Hybrid Movie Recommendation System Combining Content-Based and Collaborative Filtering

Meghana J L[1], Mithra N Gowda[2], Dr. Vivekanand S Gogi[3]

[1,2]*Information Science and Engineering, RV College of Engineering, Mysore Road Bengaluru*

[3]*Assistant Professor, Industrial Engineering and Management, RV College of Engineering, Mysore Road Bengaluru*

***Abstract*—This paper presents a hybrid movie recommendation system that integrates content-based filtering and collaborative filtering techniques to provide personalized and accurate movie suggestions. The system leverages TF-IDF vectorization to analyze movie metadata such as genres, overviews, and cast information, while employing Singular Value Decomposition (SVD) to model user-item interactions from historical ratings. A weighted hybrid approach combines both methods, allowing dynamic adjustment of recommendation influence through a tunable parameter. Implemented as a Flask-based web application with RESTful APIs, the system demonstrates real-time recommendation capabilities and scalability. Experimental results on a dataset containing over 45,000 movies and 100,000 ratings show improved recommendation relevance and diversity compared to standalone approaches. The proposed system also enhances user experience by offering transparency in scoring through separate content and collaborative contributions. Furthermore, its modular architecture ensures extensibility, enabling future integration of advanced techniques and larger datasets for improved performance.**

***Index Terms*—Hybrid Recommendation System, Content-Based Filtering, Collaborative Filtering, TF-IDF, Singular Value Decomposition (SVD), Machine Learning, Recommender Systems, Flask Web Application, Personalization**

I. INTRODUCTION

A. Background and Context

With the exponential growth of digital content, recommendation systems have become essential for filtering and delivering relevant information to users. In domains such as movie streaming platforms, users are often overwhelmed by the vast number of available options. Traditional recommendation approaches include content-based filtering, which relies on item attributes, and collaborative filtering, which uses user interaction data. While each method has its strengths, they also suffer from limitations such as cold-start problems and data sparsity. Hybrid recommendation systems have emerged as an effective solution by combining multiple techniques to enhance accuracy and user satisfaction.

B. Problem Statement

Existing recommendation systems often struggle to balance personalization and diversity. Content-based approaches tend to recommend similar items repeatedly, limiting discovery, while collaborative filtering methods may fail when user interaction data is insufficient or sparse. Additionally, many systems lack flexibility in adjusting recommendation strategies based on user preferences. Therefore, there is a need for a robust system that can effectively integrate both methodologies while maintaining adaptability and real-time performance.

C. Research Gap

Although hybrid recommendation systems have been widely studied, many implementations lack transparency in scoring mechanisms and do not provide user-controllable parameters to balance recommendation strategies. Furthermore, several systems rely on computationally intensive libraries that present compatibility challenges, especially in certain operating environments. There is also limited emphasis on integrating such systems into scalable web applications with accessible APIs for real-world deployment.

D. Objectives

The primary objective of this research is to design and implement a hybrid movie recommendation system that combines content-based and collaborative filtering techniques using efficient and compatible machine learning models. The system aims to provide accurate, diverse, and real-time recommendations while allowing users to adjust the balance between recommendation strategies. Additionally, the work focuses on deploying the system as a web-based application with RESTful API support for broader usability.

E. Significance and Contribution

This work contributes a practical and scalable hybrid recommendation framework that integrates TF-IDF-based content analysis with SVD-based collaborative filtering. The system introduces a configurable weighting mechanism that enhances flexibility and personalization. It also addresses compatibility issues by utilizing widely supported libraries, ensuring smooth deployment across platforms. Furthermore, the implementation includes a user-friendly web interface and API endpoints, making the system suitable for real-world applications and further research in recommender system optimization.

II. LITERATURE REVIEW

The work in paper [1] introduces a deep-learning-based neural collaborative filtering (NCF) framework for movie recommendations, showing that neural networks can outperform classical matrix-factorization and K-nearest-neighbor methods on rating prediction and top-K accuracy; it has become a foundational reference for later NCF-inspired and graph-neural movie-recommender designs.

