IJVAST | Vol. 2, Issue 4 | Aug 2026 | Open Access — Peer Reviewed Research Paper

RS-HybNet: A Hybrid Deep Learning Framework for Smart Product Recommendation in E-Commerce Using Collaborative Filtering, Content-Based Analysis, and Attention Mechanisms

Shruti Tathe[1], Vaishali Kolhe[2]
Department of Computer Science
Vishwalata College of Arts, Commerce and Science, Bhatgaon, Yeola, Maharashtra, India

Abstract—The rapid growth of e-commerce platforms has generated an overwhelming volume of product choices, creating an information overload problem for consumers and a product discovery challenge for retailers. Intelligent recommendation systems address this by personalising product suggestions based on individual user behaviour, preferences, and item characteristics. This paper presents RS-HybNet, a novel hybrid deep learning recommendation framework that integrates three complementary paradigms: collaborative filtering based on neural matrix factorisation of user-item interaction matrices, content-based analysis using TF-IDF vectorisation and convolutional feature extraction from product descriptions and images, and a self-attention mechanism that dynamically weights the contribution of each component based on user context. The framework is trained and evaluated on a combined dataset of 1,891,409 ratings comprising the Amazon Product Reviews subset (1,842,600 ratings, 48,320 users, 12,480 products) and a novel dataset of 48,600 ratings collected from e-commerce interactions of users in the Yeola-Nashik region of Maharashtra, India. RS-HybNet achieves Precision@10 of 0.91, Recall@10 of 0.88, F1-score of 0.895, and RMSE of 0.743 — outperforming standalone collaborative filtering (Precision@10: 0.79), content-based (0.72), deep neural CF (0.83), and matrix factorisation (0.79) baselines. Ablation studies confirm the independent contribution of each component, with the attention mechanism providing a 6.3% precision improvement over the hybrid model without attention. The system is deployed as a REST API integrated with a prototype e-commerce interface and evaluated for real-time recommendation latency, achieving sub-200 ms response time for 95th percentile requests. The collected regional dataset and model weights are publicly released to support future research on recommendation systems for Tier-2 and Tier-3 Indian markets.

Index Terms—Recommendation System, Collaborative Filtering, Content-Based Filtering, Hybrid Recommendation, Neural Matrix Factorisation, Attention Mechanism, Deep Learning, E-Commerce, Product Recommendation, TF-IDF, Cold Start Problem, Precision@K, RMSE, Amazon Reviews, Indian E-Commerce.

I. INTRODUCTION

India's e-commerce market, valued at USD 74 billion in 2022 and projected to reach USD 350 billion by 2030, presents both an enormous commercial opportunity and a significant challenge for product discoverability. With leading platforms offering catalogues of over 100 million products, the average Indian consumer faces an information overload that conventional search and browse interfaces cannot adequately address. Personalised recommendation systems have emerged as the primary mechanism for navigating this complexity — with studies consistently demonstrating that recommendation engines drive 35–40% of Amazon's revenue and 75% of Netflix content consumption [1].

Despite substantial progress in recommendation system research, two critical gaps remain underaddressed in the context of Indian e-commerce. First, the majority of

IV. RS-HYBNET ARCHITECTURE

A. Collaborative Filtering Component

The CF component implements Neural Matrix Factorisation (NeuMF) [8] as the backbone. User IDs and item IDs are embedded into dense latent vectors of dimension d = 64. Two parallel pathways process these embeddings: (1) a Generalised Matrix Factorisation (GMF) pathway that computes the element-wise product of user and item embeddings, capturing linear interaction patterns; and (2) an MLP pathway with four fully-connected layers of dimensions [256, 128, 64, 32], capturing non-linear higher-order interactions. The outputs of both pathways are concatenated and passed through a final sigmoid layer to predict the normalised rating. Dropout with p = 0.3 is applied after each MLP layer to prevent overfitting on the sparse interaction matrix [8].

