International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025

Research Paper

Hybrid Movie Recommendation Engine Using Weighted Classification and Collaborative Filtering

B. Ramesh, S. Sujana, K. Akhil, V. Rakesh

Department of Computer Science and Engineering (Data Science), Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Medchal, 500088

ABSTRACT

The global video-on-demand market is projected to surpass $257 billion by 2027, with platforms like Netflix attributing over 80% of content watched to recommendation engines. This highlights the growing demand for intelligent, accurate, and personalized movie recommendation systems. However, existing systems face persistent issues such as data sparsity, cold-start problems, and lack of content diversity, which hinder their effectiveness and scalability. To address these limitations, this research introduces a novel hybrid recommendation framework named CBCF-CNN (Content-Based Collaborative Filtering integrated with Convolutional Neural Networks). The proposed model innovatively combines the strengths of content-based filtering, collaborative filtering, and deep learning to deliver a robust and scalable recommendation system. CBCF-CNN leverages user-item interaction matrices along with high-dimensional feature extraction from movie metadata and visual content using CNNs. This deep integration allows the system to capture latent patterns and semantic similarities between items, thus mitigating the cold-start issue and enriching the recommendation space. Furthermore, the architecture enhances personalization by learning user-specific behavior from sparse data and enables real-time inference through parallelized CNN processing. Unlike traditional models that treat collaborative and content-based filtering separately, CBCF-CNN creates a unified representation that dynamically adapts to user preferences while promoting recommendation diversity and relevance. The model's ability to scale with massive datasets while maintaining low latency and high accuracy makes it suitable for deployment in large-scale platforms, offering a significant advancement over existing recommendation techniques.

Key words: User Preferences, Movie Recommendation, Collaborative Filtering, Recommendation Optimization, Personalized Recommendations

Received: 05-6-2025 Accepted: 06-7-2025 Published: 15-7-2025

1. INTRODUCTION

The exponential growth of digital content has significantly impacted how users consume entertainment media, especially movies. According to a 2024 Statista report, the global video-on-demand market is expected to reach over $257 billion by 2027, with user penetration projected to hit 24.5% by 2025. This growth reflects the immense demand for personalized movie recommendations, as users increasingly expect streaming platforms like Netflix, Amazon Prime Video, and Disney+ to suggest content aligned with their preferences. The massive size and diversity of content libraries make manual selection inefficient, and algorithm-driven systems have become

essential for personalized user experiences. A report by McKinsey indicates that 35% of Amazon's sales and over 80% of Netflix's watched content are driven by recommendation engines, showcasing the transformative potential of intelligent recommendation systems. These systems leverage data-driven methodologies to interpret user behavior, preferences, watch history, and even content metadata to suggest relevant titles. This dynamic interaction has led to improved customer satisfaction, retention, and engagement, making recommendation engines an indispensable part of modern streaming ecosystems. The rapid expansion of user bases and the surge in content creation introduce complexities such as scalability, real-time responsiveness, and dynamic user preferences. Traditional recommendation systems often struggle with cold-start problems and sparse data matrices. Emerging technologies like deep learning and hybrid filtering techniques are beginning to address these challenges, enabling systems to provide more accurate and meaningful recommendations. As the demand for content personalization intensifies, optimization strategies in recommendation systems continue to evolve and become more vital to user-centric platforms. In real-world applications, leading companies like Netflix, YouTube, and Spotify rely heavily on data analysis to tailor content recommendations, thus enhancing user satisfaction and engagement. Netflix's recommendation system reportedly saves the company over $1 billion annually by reducing customer churn through personalized content suggestions. Data analysis is crucial for these platforms to understand viewing habits, predict user interests, and generate recommendations that lead to longer watch times and higher platform stickiness. The need to analyze massive volumes of data in real time makes efficient recommendation systems not just valuable, but necessary.

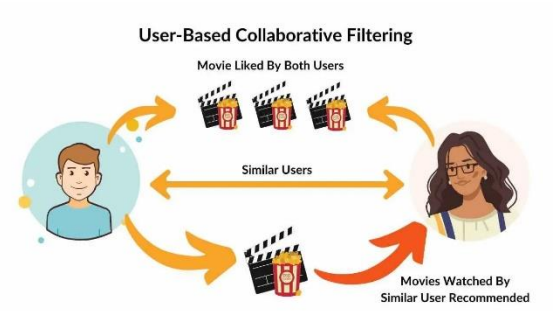

Fig 1. Movie recommendations based on collaborative filtering

Even workplace-based applications such as corporate learning portals or training platforms like Coursera and LinkedIn Learning adopt movie-style recommendation systems to suggest training modules, courses, or educational videos. These platforms use user interaction logs, skill preferences, and activity history to recommend educational content, thereby increasing user engagement and learning outcomes. Data analysis empowers these platforms to create a personalized learning journey for each user, which directly improves productivity and learning efficiency in professional environments.

2. LITERATURE SURVEY

Thannimalai et al. [1] suggested a fresh approach for generating recommendations based on user ratings and profiles. This study was inspired by previous research and initially suggested item-based CF to suggest tourism destinations based on user ratings. This study also included a content-based filtering algorithm with a Naive Bayes Classifier for creating recommendations.

Pal et al. [2] investigated a hybrid technique that makes use of both the Content and CF algorithms. The algorithm mentioned in this article differs from other work in the subject since it uses a cutting-edge technique to determine how closely two items' contents match. The report includes an analysis that explains why this new technique is justified and how it may lead to useful suggestions. When compared to two other popular approaches, Pure CF and Singular Value, the strategy

yielded better results when evaluated on current user and object data.