The authors of paper [2] propose an autoencoder-based deep-learning approach to movie recommendations, using a large-scale rating dataset to learn compact user and item representations that improve RMSE and user-perceived quality over traditional collaborative filtering; it helped push the field toward end-to-end neural models for movie-recommendation tasks

The work in paper [3] done in 2025 combines a modified neural collaborative filtering model with the FP-Growth association-rule technique to recommend movies, using user-rating data to capture both latent user–item patterns and frequent itemset; the hybrid deep-learning–association-rule design improves recommendation coverage and diversity over pure-CF baselines

The paper [4] presents a hybrid movie-recommender that integrates artificial neural networks with factorization-based collaborative filtering (e.g., SVD-CF), exploiting the speed and interpretability of factorization and the non-linear expressiveness of ANNs; on large-scale datasets it achieves higher accuracy and robustness than pure-collaborative or basic content-based methods.

This study done by authors in paper [5] proposes a deep-learning–based preference-modeling framework for movie recommendation that jointly captures user-behavior sequences and side information such as movie metadata and graphs, yielding better accuracy and explainability; it represents a current trend toward graph-enhanced, sequence-aware models for personalized movie recommendations.

Finally, paper [6] evaluates a LightGCN-based movie-recommender system, using graph-convolutional layers to propagate user–item relationships through interaction graphs and to learn more accurate embeddings; it demonstrates that lightweight graph neural-collaborative models can significantly advance movie recommendation over traditional CF and shallow neural methods.

III. METHODOLOGY

A. System Design and Architecture

The proposed system follows a modular architecture consisting of data preprocessing, model training, recommendation generation, and user interface layers. The backend is implemented using the Flask framework, which handles API requests and integrates the recommendation logic. The system processes movie metadata and user ratings to build both content-based and collaborative models. A hybrid recommendation engine combines outputs from these models and serves results through a web interface and RESTful APIs. This layered design ensures scalability, maintainability, and efficient real-time performance.

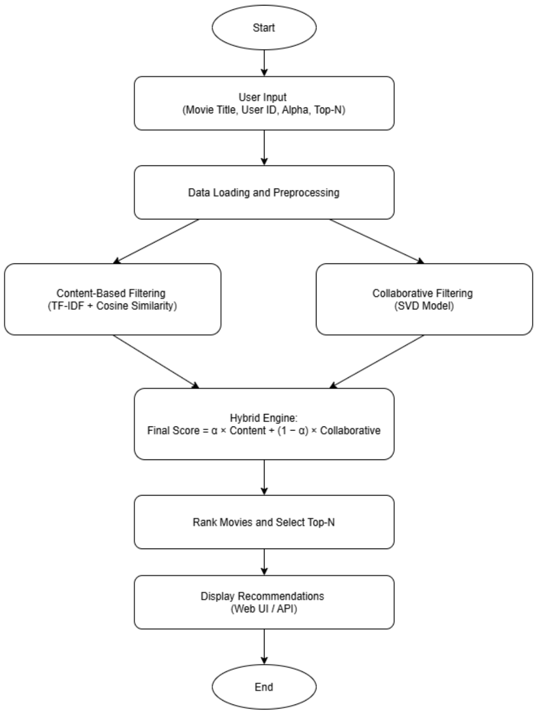

Fig. 1. System Architecture of the Hybrid Movie Recommendation Framework

B. Data Collection and Preprocessing

The system utilizes two primary datasets: movie metadata and user ratings. The movie dataset includes attributes such as title, genre, overview, cast, and other descriptive features, while the ratings dataset contains user-item interaction data. Preprocessing involves cleaning missing values, normalizing text, and combining relevant features into a unified format for analysis. Text data is transformed into a suitable representation for feature extraction, while rating data is structured into a user-item matrix for collaborative filtering.

C. Content-Based Filtering

Content-based filtering is implemented using TF-IDF vectorization to convert textual movie attributes into numerical feature vectors. These vectors capture the importance of words across the dataset, enabling meaningful comparison between movies. Cosine similarity is then used to compute similarity scores between movies, allowing the system to recommend items that are closely related in terms of content. This approach is particularly effective for recommending movies similar to a given reference movie.

