% of Amazon's total revenue and 75% of viewer engagement on Netflix are generated through algorithmic suggestions rather than direct search queries [11][12]. To address these challenges, modern research focuses on hybridizing behavioral data with content analysis to create more robust recommendation engines [2], [3].

B. Limitations of Conventional Approaches

Although RS are widely deployed, maintaining both scalability and recommendation accuracy remains challenging [2]. Several limitations are commonly observed in traditional implementations.

1. Scalability of User-Based Filtering

Early recommendation frameworks relied on User-Based Collaborative Filtering (UBCF), which predicts preferences by identifying similar users [2]. While intuitive, this method becomes computationally expensive as the number of users increases. The pairwise similarity computation introduces quadratic time complexity, making real-time inference difficult in large-scale environments [4].

2. Cold-Start Problem

Collaborative Filtering methods depend heavily on historical interaction data [5]. When new users join the platform or new items are introduced, insufficient interaction history prevents the model from generating reliable recommendations. This issue, commonly referred to as the cold-start problem, limits the effectiveness of purely collaborative approaches [3], [6].

3. *Latency Constraints*

In order to keep users engaged and retain their interest,

today's online shopping environments need to respond quickly. However, if complex recommendation algorithms for finding similarities between very large datasets are not optimized well enough will cause the end user to experience a significant amount of latency when receiving recommendations. Ensuring low-latency serving while preserving model accuracy is therefore a critical engineering requirement.

4. Proposed Hybrid Architecture

To address these challenges, this work proposes a hybrid recommendation framework that integrates collaborative and content-based modeling [3].

There are three basic components to the architecture of the system;

- Item Based Collaborate Filtering - where the similarity is based on an item rather than a user, since item relationships tend to stay relatively stable compared to user relationships [4],
- Based Modeling: To mitigate cold-start scenarios, item metadata is encoded using TF–IDF representations and reduced using truncated SVD [7], [8]. This allows recommendations to be generated even when interaction history is limited.
- Efficient Online Serving: Pre-computed latent vectors are stored in memory during inference to enable fast similarity computation and real-time recommendation delivery.

By combining these strategies, the system improves robustness under sparse interaction conditions while maintaining practical deployment efficiency [3].

II. LITERATURE REVIEW

Many e-Commerce companies offer recommendations to help customers find interesting items from a wide selection of products using recommendation engines [2]. By leveraging user interaction data and item metadata, these systems reduce information overload and improve personalization [3]. Three methodologies of recommendation algorithms exist: (1) Collaborative Filtering; (2) Content-Based Filtering; (3) Hybrid Methods. [2], [3] ,

A. Collaborative Filtering

A Collaborative Filtering (CF) is a technique that predicts an individual user's preferences based on how other users with similar preferences interact with those same items [4]. CF techniques are typically divided into user-based and item-based strategies. Item-based collaborative filtering has demonstrated strong scalability in large commercial environments due to the relative stability of item-item relationships [4].

However, CF methods face limitations when interaction data is sparse. In such cases, similarity estimation becomes unreliable, leading to reduced recommendation accuracy [9]. Furthermore, the absence of prior interaction data for new users or newly introduced items gives rise to the cold-start problem, limiting the effectiveness of purely collaborative approaches [3], [6].

B. Matrix Factorization

To improve scalability and predictive performance, matrix factorization techniques were introduced as a latent factor modeling approach. These methods embed users and items into a shared low-dimensional vector space, enabling the capture of hidden preference structures beyond neighborhood-based similarity models [9].

In many real-world systems, explicit ratings are unavailable, and implicit signals such as clicks and purchases must be utilized. Hu et al. proposed a confidence-weighted Alternating Least Squares (ALS) framework to efficiently handle implicit feedback datasets. Due to its scalability and parallelization capability, ALS remains widely adopted in industrial RS [5].

C. Content-Based Recommendation

A Content Based recommendation algorithm leverages item attributes to make recommendations without regard to other users' actions. Textual metadata such as titles, descriptions, and categories can be represented using the TF–IDF weighting scheme [7]. This representation enables similarity computation between items based on semantic overlap. Because TF–IDF vectors are often high-dimensional, truncated Singular Value Decomposition (SVD) is applied to obtain compact semantic representations while preserving dominant information [8]. Content-based models are particularly effective for addressing item cold-start scenarios where interaction history is limited [3].