B. Content-Based Component

published recommendation algorithms are developed and evaluated on Western consumer datasets (Amazon US, MovieLens, Yelp) that may not accurately reflect the purchasing behaviour, product categories, price sensitivity, and seasonal patterns of Indian consumers — particularly those in Tier-2 and Tier-3 cities where the next wave of e-commerce growth is concentrated [2]. Second, most deployed recommendation systems rely predominantly on collaborative filtering, which suffers severely from the cold-start problem for new users and new products — a critical limitation given that Indian e-commerce platforms onboard millions of new users monthly from previously unserved geographies [3].

This paper addresses both gaps through two primary contributions. First, we present RS-HybNet, a hybrid recommendation architecture that combines neural collaborative filtering, content-based feature extraction, and a self-attention weighting mechanism to produce recommendations that are simultaneously accurate for established users and gracefully degraded (rather than completely absent) for cold-start scenarios. Second, we collect, annotate, and publicly release a regional e-commerce dataset of 48,600 product ratings from users in the Yeola-Nashik belt of Maharashtra — a dataset that, to the best of the authors' knowledge, is the first publicly available product recommendation dataset specifically representing Tier-3 Indian consumer behaviour [4].

The remainder of this paper is organised as follows: Section II reviews related work; Section III describes the datasets; Section IV presents the RS-HybNet architecture; Section V details the experimental methodology; Section VI presents and discusses results; Section VII addresses system deployment; and Section VIII concludes with future directions.

II. RELATED WORK

A. Collaborative Filtering

Collaborative filtering (CF) leverages the collective behaviour of users — the intuition that users who have agreed on past items will agree on future items. User-based CF [5] computes user similarity through cosine or Pearson correlation of rating vectors and recommends items liked by similar users. Item-based CF [6] inverts this perspective, finding items similar to those already liked by the target user. Both approaches suffer from the fundamental scalability limitation of storing and computing $O(U^2)$ or $O(I^2)$ similarity matrices for large U users and I items. Matrix factorisation methods [7] address scalability by decomposing the user-item rating matrix into lower-dimensional latent factor

The content component extracts both textual and visual product features. For textual features, product titles and descriptions are encoded using TF-IDF with vocabulary size V = 20,000, followed by a two-layer TextCNN [18] with filter sizes [2, 3, 4] and 128 filters per size, producing a 384-dimensional text feature vector per product. For visual features, product thumbnail images are processed through a ResNet-18 [19] pretrained on ImageNet, with the final average-pooling layer output (512-dimensional) used as the visual feature vector. Text and visual features are concatenated into a 896-dimensional content vector, which is then projected to $d = 64$ dimensions through a linear projection layer to match the CF embedding space. User content preferences are modelled as a weighted average of content vectors of their historically rated items, weighted by normalised rating scores [11].

C. Self-Attention Fusion Mechanism

The key innovation of RS-HybNet is the self-attention based fusion of CF and content signals. Given the CF prediction score s_{CF} and content similarity score s_{CB} for a user-item pair, the model learns context-dependent attention weights α_{CF} and α_{CB} through a two-head self-attention mechanism. The attention inputs are the user embedding u, item embedding v, and a context vector c encoding session features (time of day, device type, browsing category). The attention outputs are normalised weights satisfying $\alpha_{CF} + \alpha_{CB} = 1$. The final recommendation score is computed as: $s_{final} = \alpha_{CF} \times s_{CF} + \alpha_{CB} \times s_{CB}$. This adaptive weighting enables the system to rely more heavily on content signals for cold-start users (where CF signal is weak) and collaborative signals for established users (where CF provides superior personalisation) [20].

D. Training Procedure

RS-HybNet is trained end-to-end using the Adam optimiser with learning rate $\eta = 0.001$ and batch size 512. The loss function is mean squared error (MSE) on normalised ratings, supplemented by a Bayesian Personalised Ranking (BPR) pairwise loss term with weight $\lambda = 0.3$ to encourage correct ranking of relevant over non-relevant items. L2 regularisation with coefficient 1e-5 is applied to all embedding layers. Training runs for a maximum of 50 epochs with early stopping based on validation RMSE with patience of 5 epochs. The content feature extractors (TextCNN and ResNet-18) are pretrained on a product attribute prediction auxiliary task before joint fine-tuning in the full RS-HybNet architecture [21].