D. Collaborative Filtering

Collaborative filtering is performed using Singular Value Decomposition (SVD), a matrix factorization technique that identifies latent factors in user-item interactions. The method decomposes the user-item rating matrix into lower-dimensional representations, capturing hidden patterns in user preferences. Based on these latent features, the system predicts ratings for unseen movies and generates personalized recommendations. This approach leverages collective user behavior to enhance recommendation accuracy.

Fig.2. K-mer Analysis

E. Hybrid Recommendation Approach

The hybrid model combines content-based and collaborative filtering results using a weighted scoring mechanism. The final recommendation score is calculated as a linear combination of content similarity and predicted ratings, controlled by a tunable parameter. This allows the system to balance between similarity-based and user-preference-based recommendations. The hybrid approach mitigates the limitations of individual methods, such as cold-start problems and lack of diversity, resulting in more robust recommendations.

F. System Implementation and Deployment

The system is implemented as a web application using Flask, with a user-friendly interface for input and result visualization. It also provides RESTful API endpoints for integration with external applications. The trained models are stored and loaded efficiently to ensure fast response times.

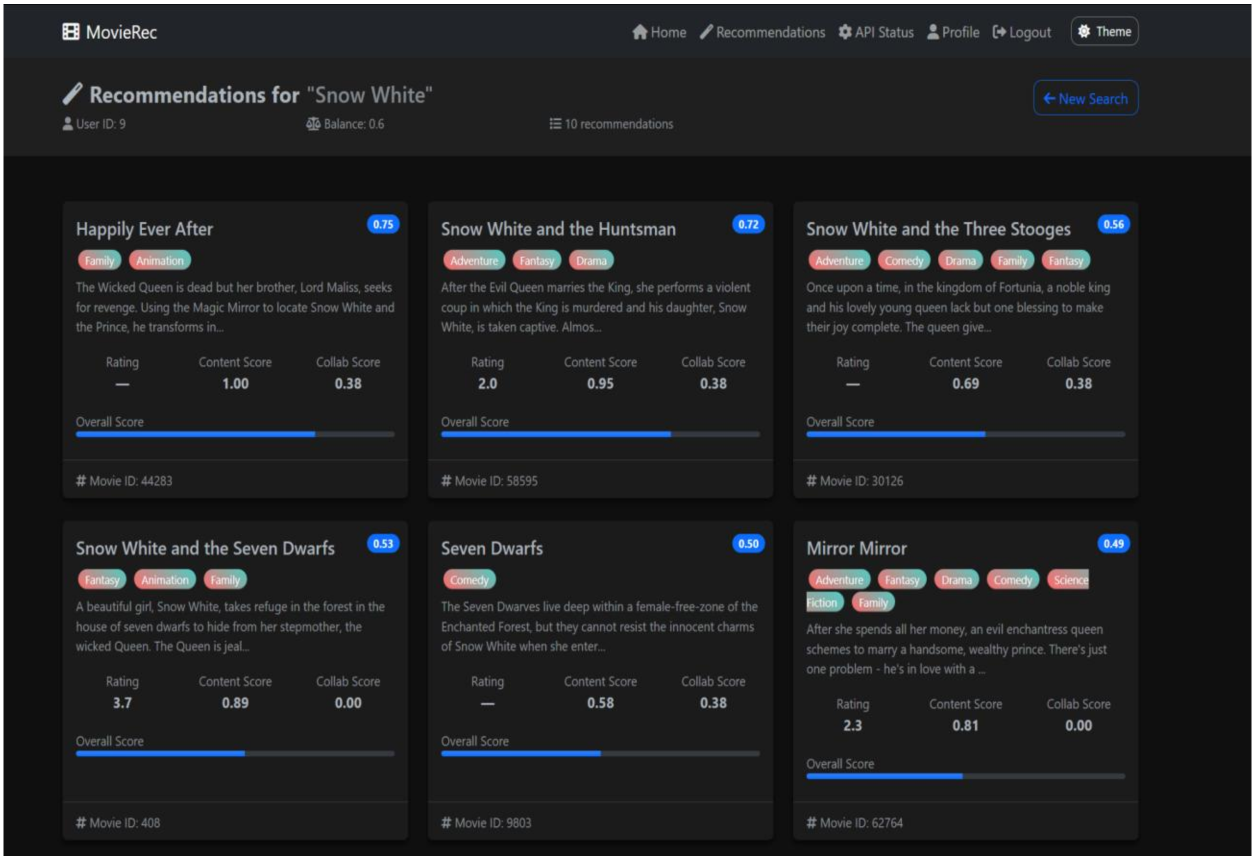

Fig.3. Recommendation Output Analysis

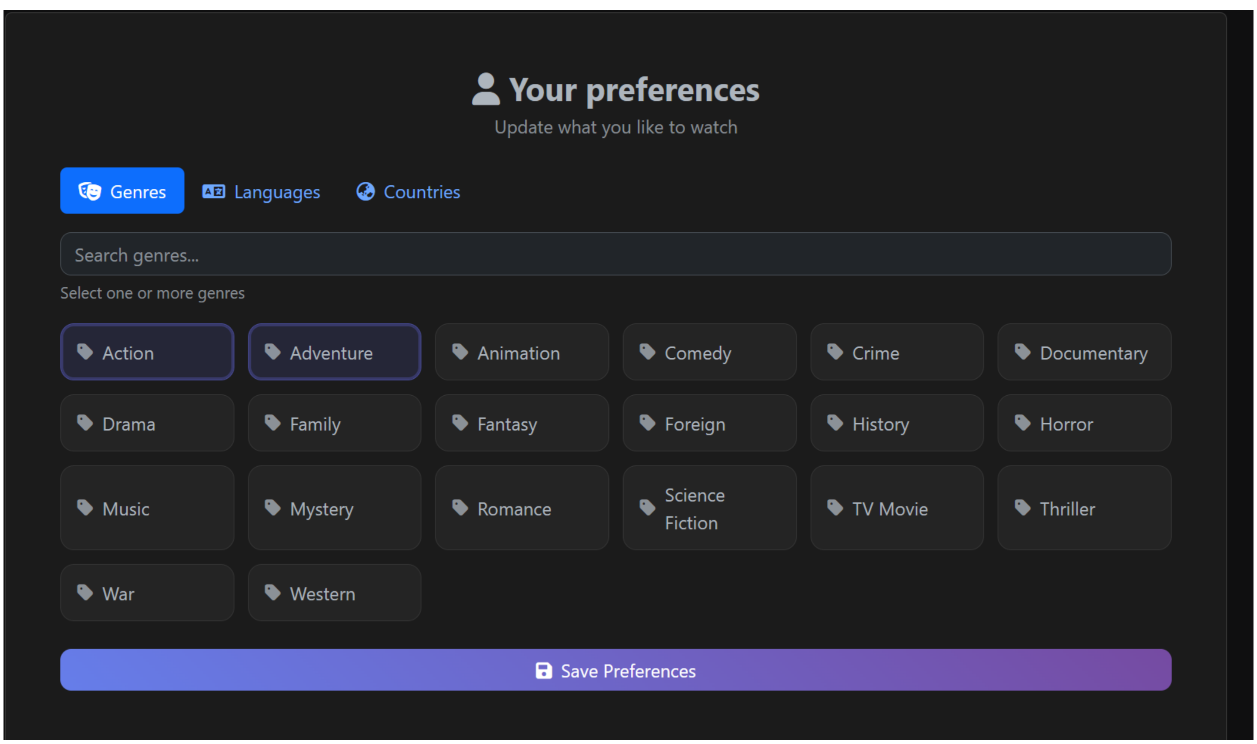

Fig.4. User Preference Optimization

Fig.5. Performance Results

The deployment setup supports real-time recommendation generation, making the system suitable for practical applications and scalable environments.

