Enhancing Personalized E-Commerce Recommendations through a Hybrid Recommendation Framework

Jia Chen Goh1, a) and Sook Ling Lew1, 2, b)

*1Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Bukit Beruang, Melaka, Malaysia*

*2Centre for Advanced Analytics, CoE for Artificial Intelligence Multimedia University, Jalan Ayer Keroh Lama, 75450, Bukit Beruang, Melaka, Malaysia*

*b) Corresponding author:* [*sllew@mmu.edu.my*](mailto:sllew@mmu.edu.my)

*a) jc\_goh\_2002@hotmail.com*

**Abstract.** Trends in e-commerce are constantly shifting and so are the habits of online shoppers. In this fast-changing environment, recommendation systems play a key role in providing personalized experiences and helping businesses stay competitive. However, these systems are not without their challenges. For example, they struggle with things like the cold start problem (when the system cannot make good recommendations for new users or products due to a lack of data), untrusted or manipulated data, and a general lack of personalization. When recommendations are too generic, they miss the mark on what specific users really want. This study aims to solve these issues by proposing a hybrid recommendation system. By combining collaborative filtering, content-based filtering and popularity-based filtering, the system can make better and more personalized suggestions. The idea is that each method brings something valuable to the table, so putting them together can solve problems like the cold start issue and the need for more trustworthy data. The experimental results show the new recommendation system significantly boosts accuracy and user satisfaction. It outperformed previous hybrid models in several key areas, including recall (with improvements of up to +0.024 at Recall@15), precision and F1-score. This means the system provides more frequent recommendations and keeps users happier with more relevant and reliable suggestions. However, this study requires further improvement in its algorithms to enhance their precision and scalability. Future research will focus on working with larger datasets and making the system more adaptable across different areas.

# Introduction

E-commerce refers to all kinds of buying and selling of goods and services, whether it is online shopping or secure transaction over the internet. It includes a variety of business models, those like B2C and B2B, that have changed the way of conventional business. The convenience of online shopping is replacing the need to visit stores physically. It allows transactions to be done quickly and securely [1]. As e-commerce has flourished, competition has sharpened in the digital marketplace. Organizations are steadily adopting systems that attract consumers and increase sales. These systems help understand the needs and wants of customers that are not the same from person to person and keep changing over time to bring about betterment in the consumer experience. According to recent reports, user engagement and conversion rates in E-commerce platforms are increasing by up to 30% due to the use of AI [2] Yet, past recommend systems never take user individual preferences into account. As a result, the suggestions are too general. A hybrid system that uses collaborative filtering, content-based filtering, and popularity-based filtering is suggested.

# LITERATURE REVIEW

## Existing Recommendations Systems

The current era features numerous existing recommendation systems which operate through different algorithms and frameworks. The chapter examines the historical development of recommendation systems together with their contemporary advancements and their ability to draw customers and boost e-commerce platform sales. Machine learning provides adaptable solutions that include product recommendations. The modern e-commerce platforms utilize four main recommendation system categories which include popularity-based filtering and content-based filtering and collaborative filtering and hybrid filtering [3],[4],[5].

The Popularity-based Filtering system recommends popular products to users without considering their personal behavior patterns. The system depends on general market demand and popularity to suggest items through metrics such as views and likes and shares. The e-commerce platform uses best-selling and most-viewed product suggestions as an example. The method works well for users who want to discover new products or find items matching their current tastes by using overall popularity metrics [3].

Content-based filtering makes recommendations by looking at the features of items themselves. It focuses on analyzing the content of each item. This method works especially well for recommending things like web pages, articles, and news stories. It looks at the kinds of items a user has interacted with or rated positively in the past, and then finds similar ones to suggest. The idea is to align recommendations with the user's personal tastes and previous behavior [2]. Figure 1 shows an example of the process of a content-based recommend system. Collaborative filtering is another common method. It is based on the idea that people who have liked the same things in the past will likely agree again. This method falls into two main categories: (1) model-based and (2) memory-based. Model-based collaborative filtering builds a model to predict user preferences, while memory-based collaborative filtering compares user-item interactions directly to generate suggestions. Memory-based methods are further divided into two types: (1) item-based and (1) user-based collaborative filtering. Item-based collaborative filtering looks at how similar items are to each other, while user-based collaborative filtering focuses on how similar users are. Based on these similarities, the system predicts what a user might like next [7]. Figure 2 illustrates collaborative filtering in a recommender system.

User-based Collaborative Filtering (UBCF) specifically recommends items by finding users with similar preferences. For instance, if Tim and John tend to like the same things, and Tim enjoys sundaes and donuts, the system suggests sundaes and donuts items to John too [7].