V. EXPERIMENTAL METHODOLOGY

A. Evaluation Metrics

System performance is evaluated using both rating prediction and ranking metrics. Rating prediction accuracy is matrices, enabling generalisation to unobserved ratings and scaling to millions of users and items.

B. Deep Learning for Recommendation

He et al. [8] proposed Neural Collaborative Filtering (NCF), replacing the inner product of matrix factorisation with a multi-layer perceptron (MLP) to model non-linear user-item interactions. NeuMF, their combined model, achieved state-of-the-art performance on multiple benchmarks. Wide and Deep Learning [9], developed at Google for app recommendation, jointly trains a wide linear model for memorisation and a deep neural network for generalisation. Attention-based recommendations have gained prominence following the transformer architecture's success in NLP: the BERT4Rec [10] model applies bidirectional self-attention over user interaction sequences, achieving strong performance on sequential recommendation tasks.

C. Content-Based and Hybrid Systems

Content-based filtering analyses item attributes — descriptions, categories, brands, images — to recommend items similar to those previously liked by the user [11]. TF-IDF vectorisation of product descriptions has been widely used for text-based content representation, while convolutional neural networks (CNNs) extract visual features from product images. Hybrid systems combining CF and content-based approaches have consistently outperformed either approach in isolation [12]. The challenge lies in the optimal combination strategy: simple score fusion, stacked generalisation, or learned attention-based weighting. Our RS-HybNet adopts the latter approach, learning to weight CF and content signals per-user and per-context [13].

D. Indian E-Commerce Recommendation

Research specifically addressing Indian e-commerce recommendation characteristics is limited. Sharma et al. [14] investigated price sensitivity as a recommendation signal for Indian fashion e-commerce and found that price-range filtering improved user satisfaction scores by 22%. Gupta and Malik [15] studied regional language product description impact on recommendation quality for Hindi-speaking users. Neither study addressed Tier-3 city consumer behaviour or the cold-start challenge for newly onboarded users — the specific gaps addressed in the present work.

III. DATASETS

A. Amazon Product Reviews Dataset

The Amazon Product Reviews dataset [16], maintained by UCSD, contains product ratings and reviews across 29

measured by Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) between predicted and actual ratings on the held-out test set. Ranking quality for the top-K recommendation list is evaluated using Precision@K, Recall@K, F1-score@K, and Normalised Discounted Cumulative Gain (NDCG@K) with K = 10. Leave-one-out evaluation [8] is used for ranking metric computation — for each user, the most recent rated item is held out as the positive test item, and 99 randomly sampled unrated items are used as negative samples; the model ranks all 100 items and metrics are computed based on the ranking of the held-out positive item [22].

B. Baseline Methods

RS-HybNet is compared against six baselines representing the major recommendation paradigms: (1) User-based Collaborative Filtering with Pearson correlation similarity; (2) Item-based Collaborative Filtering with cosine similarity; (3) Matrix Factorisation via Singular Value Decomposition (SVD) using the Surprise library with latent factors k = 100; (4) Content-Based Filtering using TF-IDF cosine similarity on product descriptions; (5) Deep Neural Collaborative Filtering (NeuMF) without content features; and (6) a Simple Hybrid concatenating CF and content features without attention weighting. All baselines use the same train/validation/test split and hyperparameter tuning procedure as RS-HybNet [23].

C. Implementation Details

RS-HybNet is implemented in Python 3.11 using PyTorch 2.2.0. Training is performed on a single NVIDIA RTX 3090 GPU (24 GB VRAM). The Hugging Face Transformers library provides the pretrained ResNet-18 weights. Hyperparameter optimisation uses Optuna with 50 trials of Bayesian search over embedding dimension (32–256), MLP layer sizes, dropout rate (0.1–0.5), learning rate (1e-4 to 1e-2), and BPR loss weight (0.1–0.5). Reproducibility is ensured by fixing random seeds across PyTorch, NumPy, and Python. The Surprise library (version 1.1.3) is used for SVD and user/item CF baselines [24].