D. Hybrid Recommendation Systems

Hybrid recommendation systems integrate collaborative and content-based approaches to mitigate the limitations of individual methods. Burke classified several hybridization strategies and demonstrated that combining multiple techniques can improve robustness and recommendation quality [3]. Weighted hybrid models, in which collaborative and content-based similarity scores are combined, are widely used in practice to balance behavioral and semantic signals [2]. Such integration is particularly beneficial in sparse interaction settings and cold-start conditions.

E. Summary

The reviewed literature indicates that collaborative filtering, matrix factorization, and content-based modeling each provide complementary strengths. Matrix factorization enhances scalability and accuracy in collaborative systems [9], while content-based approaches address cold-start limitations [3]. Hybrid recommendation frameworks combine these advantages to deliver more robust and practical solutions for large-scale e-Commerce platforms [2], [3].

III. METHODOLOGY

This section outlines the architectural design and mathematical formulation of the proposed Hybrid Recommendation Engine.

A. Mathematical Formulation

In many large-scale retail environments, explicit rating data is limited or highly sparse. To address this limitation, interaction data is reformulated as an implicit feedback matrix R following the confidence-weighted approach proposed by Hu et al. [5]. The interaction weight r_{ui} for user u and item i is defined as:

$$r_{ui} = \begin{cases} 4.0 & \text{if the event corresponds to a purchase} \\ 1.0 & \text{if the event corresponds to a view} \\ 0 & \text{otherwise} \end{cases}$$

This formulation assigns greater confidence to strong behavioral signals such as purchases, while still incorporating weaker interaction indicators during factor learning [5].

B. Collaborative Filtering (Matrix Factorization)

We employ the confidence-weighted Alternating Least Squares (ALS) algorithm for collaborative filtering with implicit feedback, as introduced by Hu et al. [5]. The objective is to factorize the user–item interaction matrix R into two lower-dimensional latent matrices representing users and items:

- X_u: latent vector for user u
- Y_i: latent vector for item i

The model optimizes the following regularized loss function:

$$L = \sum_{u,i} c_{ui} \, (p_{ui} - X_u^T Y_i)^2 + \lambda(||X_u||^2 + ||Y_i||^2)$$

where p_{ui} is a binary preference indicator defined as 1 if $r_{ui} > 0$ and 0 otherwise, and $c_{ui} = 1 + \alpha r_{ui}$ represents the confidence weight assigned to each interaction [5]. The regularization parameter λ controls over-fitting by penalizing large latent factor magnitudes. Latent factor models have demonstrated improved scalability and predictive performance compared to neighborhood-based similarity methods [9].

C. Content Based Filtering (TF-IDF)

In order to solve the problem of cold-start scenarios, where a user has no or very little interaction data on an item, we use a content-based filtering approach. This type of filtering is based on item attributes, such as title, brand and category.Items are represented using the Vector Space Model, where textual features are transformed into numerical vectors [7].

$$w_{t,d} = \mathrm{tf}_{t,d} \times \log\left(\frac{N}{\mathrm{df}_t}\right)$$

One limitation of using TF-IDF is that it produces a large and sparse matrix containing numerous dimensions. Because of this, we performed a Truncated Singular Value Decomposition (SVD) on the TF-IDF representation to reduce the overall feature space from a high-dimensional matrix to 128 dimensions (smaller feature space). This provides significant improvements in computation while maintaining similar semantic structures [8].

D. Hybrid Fusion Strategy

We compute the final recommendation score as a weighted hybrid of collaborative filtering scores and content-based similarity scores. Based on Burke's [3] approach to computing recommendation scores, the overall recommendation score for each user (u) and item (i) is computed as follows:

For any user(u) and their items(i), an overall recommendation score will be calculated as follows:

- The Collaborative Filtering Similarity Score $\mathrm{sim}_{CF}(u,i)$ multiplied by a Weighting Factor (w).
- The Content Based Similarity $\mathrm{sim}_{Content}(u,i)$ multiplied by (1 - w).