This method suggests items to end-users by finding users who have similar tastes. A good example of this is the user-based Nearest Neighbour (KNN) algorithm, which looks for the closest "matches"—other users whose preferences line up among them [7]. It does this by calculating how similar among them. Think of it like finding people who enjoy the same things they do, and then using their picks to guide end-users’ own recommendations. Now, there is also item-based collaborative filtering (IBCF), which works a bit differently. Instead of focusing on users, it looks at the items themselves. So if John likes ice cream, the system recommends a sundae because they are both in the same category of delicious treats [7]. With IBCF, the system does two main things: it finds similar items and then predicts what users might like based on that. One big advantage of IBCF over the user-based method is that it pre-calculates the similarities between items, so it does not need to go searching for similar users every time. It is a lot more efficient [5]. As shown in Figure 3, it illustrates user-based collaborative filtering process in a recommend system. Figure 4 illustrates item-based collaborative filtering process in a recommender system.

|  |  |  |  |
| --- | --- | --- | --- |
| A diagram of a product  Description automatically generated | A diagram of a product  Description automatically generated | Concepts of user-based and item-based filtering | Download Scientific  Diagram | Concepts of user-based and item-based filtering | Download Scientific  Diagram |
| **FIGURE 1.** Content-based filtering | **FIGURE 2.** Collaborative filtering | **FIGURE 3.** User-based collaborative filtering | **FIGURE 4** Item-based collaborative filtering |

### Hybrid Filtering

By integrating multiple techniques, hybrid models create more resilient and dependable recommendation systems, resulting in improved overall performance [6]. Figure 5 illustrates the hybrid filtering framework proposed in this study, which combines different recommendation systems to optimize the system and overcome the limitations of standalone algorithms.

A diagram of a diagram

Description automatically generated

**FIGURE 5.** Hybrid filtering

## Prediction Algorithms

Prediction algorithms are vital in recommendation systems for identifying similar users or items via historical interactions. KNN calculates distances to find similar users/items. Using the Alibaba dataset, Item KNN had the lowest short-term forecast accuracy (11.74%) but the highest accuracy (44.37%), recall (15.73%), and user coverage (94.74%). K-means-enhanced KNN improved movie rating predictions [6]. Slope One predicts using average rating differences. Weighted and bi-polar variants improve it by factoring in rating counts and user sentiment [9]. Accuracy improves further with trusted data integration. RBF Neural Network, enhanced for mobile e-commerce, learns user behavior and location data. Combined with Dempster-Shafer theory, it boosts performance [10]. XGBoost merges classification and prediction using collaborative filtering. It considers neighboring products' rating variances for recommendations [11]. Matrix Factorization (MF) decomposes the user-item matrix, solving sparsity and cold-start issues using review-based side info and deep autoencoders [11]. W-RNN, a time-windowed RNN, surpasses traditional models in short-term prediction accuracy and convergence speed [12]. CNNs, enhanced with Glove and skip-gram, improve sentiment analysis and word representation in product reviews, increasing prediction accuracy [13]. RNNs assess reviews for promotion strategies. Evaluated via precision, recall, accuracy, and F1 score, the proposed model outperforms traditional methods in product prediction [14].

## Similarity Calculation Methods

In similarity calculation method, instead of predicting missing values, it relies on existing data to compute similarity between users, items, or user-item pairs. The following sub-sections justify the adoption of the prediction algorithms in this study. The Pearson Correlation Coefficient (PCC) is a widely used algorithm for measuring the similarity between users or items. PCC computes the similarity between two entities a and b by evaluating their ratings on common items. The formula is given by Equation (1) where Raj​ and Rbj are the ratings of users a and b for item j, and Ra and Rb​ are their average ratings. PCC values range from -1 to +1, indicating the strength and direction of the correlation [8],[15]. CPCC modifies PCC to reduce the impact of positive and negative assessments by using the median rating Rmed in Equation (2) instead of the mean. This approach is beneficial for datasets with uneven rating distributions but is inapplicable to datasets with even rating scales [8],[15]. Jaccard Similarity measures the proportion of common items rated by two users compared to the total number of items they have rated. It is defined as Equation (3). The Extended Jaccard Similarity incorporates the frequency of item purchases to provide a more detailed similarity measure in Equation (4).