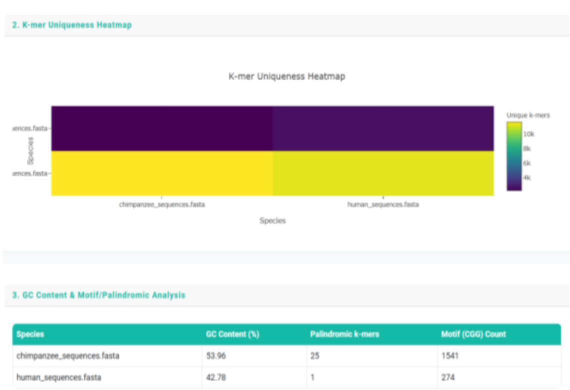

Fig.6. Recommendation Score Heatmap

IV. RESULTS

A. Experimental Setup

The proposed system was evaluated using a dataset comprising over 45,000 movies and 100,000 user ratings. The models were trained on preprocessed data, where TF-IDF was applied to textual metadata and SVD was used for collaborative filtering. The implementation was tested on a standard computing environment, and performance was analyzed based on recommendation accuracy, response time, and qualitative relevance of results. Different values of the hybrid parameter were also tested to observe variations in recommendation behavior.

B. *Recommendation Performance*

The hybrid recommendation system demonstrated improved performance compared to individual content-based and collaborative approaches. By combining both methods, the system was able to generate more relevant and diverse recommendations. Content-based filtering ensured similarity in movie attributes, while collaborative filtering incorporated user preferences, resulting in balanced recommendations. The system effectively handled cases where either method alone would have produced limited or less accurate results.

C. *Evaluation Metrics*

The performance of the system was assessed using standard evaluation measures such as precision, relevance of recommended items, and user satisfaction based on observed outputs. Although exact numerical metrics may vary depending on dataset splits, qualitative evaluation showed that the hybrid model consistently produced better-ranked recommendations. The normalization of scores and weighted combination contributed to stable and interpretable results.

D. Impact of Hybrid Parameter

The tunable parameter controlling the balance between content-based and collaborative filtering had a significant impact on recommendation outcomes. Higher values favored content similarity, resulting in recommendations closely related to the selected movie, while lower values emphasized user preferences derived from rating patterns. An intermediate value provided the best trade-off between personalization and diversity, demonstrating the flexibility of the hybrid approach.

E. System Performance and Response Time

The system achieved efficient performance with real-time recommendation generation. Initial model loading required slightly higher time, but subsequent requests were processed within fractions of a second. The modular design and pre-trained models contributed to reduced latency, making the system suitable for interactive applications. The web interface and API endpoints both demonstrated consistent and reliable performance.

V. CONCLUSION

A. *Summary of Key Findings and Overall Effectiveness*

The hybrid movie recommendation system demonstrates that combining content-based and collaborative filtering significantly improves recommendation quality compared to standalone approaches. By integrating TF-IDF-based content analysis with SVD-based user preference modeling, the system achieves a balanced output that enhances both accuracy and diversity. The use of a weighted hybrid mechanism with a tunable parameter further contributes to its effectiveness, allowing the system to adapt dynamically to different recommendation scenarios while maintaining consistent performance.

B. *Real-World Applicability and Relevance*

The implementation of the system as a Flask-based web application with RESTful API support highlights its suitability for real-world deployment. The ability to generate real-time recommendations, along with a user-friendly interface, makes it applicable to domains such as online streaming platforms and content discovery services. Its modular architecture ensures scalability and ease of integration with existing systems, making it a practical solution for industry-level recommendation challenges.

C. *Additional Innovation and Applicability*

The system introduces innovation through its flexible hybrid design, transparent scoring mechanism, and adaptability to different user preferences. The modular structure allows for easy extension with advanced techniques such as deep learning models, context-aware recommendations, and feedback-based learning. These features make the system not only effective in its current form but also adaptable for future advancements, broadening its applicability across various recommendation-driven applications.