VI. RESULTS AND DISCUSSION

A. Overall Performance Comparison

Table I presents the comparative performance of RS-HybNet against all baselines on the combined test set. RS-HybNet achieves Precision@10 of 0.91, Recall@10 of 0.88, F1-score of 0.895, and RMSE of 0.743 — the best performance across all metrics. The 12.3% relative improvement in Precision@10 over the best single-paradigm baseline (Deep Neural CF, 0.83) and 14.9% improvement over matrix factorisation (0.79) demonstrate the value of the hybrid approach. The content-based component is particularly categories. For this study, we use a subset spanning Electronics, Clothing and Apparel, Home and Kitchen, and Books categories — the four highest-volume categories on Indian e-commerce platforms. After deduplication, removal of ratings with missing product metadata, and filtering users with fewer than 5 ratings, the resulting subset contains 1,842,600 ratings from 48,320 users across 12,480 products. The dataset exhibits a sparsity of 99.69%, typical of large-scale recommendation datasets.

B. Regional Yeola-Nashik Dataset (Novel Contribution)

To address the absence of Tier-3 Indian consumer data in the research literature, we collected a regional e-commerce interaction dataset over a 6-month period (October 2024 – March 2025) in collaboration with three local e-commerce resellers operating in Yeola, Manmad, and Nandgaon talukas of Nashik District. The dataset comprises 48,600 explicit product ratings (1–5 stars) from 3,240 registered users across 860 products spanning five categories: agricultural equipment, textiles, household goods, mobile accessories, and FMCG. Product descriptions were available in Marathi and Hindi in addition to English, providing a multilingual dimension absent from standard benchmark datasets [4]. Key statistics for both datasets are presented in Table II.

C. Data Preprocessing

Product text descriptions were preprocessed through tokenisation, stop-word removal (English, Hindi, and Marathi stop-word lists), and stemming using the Snowball stemmer for English and a custom rule-based stemmer for Marathi. Product images were resized to 224×224 pixels and normalised using ImageNet statistics. User interaction sequences were truncated to the most recent 50 interactions. Ratings were normalised to zero mean per user to remove individual rating scale bias. The combined dataset was split into 80/10/10 train/validation/test partitions using stratified temporal splitting — earlier interactions for training, later interactions for validation and test — to simulate realistic deployment conditions [17].

IV. RS-HYBNET ARCHITECTURE

A. Collaborative Filtering Component

The CF component implements Neural Matrix Factorisation (NeuMF) [8] as the backbone. User IDs and item IDs are embedded into dense latent vectors of dimension $d = 64$. Two parallel pathways process these embeddings: (1) a Generalised Matrix Factorisation (GMF) pathway that computes the element-wise product of user and item embeddings, capturing linear interaction patterns; and (2) an MLP pathway with four fully-connected layers of dimensions [256, 128, 64, 32], capturing non-linear higher-

important for cold-start scenarios: for users with fewer than 5 training interactions, RS-HybNet achieves Precision@10 of 0.74 compared to 0.31 for collaborative filtering baselines — a 2.4× improvement attributable to content-based fallback and attention-weighted fusion [25].

B. Ablation Study

Table III presents the ablation study results, isolating the contribution of each RS-HybNet component. The full model outperforms all ablated variants. Removing the attention mechanism (CF + Content, no Attention) reduces Precision@10 from 0.91 to 0.85 — a 6.6% degradation — confirming that adaptive context-dependent weighting is critical, not merely the combination of signals. The CF component contributes more than the content component in the overall system (CF alone: 0.79 vs Content alone: 0.72), consistent with the general literature finding that collaborative signals are stronger predictors for established users. However, for cold-start users the relative importance reverses — content features provide the primary recommendation signal [26].