(1)

(2)

(3)

(4)

## Extract Key from Items

Identifying item features is crucial in collaborative and content-based filtering. It aligns user preferences with item categories, improving recommendation accuracy. LDA, a probabilistic topic model, treats documents as topic mixtures and topics as word distributions. It is used in NLP to uncover latent topics for text-based recommendations [11]. TF-IDF measures term importance: TF captures frequency in a document, IDF measures rarity across the corpus. The TF-IDF score is calculated as in Equation (5) [15],[16]. TUDPA preprocesses review comments to identify target users, applying tokenization, POS tagging, and parsing with NLP techniques [15]. DMT uses multiple Transformers to model diverse user behaviors. With Multi-gate Mixture-of-Experts, it optimizes for click-through and conversion rates, enhancing accuracy and fairness [17].

(5)

# Methods and Materials

Figure 6 presents the proposed hybrid recommendation system, which integrates three different recommendation methods: collaborative filtering, content-based filtering, and popularity-based filtering. This hybrid approach aims to improve recommendation accuracy by leveraging the strengths of each method.

**A diagram of a computer system

Description automatically generated**

**FIGURE 6.** Proposed framework

## Dataset

Data is sourced from Kaggle’s CI&T DeskDrop dataset [18], covering 12 months (Mar 2016–Feb 2017). It includes ~3,000 articles and over 73,000 user interactions. The dataset consists of two CSV files: “shared\_articles.csv” and “users\_interactions.csv.” To improve accuracy and address the cold-start issue, users with fewer than 5 interactions are excluded. Interaction strengths are assigned as follows: VIEW = 1.0, LIKE = 2.0, BOOKMARK = 2.5, FOLLOW = 3.0, and COMMENT CREATED = 4.0, enhancing relevance. Matrix Factorization is used for collaborative filtering. Content-based filtering applies stop word removal, TF-IDF, and Constrained Pearson for similarity. Popularity-based filtering ranks items by popularity. The hybrid framework combines all three methods, with collaborative filtering weighted highest, followed by content-based, then popularity-based, reflecting user behavior and preferences. Recommendations are based on: items liked by similar users, items similar to past purchases, and top-rated/selling/wishlisted items. The Top-N method selects a fixed number of recommendations per user.

# Analysis and prediction

Recall rates are used as a metric to evaluate the performance of the recommendation model because they measure the proportion of relevant items successfully recommended. Recall increases with the number of recommendations (n), capturing more relevant items. In contrast, accuracy measures overall correctness but is not as meaningful in recommendation systems due to the nature of the data. For instance, with 1,000 items and only 100 relevant ones, a model retrieving 10 items with 5 true positives and 5 false positives would result in a Recall@10 of 0.5, but an accuracy of just 0.005. This low accuracy is misleading as it includes many items irrelevant to the user. Recall, on the other hand, focuses on how well the system identifies relevant items in top recommendations, aligning better with the system's primary goal. Higher Recall@N values indicate higher recall rates because a larger N increases the likelihood of capturing interacted items among the top N ranked items. The proposed method demonstrates the highest recall rate across all Recall@N values, followed by the Hybrid model, Collaborative-Based model, Popularity-Based model, and finally the Content-Based model. The proposed model outperforms the Hybrid model, showing higher recall values at Recall@5, Recall@10, Recall@15, and Recall@20 by 0.018, 0.024, and 0.020 respectively. The proposed method with different weights shows varying performance. The first method uses basic weights to study the impact of popularity as a factor. Lower popularity weights result in higher recall rates and better performance. The optimal weight for Collaborative Filtering is 1000, achieving the highest recall rates. For Content-Based Filtering, a medium weight between 5 to 100 yields the best performance. The best weights for the proposed method are Content-Based at 50, Collaborative Filtering at 1000, and Popularity at 0.1.

## Different Metrics of All Models

This section evaluates models using accuracy, precision, and F1-score. These metrics highlight that recall rate is the most informative for assessing model effectiveness. Accuracy measures overall correctness in predictions. As N increases, accuracy improves if relevant items are included. The proposed method has the highest accuracy but not a high percentage, at only 4%. This does not imply inaccuracy; rather, it reflects the challenge of recommending a small subset of relevant items from a large dataset. Precision measures the proportion of relevant recommended items. As N increases, precision tends to decrease because more non-relevant items are included. The proposed model has the highest precision, followed by the Hybrid, Collaborative, Popularity-Based, and Content-Based models. F1-score, the harmonic mean of precision and recall, shows the same ranking as the other metrics. The proposed model has the highest F1-score, indicating peak performance.

## Testing and Prediction

The testing begins with inspecting user interests based on their interaction history as shown in Figure 7. Subsequently, the recommendation system suggests items to the user as Figure 8.