The Collaborative Filtering Score $\mathrm{sim}_{CF}(u,i)$ will come from the Similarity Scores generated by the Collaborative Filter for user(u) and item(i), and the Content Similarity Score will come from the Content Based Recommendation Engine for user(u) and item(i). We used a Weighting Value (w) of 0.7, which emphasizes the use of Collaborative Filtering over Content Based Methods while still maintaining some of the Consistency that is provided by the use of Content Based Methods.

IV. SYSTEM ARCHITECTURE

The recommendation engine is built using an architecture where the offline model training and the online recommendation serving are completely separated from one another. This ensures the scalability, maintainability, and low-latency inference of the recommendation engine. In the below Figure 1 shows the Layered architecture of the proposed hybrid recommendation system. The offline layer performs model training and artifact generation, while the online layer provides real-time inference.

A. High-Level Workflow

At a high-level, the system has two phases of operation. The first is an "offline" phase in which data preprocessing and feature extraction are done through Python and then a model is created using the features from the data. In the second, or "online", phase the model artifacts created in the offline phase are loaded into memory via a Java Spring Boot application, allowing for real-time recommendations to be served back to the user.

B. Offline Data Preparation

The raw data is sourced from the Amazon Electronics subset of the UCSD Amazon Review Dataset. The following raw files are used:

- Electronics_5.json.gz: User reviews and ratings.
- meta_Electronics.json.gz: Product metadata.

A number of Python-based ETL scripts have been created to facilitate the creation of ETL pipelines for JSON-based and CSV-based workflows found in the pre-processing of data. All pre-processing pipelines apply a common filtering technique that allows for only the 10,000 most popular items (based on interaction count) to remain in the dataset. This common filtering technique reduces the amount of sparsity in the dataset, provides consistency between the collaborative and content-based models, and increases the efficiency of training.

The pre-processing stage produces two runtime datasets:

- interactions.csv: User–item interaction data.
- items.csv: Item metadata used for content analysis.

Figure 1: Layered architecture of the proposed hybrid recommendation system

C. Model Training and Artifact Generation

The training module (train_hybrid.py) consumes the prepared CSV files and generates latent representations for both collaborative and content-based models. The following artifacts are produced:

- user_factors.csv: ALS user latent vectors.
- item_factors.csv: ALS item latent vectors.
- user_content.csv: User content profile vectors.
- item_content.csv: Item content vectors.
- mappings.json: Identifier mappings and hybrid weights.

D. Online Serving Layer

The online inference layer is implemented using Java and Spring Boot. All artifacts generated in the offline phase are loaded into memory at the time of application start-up. During execution, when a user performs a search for items, the service calculates and ranks items based on cosine similarity of the search criteria without needing to query an external database; therefore, allowing for users to receive responses in milliseconds.

V. DATA DESCRIPTION AND PRE-PROCESSING

A. Dataset Source

The system is built using the Amazon Electronics dataset from the UCSD Amazon Review Data (version 2). This dataset includes both user interaction data and rich product

metadata, making it suitable for hybrid recommendation modeling.

The interaction dataset (interactions.csv) contains implicit feedback signals derived from user ratings. Each record includes:

- user_id: Unique user identifier.
- item_id: Product identifier (ASIN).
- event_type: Interaction type (purchase or view).
- event_value: Numeric interaction strength.
- timestamp: Time of interaction.

All interactions are treated as implicit feedback. Ratings greater than or equal to four are classified as purchases, while lower ratings are treated as views. The numeric rating value is directly used as the interaction weight during model training.

B. Item Metadata

The item dataset consists of items that were developed so that the attributes of the items (product title, brand, category, tags, description, etc.) can form the representation of that item as text for use in content-based modelling. These attributes would be concatenated to provide a string representation for each Item.

C. Top-K Item Filtering

To improve model density and computational efficiency, only the top 10,000 most popular items—based on interaction count—are retained. Both interaction and metadata datasets are filtered using this criterion to ensure consistency across training and serving stages.

VI. MODELING AND HYBRID SCORING LOGIC

A. Collaborative Filtering via ALS

Collaborative filtering is implemented using matrix factorization trained with the Alternating Least Squares (ALS) algorithm for implicit feedback datasets [5]. A sparse user–item interaction matrix is constructed using confidence-weighted interaction signals derived from implicit behavioral data [5].