C. Regional Dataset Analysis

Evaluation on the Yeola-Nashik regional dataset reveals important differences from the Amazon benchmark. The regional dataset exhibits a stronger price sensitivity pattern — items priced within 15% of the user's median purchase price are rated 0.6 stars higher on average, and recommendation accuracy improves significantly when price-range filtering is applied as a pre-filter. Agricultural equipment recommendations benefit substantially from seasonal context features (pre-kharif, post-rabi) — adding session-month encoding to the context vector c improves Precision@10 for this category from 0.84 to 0.91. Multilingual product description processing (Marathi and Hindi tokenisation) provides a 4.2% precision improvement over English-only text processing for the regional dataset [4].

D. System Latency Analysis

Recommendation latency was measured across 10,000 API calls against the deployed REST API endpoint. Median latency is 84 ms, 95th percentile latency is 187 ms, and 99th percentile latency is 312 ms — comfortably below the 500 ms user-perceptible delay threshold for interactive product browsing. CF embedding lookup and attention computation account for 61% of latency; content feature extraction accounts for 32%; and network/serialisation overhead accounts for the remaining 7%. Batch recommendation pre-computation for top-500 active users reduces API latency by 68% for that user segment, suggesting that a tiered caching strategy would further improve production latency [27].

VII. SYSTEM DEPLOYMENT

order interactions. The outputs of both pathways are concatenated and passed through a final sigmoid layer to predict the normalised rating. Dropout with $p = 0.3$ is applied after each MLP layer to prevent overfitting on the sparse interaction matrix [8].

B. Content-Based Component

The content component extracts both textual and visual product features. For textual features, product titles and descriptions are encoded using TF-IDF with vocabulary size $V = 20{,}000$, followed by a two-layer TextCNN [18] with filter sizes [2, 3, 4] and 128 filters per size, producing a 384-dimensional text feature vector per product. For visual features, product thumbnail images are processed through a ResNet-18 [19] pretrained on ImageNet, with the final average-pooling layer output (512-dimensional) used as the visual feature vector. Text and visual features are concatenated into a 896-dimensional content vector, which is then projected to $d = 64$ dimensions through a linear projection layer to match the CF embedding space. User content preferences are modelled as a weighted average of content vectors of their historically rated items, weighted by normalised rating scores [11].

C. Self-Attention Fusion Mechanism

The key innovation of RS-HybNet is the self-attention based fusion of CF and content signals. Given the CF prediction score s_{CF} and content similarity score s_{CB} for a user-item pair, the model learns context-dependent attention weights α_{CF} and α_{CB} through a two-head self-attention mechanism. The attention inputs are the user embedding u, item embedding v, and a context vector c encoding session features (time of day, device type, browsing category). The attention outputs are normalised weights satisfying $\alpha_{CF} + \alpha_{CB} = 1$. The final recommendation score is computed as: $s_{final} = \alpha_{CF} \times s_{CF} + \alpha_{CB} \times s_{CB}$. This adaptive weighting enables the system to rely more heavily on content signals for cold-start users (where CF signal is weak) and collaborative signals for established users (where CF provides superior personalisation) [20].

D. Training Procedure

RS-HybNet is trained end-to-end using the Adam optimiser with learning rate $\eta = 0.001$ and batch size 512. The loss function is mean squared error (MSE) on normalised ratings, supplemented by a Bayesian Personalised Ranking (BPR) pairwise loss term with weight $\lambda = 0.3$ to encourage correct ranking of relevant over non-relevant items. L2 regularisation with coefficient 1e-5 is applied to all embedding layers. Training runs for a maximum of 50 epochs with early stopping based on validation RMSE with patience of 5 epochs. The content feature extractors

RS-HybNet is deployed as a containerised REST API service using FastAPI (Python) with Docker containerisation. The model serving stack employs ONNX Runtime for inference optimisation, achieving 3.2× speedup over native PyTorch inference at batch size 1. The API exposes three endpoints: /recommend/{user_id}?k={k} for personalised top-K recommendations, /similar/{item_id}?k={k} for item-to-item recommendations, and /trending?category={category} for popularity-based fallback. A Redis cache stores pre-computed recommendations for active users with a 1-hour TTL. The system is deployed on a 4-vCPU, 16 GB RAM virtual machine and handles 450 concurrent recommendation requests — sufficient for the prototype regional e-commerce platform. Model retraining is scheduled weekly using accumulated new interaction data through an automated MLflow pipeline [28].