A screenshot of a computer

Description automatically generated

**FIGURE 7.** Inspect the interest of the user (only the first four rows are displayed to conserve space)

By comparing the recommended items with the user's actual interests, we assess the system's effectiveness in accurately predicting and matching user preferences.

A screenshot of a computer

Description automatically generated**FIGURE 8.** Top 5 recommendations to the user with the proposed model (only the first four rows are displayed to conserve space)

# CONCLUSION

This paper presents an efficient hybrid recommendation system that developed through a detailed comparison of existing methods for effective implementation. The comparison involved Recall@N metrics, model testing performance based on content-based models, and weight change measurement in the hybrid approach. The selection of recall rate as the primary measurement criterion was justified by transparent visualizations and comprehensive explanations, showing the system's correlation with user trends. The successful performance of tests affirmed its ability to precisely represent user behavior, a great leap in recommendation system design, which aims to provide more personalized and optimized user experiences.

# References

1. H. Beyari, “Recent e-commerce trends and learnings for e-commerce system development from a quality perspective,” IJQR **15**(3), 797–810 (2021).
2. G. Singh, C. Sahasrabudhe, H. Kagalwala, T.Y. Kumar, S. Labhane, and P. Dhakate, “AI Powered Recommender Systems for E-commerce,” in *2024 Second International Conference Computational and Characterization Techniques in Engineering &amp; Sciences (IC3TES)*, (IEEE, Lucknow, India, 2024), pp. 1–5.
3. A.T. Ulgen, M. HASKUL, and YAZ Yayınları, “Developments in the Field of Engineering 2023,” (2023).
4. W.-E. Kong, T.-E. Tai, P. Naveen, K.-W. Ng, and L. D. Krisnawati, “Exploring generative AI recommender systems in e-commerce: Model, evaluation metric, and comparative review,” *Journal of Informatics and Web Engineering* **4**(3), 278–298 (2025).
5. W.-W. Lee, N. Hashim, and S. Al-Juboori, “User behaviour prediction in e-commerce using logistic regression,” *Journal of Informatics and Web Engineering* **4**(3), 299–323 (2025).
6. F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” Egyptian Informatics Journal **16**(3), 261–273 (2015).
7. Y.-C. Chen, L. Hui, and T. Thaipisutikul, “A collaborative filtering recommendation system with dynamic time decay,” J Supercomput **77**(1), 244–262 (2021).
8. R. Ahuja, A. Solanki, and A. Nayyar, “Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor,” in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, (IEEE, Noida, India, 2019), pp. 263–268.
9. L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, “A trust-based collaborative filtering algorithm for E-commerce recommendation system,” J Ambient Intell Human Comput **10**(8), 3023–3034 (2019).
10. Y. Guo, C. Yin, M. Li, X. Ren, and P. Liu, “Mobile e-Commerce Recommendation System Based on Multi-Source Information Fusion for Sustainable e-Business,” Sustainability **10**(1), 147 (2018).
11. B. Maleki Shoja, and N. Tabrizi, “Customer Reviews Analysis With Deep Neural Networks for E-Commerce Recommender Systems,” IEEE Access **7**, 119121–119130 (2019).
12. L. Zhou, “Product advertising recommendation in e-commerce based on deep learning and distributed expression,” Electron Commer Res **20**(2), 321–342 (2020).
13. Y.M. Latha, and B.S. Rao, “Product recommendation using enhanced convolutional neural network for e-commerce platform,” Cluster Comput **27**(2), 1639–1653 (2024).
14. Ananthi. P, Shanmugapriya. P, Aruna. S, and R. T.E, “E-Commerce Recommender System based on Customer Reviews Analysis using Deep Neural Network,” in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, (IEEE, Namakkal, India, 2023), pp. 827–831.
15. M. Ayub, M.A. Ghazanfar, M. Maqsood, and A. Saleem, “A Jaccard base similarity measure to improve performance of CF based recommender systems,” in *2018 International Conference on Information Networking (ICOIN)*, (IEEE, Chiang Mai, 2018), pp. 1–6.
16. C.N. Dang, M.N. Moreno-García, and F.D.L. Prieta, “An Approach to Integrating Sentiment Analysis into Recommender Systems,” Sensors **21**(16), 5666 (2021).
17. **Y. Gu, Z. Ding, S. Wang, L. Zou, Y. Liu, and D. Yin,** "Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems," in Proc. 29th ACM Int. Conf. Inf. Knowl. Manag. (CIKM ’20), 2020.
18. **G. Moreira,** "Articles Sharing and Reading from CI&T DeskDrop," Kaggle, 2016. [Online]. Available: https://www.kaggle.com/datasets/gspmoreira/articles-sharing-reading-from-cit-deskdrop