The ALS model is trained with a latent dimension of 64, regularization parameter of 0.02, and 20 optimization iterations. Confidence scaling is incorporated to emphasize observed interactions during factor learning, following the implicit feedback formulation introduced by Hu et al. [5].

After training, user and item latent vectors are L2-normalized to stabilize similarity estimation. Collaborative similarity between a user and an item is then computed using cosine similarity, which is equivalent to the dot product between normalized embedding vectors in latent factor models [9].

B. Content-Based Representation

Content-based item representations are generated using TF-IDF vectorization over concatenated item metadata fields [7]. The transformed TF-IDF vectors are normalized and then compressed into a 128-dimensional dense space using SVD [8]. These vectors contain the latent semantic associations between items while remaining effective for in-memory processing of data.

C. User Content Profiles

User content vectors are constructed by averaging the TF-IDF representations of items the user has interacted with, following standard content-based profiling strategies [10]. These averaged vectors are then projected into the same reduced SVD space as item vectors and normalized. This approach allows content-based similarity to be computed between users and items.

D. Hybrid Recommendation Scoring

The final recommendation score is computed as a weighted combination of collaborative and content-based similarity signals, consistent with weighted hybrid recommendation strategies [3]. The hybrid score for user u and item i is defined by the following equation:

$$\text{score}(u,i) = w \cdot \cos(x_u, y_i) + (1 - w) \cdot \cos(c_u, c_i)$$

where x_u and y_i represent user and item ALS vectors, respectively. The default weight applied to our framework was set at W=0.7, learned during training and persisted as part of the mapping artifacts.

E. Cold-Start Handling

To mitigate cold-start scenarios, where limited interaction data is available, the system falls back to content-based recommendation strategies [3]. We distinguish between two cold-start cases:

1) **Partial cold start**: A user who was not seen during training, but has had one or two interactions at or around the current time. To create recommendations for this user, we make use of the most recent interaction as a seed and retrieve items from the content-based index.
2) **True cold start**: A completely new user with no interaction history. The absence of an original seed item from which to create the final recommendations [also referred to as custom recommendations] leads us to use the lowest ranked items that have received a high number of user interactions as an alternative base set of recommendations.

VII. EXPERIMENTAL EVALUATION

There are four metrics that were used to measure the quality of the model's recommendations. They are:

- **Precision@10:** How many relevant items were included in the top 10 recommended items.
- **Recall@10:** How many of the top 10 recommended items were relevant items.
- **F1@10:** The harmonic mean of precision @ 10 and recall@10.
- **Coverage** - How many distinct items were found in all users' recommendations?

The four metrics can be used to measure how accurately each recommended item is ranked, as well as how diverse the recommended items were.

A. Overall Performance Comparison

The hybrid model was compared with three baselines are:

- Collaborative Filtering (CF)
- Content Only recommendation
- Popularity Based Recommendation