VIII. CONCLUSIONS AND FUTURE WORK

This paper has presented RS-HybNet, a hybrid deep learning recommendation framework integrating neural collaborative filtering, content-based analysis, and self-attention fusion. The system achieves state-of-the-art Precision@10 of 0.91 and RMSE of 0.743 on a combined benchmark and regional Indian e-commerce dataset, outperforming all single-paradigm baselines and a non-attention hybrid by significant margins. The attention mechanism's 6.6% precision contribution and the content component's 2.4× cold-start improvement together demonstrate the value of the hybrid, attention-weighted architecture. The novel Yeola-Nashik regional dataset provides a reusable benchmark for future Tier-3 Indian e-commerce recommendation research.

Future work will investigate: (1) large language model (LLM)-based product description encoding using multilingual BERT to replace TF-IDF for improved Marathi and Hindi text understanding; (2) graph neural network (GNN) based user-item interaction modelling to capture higher-order social influence patterns observed in the regional dataset; (3) federated learning deployment to enable personalisation without centralising sensitive user interaction data, addressing DPDP Act 2023 compliance requirements; (4) reinforcement learning from human feedback (RLHF) to optimise long-term user engagement beyond immediate click-through metrics; and (5) expansion of the regional dataset to all six districts of Nashik Division.

(TextCNN and ResNet-18) are pretrained on a product attribute prediction auxiliary task before joint fine-tuning in the full RS-HybNet architecture [21].

TABLE I — Performance Comparison: RS-HybNet vs. Baseline Recommendation Algorithms on Combined Test Set

Algorithm	Precision @10	Recall @10	F1 Score	RMSE	Training Time (s)	Cold Start
User-based CF	0.71	0.68	0.695	0.982	42.3	Poor
Item-based CF	0.74	0.71	0.724	0.941	38.6	Poor
Matrix Factorization (SVD)	0.79	0.76	0.774	0.876	124.7	Poor
Content-Based (TF-IDF)	0.72	0.69	0.704	0.921	18.4	Good
Deep Neural CF	0.83	0.79	0.809	0.812	386.2	Fair
Proposed Hybrid (RS-HybNet)	0.91	0.88	0.895	0.743	198.4	Good

TABLE II — Dataset Statistics: Amazon Product Reviews Subset and Yeola-Nashik Regional Dataset

Dataset	Users	Products	Ratings	Sparsity (%)	Rating Scale	Domain
Amazon Product Reviews (subset)	48,320	12,480	1,842,600	99.69	1–5	E-commerce
MovieLens 1M	6,040	3,706	1,000,209	95.53	1–5	Entertainment
Collected (Yeola Region)	3,240	860	48,600	98.26	1–5	Local E-comm
Training Split	80%	—	1,553,127	—	—	—
Validation Split	10%	—	194,141	—	—	—
Test Split	10%	—	194,141	—	—	—

TABLE III — Ablation Study: Contribution of Individual RS-HybNet Components (Precision@10 and RMSE)

Model Variant	CF Component	Content Component	Attention Mechanism	Precision @10	RMSE
CF Only	✓	✗	✗	0.79	0.876
Content Only	✗	✓	✗	0.72	0.921
CF + Content (no Attention)	✓	✓	✗	0.85	0.812
CF + Attention (no Content)	✓	✗	✓	0.87	0.798
Full RS-HybNet	✓	✓	✓	0.91	0.743

IX. REFERENCES

[1] Resnick P. et al., GroupLens: An open architecture for collaborative filtering, Proc. ACM CSCW, pp. 175–186, 1994.

[2] IBEF, E-Commerce Industry in India — Market Size and Growth, India Brand Equity Foundation Report, New Delhi, 2024.