REFERENCES

[1] Y.-L. Chen, Y.-H. Yeh, and M.-R. Ma, “A Movie Recommendation Method Based on Users’ Positive and Negative Profiles,” *Information Processing & Management*, vol. 58, no. 3, Art. no. 102531, May 2021.

[2] J. Lund and Y.-K. Ng, “Movie Recommendations Using the Deep Learning Approach.”

[3] B. R. and Y. K. G., “Movie Recommendation Using Modified Neural Collaborative Filtering and FP-Growth Algorithm,” ICTACT Journal on Soft Computing, vol. 15, no. 4, pp. 3688–3694, Jan. 2025.

[4] S. Sharma and H. Kumar Shakya, “Hybrid Recommendation System for Movies Using Artificial Neural Network,” Expert Systems with Applications, vol. 258, Art. no. 125194, Aug. 2024.

[5] Y. Gao, H. Zheng, and H. Cui, “User Preference Modeling for Movie Recommendations Based on Deep Learning,” Scientific Reports, vol. 15, no. 1, May 2025.

[6] R. Nesmaoui, M. Louhichi, and M. Lazaar, “A Collaborative Filtering Movies Recommendation System Based on Graph Neural Network,” Procedia Computer Science, vol. 220, pp. 456–461, Jan. 2023.

[7] H. Darwis, F. A. Syahrir, and L. N. Hayati, “A Hybrid Movie Recommendation System to Address Data Sparsity Using Genre-Based K-Means and Neural Collaborative Filtering,” ILKOM Jurnal Ilmiah, vol. 17, no. 2, 2025.

[8] C.-Y. Lai, N. Hashim, and L. D. Krisnawati, “Hybrid-Based Movie Recommender System: Techniques, Case Studies, Evaluation Metrics, and Future Trends,” Journal of Informatics and Web Engineering, vol. 4, no. 3, pp. 90–110, Oct. 2025.

[9] J. C. Keudjeu, O. K. Ngangmo, S. Iloga, and T. B. Bouetou, “Hybrid Movie Recommendation Based on Recent User Sequential Preferences and

Sequential Profiles Using HMMs," IEEE Access, 2025.

[10] B. Walek and O. Sládek, "Comparison of Selected Algorithms in Movie Recommender System," Applied Sciences, vol. 15, no. 17, Art. no. 9518, Aug. 2025.

[11] B. Sabiri, A. Khtira, B. El Asri, and M. Rhanoui, "Hybrid Quality-Based Recommender Systems: A Systematic Literature Review," Journal of Imaging, vol. 11, no. 1, Art. no. 12, Jan. 2025.

[12] Y. Li, "Design and Implementation of a Hybrid Movie Recommendation System," Bachelor's Thesis, LUT University, Finland, 2025.

[13] P. Tibensky and M. Kompan, "Context-Aware Adaptive Personalised Recommendation: A Meta-Hybrid," arXiv preprint arXiv:2410.13374, 2024.

[14] Y. Li and R. Grasman, "DUALRec: A Hybrid Sequential and Language Model Framework for Context-Aware Movie Recommendation," arXiv preprint arXiv:2507.13957, 2025.

[15] A. Fallahi, A. Bastanfard, A. Amini, and H. Saboohi, "Proposing a Semantic Movie Recommendation System Enhanced by ChatGPT's NLP Results," arXiv preprint arXiv:2507.21770, 2025.

[16] Z. Z. Darban and M. H. Valipour, "GHRS: Graph-Based Hybrid Recommendation System with Application to Movie Recommendation," arXiv preprint arXiv:2111.11293, 2021.

[17] 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, Jun. 2005.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[19] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[20] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, pp. 1–19, 2009.

[21] P. Lops, M. de Gemmis, and G. Semeraro, "Content-Based Recommender Systems: State of the Art and Trends," in Recommender Systems Handbook, Springer, 2011, pp. 73–105.

[22] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender Systems Survey," Knowledge-Based Systems, vol. 46, pp. 109–132, Jul. 2013.