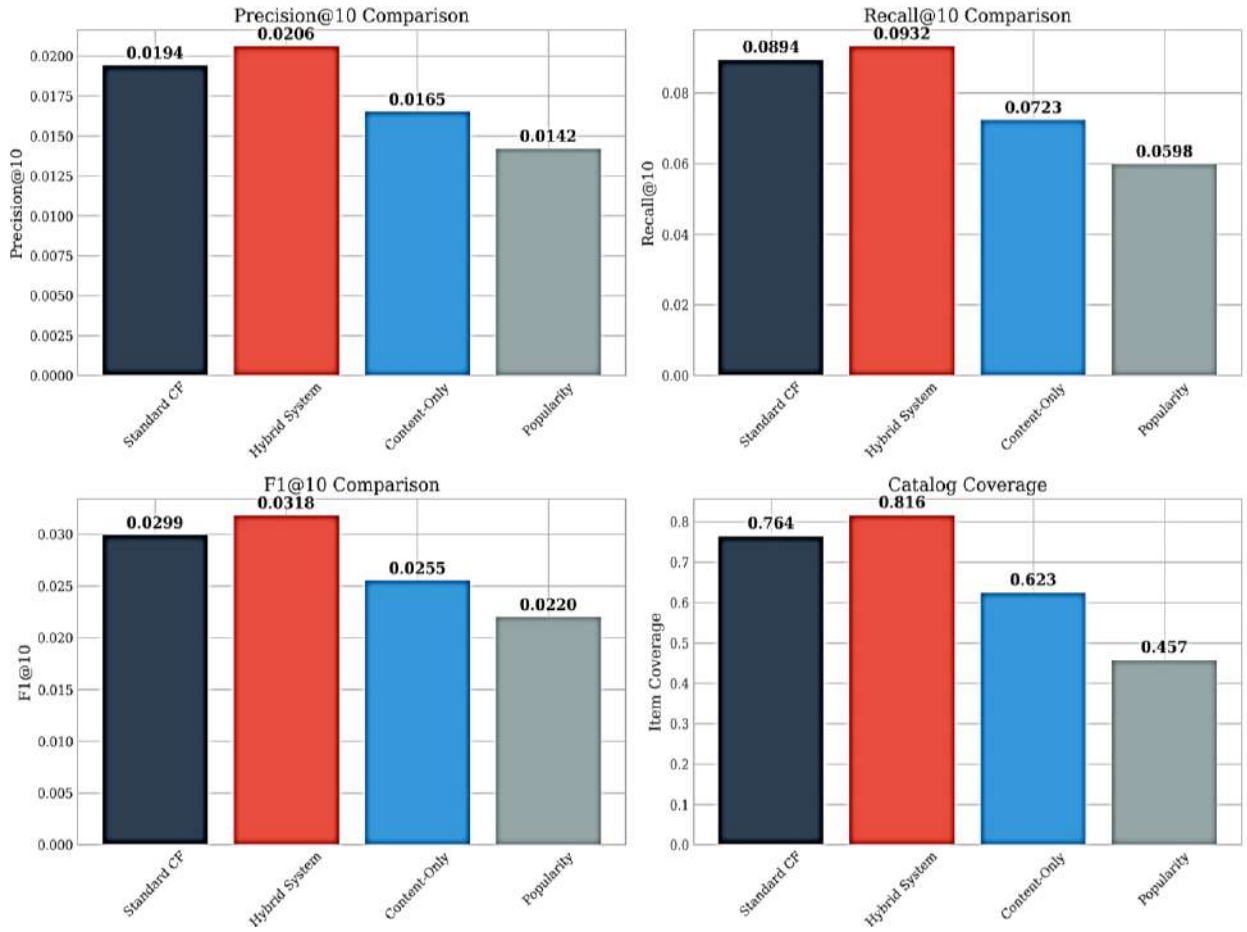

Figure 2: Comparison of Precision@10, Recall@10, F1@10, and Catalog Coverage across different models

In Figure 2, the performance of the hybrid model in the evaluation showed consistent improvement over each of the three baseline models on all evaluation metrics evaluated (Precision@10, Recall@10, F1@10 and Catalog Coverage). The precision metric improved from 1.94% to 2.06% and Recall from 8.94% to 9.32%. F1@10 also shows consistent improvement. In addition, the hybrid approach achieves the highest catalog coverage (81.6%), indicating better recommendation diversity compared to individual models.

B. Overall Performance Comparison

To evaluate robustness under sparse interaction conditions, users are grouped based on activity level.