[3] Lam X.N. et al., Addressing cold-start problem in recommendation systems, Proc. ACM AIMSA, pp. 208–219, 2008.

[4] Tathe S. and Kolhe V., Yeola-Nashik Regional E-Commerce Dataset v1.0, Zenodo Repository, DOI: 10.5281/zenodo.2026001, 2025.

[5] Herlocker J.L. et al., Algorithmic framework for performing collaborative filtering, Proc. ACM SIGIR, pp. 230–237, 1999.

[6] Sarwar B. et al., Item-based collaborative filtering recommendation algorithms, Proc. ACM WWW, pp. 285–295, 2001.

[7] Koren Y., Bell R. and Volinsky C., Matrix factorization techniques for recommender systems, IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.

[8] He X. et al., Neural collaborative filtering, Proc. ACM WWW, pp. 173–182, 2017.

[9] Cheng H.T. et al., Wide and deep learning for recommender systems, Proc. ACM RecSys Workshop, pp. 7–10, 2016.

[10] Sun F. et al., BERT4Rec: Sequential recommendation with bidirectional encoder representations, Proc. ACM CIKM, pp. 1441–1450, 2019.

[11] Lops P., de Gemmis M. and Semeraro G., Content-based recommender systems: State of the art and trends, in Recommender Systems Handbook, Springer, pp. 73–105, 2011.

[12] Burke R., Hybrid recommender systems: Survey and experiments, User Modeling and User-Adapted Interaction, vol. 12, pp. 331–370, 2002.

[13] Vaswani A. et al., Attention is all you need, Proc. NeurIPS, pp. 5998–6008, 2017.

[14] Sharma A. et al., Price-aware recommendation for Indian fashion e-commerce, Proc. ACM RecSys, pp. 412–416, 2023.

[15] Gupta R. and Malik S., Multilingual product recommendation for Hindi-speaking Indian consumers, J. Inf. Sci., vol. 48, pp. 822–835, 2022.

[16] He R. and McAuley J., Ups and downs: Modelling the visual evolution of fashion trends with one-class collaborative filtering, Proc. ACM WWW, pp. 507–517, 2016.

[17] Ji S. et al., A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Trans. Neural Networks Learn. Syst., vol. 33, no. 2, pp. 494–514, 2022.

[18] Kim Y., Convolutional neural networks for sentence classification, Proc. EMNLP, pp. 1746–1751, 2014.

[19] He K. et al., Deep residual learning for image recognition, Proc. IEEE CVPR, pp. 770–778, 2016.

[20] Chen J. et al., Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention, Proc. ACM SIGIR, pp. 335–344, 2017.

[21] Paszke A. et al., PyTorch: An imperative style, high-performance deep learning library, Proc. NeurIPS, pp. 8024–8035, 2019.

[22] He X. et al., Fast matrix factorization for online recommendation with implicit feedback, Proc. ACM SIGIR, pp. 549–558, 2016.

[23] Hug N., Surprise: A Python library for recommender systems, Journal of Open Source Software, vol. 5, no. 52, p. 2174, 2020.

[24] Akiba T. et al., Optuna: A next-generation hyperparameter optimization framework, Proc. ACM KDD, pp. 2623–2631, 2019.

[25] Zhang S. et al., Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys, vol. 52, no. 1, pp. 1–38, 2019.

[26] Adomavicius G. and Tuzhilin A., Toward the next generation of recommender systems: A survey, IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2005.

[27] Liu Q. et al., Real-time attention based look-alike model for recommender system, Proc. ACM KDD, pp. 2765–2773, 2019.

[28] Zaharia M. et al., MLflow: A system for machine learning lifecycle, Proc. VLDB, pp. 1–4, 2018.

[29] Tathe S. and Kolhe V., Attention-weighted hybrid recommendation for regional Indian e-commerce, Proc. National Conference on AI and Data Science, SPPU Pune, February 2025.

[30] Ricci F. et al., Recommender Systems Handbook, 2nd ed., Springer, New York, 2015.