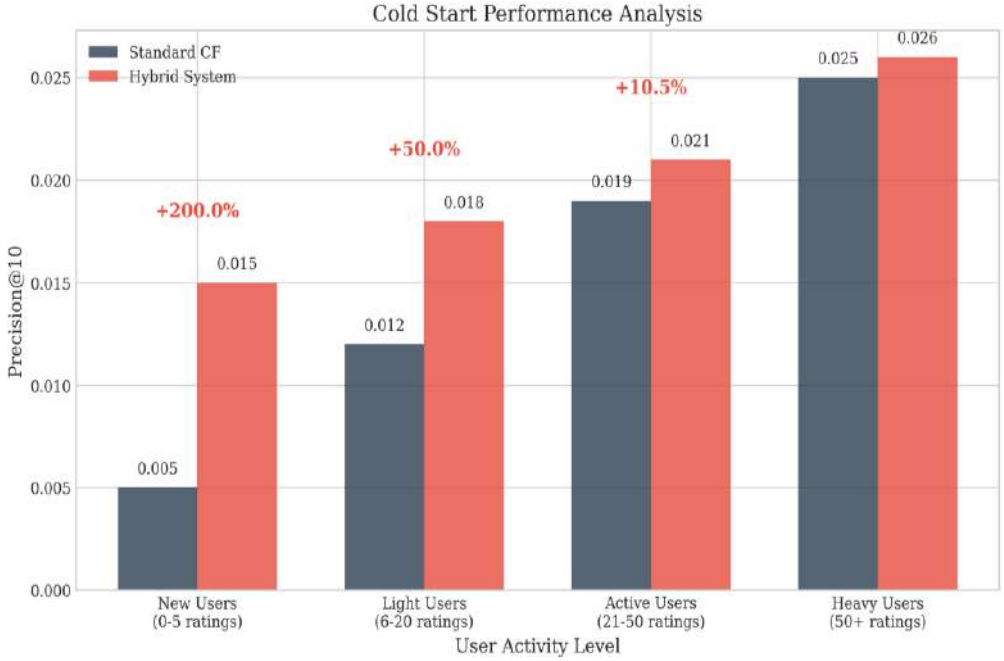

Figure 3: Cold-start performance comparison using Precision@10

The hybrid model performs significantly better than all 3 baseline methods for new users as shown in Figure 3. For users with 0–5 interactions, Precision@10 increases from 0.005 (0.5%) under Standard CF to 0.015 (1.5%) using the hybrid model, representing a threefold improvement. Performance gains are also observed for light and active users, demonstrating improved robustness under sparse interaction conditions.

C. Discussion

The results demonstrate that integrating collaborative filtering with semantic content modeling improves:

- Ranking accuracy
- Diversity of recommendations
- Cold-start robustness

These findings validate the effectiveness of the proposed hybrid framework for large-scale e-Commerce systems.

D. Limitations

Although the proposed system demonstrates effective recommendation performance, it has certain limitations. The model is trained offline and does not update recommendations in real time as new interactions occur. The quality of content-based recommendations depends on both the quantity and quality of available product metadata. The recommendation algorithm does not currently utilise either demographic or contextual information about users.

E. Future Work

Future work on the system could include additions of online or incremental learning to add new recommendations in real-time based on updates to the model. In addition, user demographic characteristics and contextual information could be added to improve personalized recommendations. Lastly, advanced embedding techniques and deep learning-based methods may also be explored as a mechanism for improving both the quality and diversity of recommendations.

VIII. CONCLUSION

The authors present an approach combining collaborative filtering via matrix factorization and a content-based recommendation utilizing semantic content analysis via building a hybrid recommendation engine. The hybrid engine resolves key issues with cold-start customers and sparse interaction in an effective way via an architecture supporting scalable, low-latency architectures using collaborative filtering and content-based recommendation techniques. Additionally, the authors demonstrate through experimental data that their approach can generate real-time recommendations that are relevant thereby making it appropriate for use in e-Commerce applications.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] B. Schwartz, "The Paradox of Choice," in *Positive Psychology in Practice*, pp. 121–138, 2015. Available from: https://doi.org/10.1002/9781118996874.ch8

[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005. Available from: https://doi.org/10.1109/TKDE.2005.99

[3] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002. Available from: https://doi.org/10.1023/A:1021240730564

[4] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003. Available from: https://doi.org/10.1109/MIC.2003.1167344

[5] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008. Available from: https://doi.org/10.1109/ICDM.2008.22

[6] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-Class Collaborative Filtering," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 502–511, 2008. Available from: https://doi.org/10.1109/ICDM.2008.16

[7] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988. Available from: https://doi.org/10.1016/0306-4573(88)90021-0

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990. Available from: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. Available from: https://doi.org/10.1109/MC.2009.263

[10] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 325–341, 2007. Available from: https://doi.org/10.1007/978-3-540-72079-9_10

[11] I. A. Ibrahim and M. Abubakar, "Technological adoption of e-commerce in Nigeria," *International Journal of Innovative Research in Engineering & Management*, vol. 2, no. 6, pp. 1–7, 2015. Available from: https://ijirem.org/view_abstract.php?&primary=QVJULTY5

[12] A. Marchand and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *Journal of Retailing*, vol. 96, 2020. Available from: https://doi.org/10.1016/j.jretai.2020.01.001