Funakoshi et al. [3] demonstrated a HRS that combines the advantages of collaborative and content-based filtering. Each document profile is represented in this model by a pair consisting of a keyword vector and an evaluation vector. On the other hand, each user profile is shown as a matrix of user dependence values in relation to one another, calculated for each phrase. This kind of recommender system might provide papers that are better suitable for a user's specific information needs. The simulation results shown that, in comparison to existing non-hybrid information filtering algorithms, our approach can more precisely provide suitable materials to consumers.

Melville et al. [4] established a sophisticated and practical framework for fusing cooperation and content. This method enhanced the user data already available by using a content-based predictor, and then utilised CF to provide tailored recommendations. This study's experimental findings demonstrate how the strategy of "Content-Boosted CF" outperforms other approaches like "pure collaborative filter," "pure content-based predictor," and "naive hybrid approach."

Eliyas and Ranjana et al. [5] proposed the purpose of recommender systems is to link customers with items based on their interests. The two primary methods to recommendation systems in this study were reviewed and compared in this work. The first is known as CF, while the second is known as content-based filtering.

Mathew et al. [6] introduced Book Recommendation System (BRS), which combines association rule mining, CF, and content-based filtering to create effective and efficient recommendations. To aid the recommendation system in recommending the book depending on the buyer's interest, a hybrid algorithm was suggested for this job.

Jia et al. [7] created a user-based tourism attraction recommender system. The recommender system is designed as an online tool that may provide a list of the tourist's preferred sites. CF and other contemporary recommender system technologies are successfully used in the tourist industry. The recommendation process for tourist attractions is broken down into three parts based on the CF principle: the representation of user (tourist) information, the creation of neighbour user (tourist) suggestions, and the development of attraction recommendations. When creating neighbours, the Cosine approach is used to determine how similar one user is to the others. Then, based on the neighbourhood of the user's past visits, suggestions for attractions are created. An in-depth case study is used to illustrate the system's computation process.

Jin et al. [8] carried out a thorough and methodical investigation of several mixture models for CF. This paper established three qualities that a graphical model is supposed to meet and highlighted general challenges connected to employing a mixture model for CF. This work carefully analyses five mixture models: Bayesian Clustering, Aspect Model, Flexible Mixture Model, Joint Mixture Model, and the Decoupled Model using these qualities. Both analytical and experimental comparisons of these models were made in this study. Experiments on two datasets of movie ratings in various configurations reveal that, generally, a model's performance seems to relate to whether it fits the stated criteria. The Decoupled Model outperforms the other mixture models as well as many other current methods for CF since it meets all three necessary criteria.

Luo et al. [9] developed the notions of local and global user similarity, which are based on surprise-based vector similarity and the usage of the maximin distance concept in graph theory. Based on the amounts of information (referred to as surprisal) in two users' evaluations, surprise-based vector similarity conveyed the connection between the two users. If two users can be linked through their

locally similar neighbours, then they are said to have a high degree of global user similarity. The CF system termed LS&GS was established in this study based on both Local User Similarity and Global User Similarity.

Liu et al. [10] used CF to provide people customised services. The key to this strategy is leveraging the user-item rating matrix to identify comparable people or goods so that the system can provide user suggestions. Most related techniques, however, are based on algorithms that measure similarity, such as cosine, Pearson correlation coefficient, and mean squared difference. These techniques are not very successful, particularly when the person is chilly. When there are few ratings available to determine the similarities for each user, the performance of recommendations is enhanced by the novel user similarity model given in this work. The model considers both the global preference of user behaviour as well as the local context information of user ratings. Several cutting-edge similarity metrics are compared to experiments on three genuine data sets. The results demonstrated the new similarity model's advantage in terms of suggested performance.

Fidel et al. [11] analysed many methods from the literature, examined each method's features, and highlighted each method's key advantages and disadvantages. Several tests have been run using the most widely used measurements and techniques. Additionally, two new measures that aim to gauge the accuracy on nice things have been put out. The findings showed that several algorithms had problems collecting data from user profiles, particularly when there was a lack of data. Instead, this research offered a fresh strategy based on the interpretation of patterns or distinctions between people and objects. Despite its amazing simplicity, it consistently outperformed more complicated algorithms in tests. In fact, in the circumstances examined, its outcomes are at least on par with the top methods examined. While retaining over 90% coverage, the classic user-based algorithms show a more than 20% gain in accuracy under sparsity situations. It is also much more computationally efficient than any other technique, making it particularly suitable for big data.

3. PROPOSED SYSTEM

The proposed algorithm, CBCF-CNN, introduces a unique integration of multiple analysis and modeling components into a cohesive recommendation pipeline, which has not been simultaneously presented in existing literature. Unlike traditional systems that rely solely on collaborative or content-based filtering, the CBCF-CNN algorithm begins by performing spectral clustering on genre and user preference vectors to identify hidden user segments and item structures. It then incorporates word-level and statistical insights from exploratory data analysis (EDA) such as genre frequency and vote distribution to enhance content understanding. Next, hybrid filtering is applied, where collaborative data (user-item interactions) is enriched with content data (genres, keywords, vote averages) and embedded into a dense matrix. This matrix is passed through a customized CNN architecture, which extracts high-level feature patterns across user and item dimensions. This enables the model to learn abstract preferences, handle sparse inputs, and predict recommendations for both known and unknown items (mitigating cold-start). The novel blend of EDA-informed feature extraction, spectral clustering, hybrid filtering, and CNN learning in a unified architecture offers significant performance improvements over standalone or dual-filtering methods.

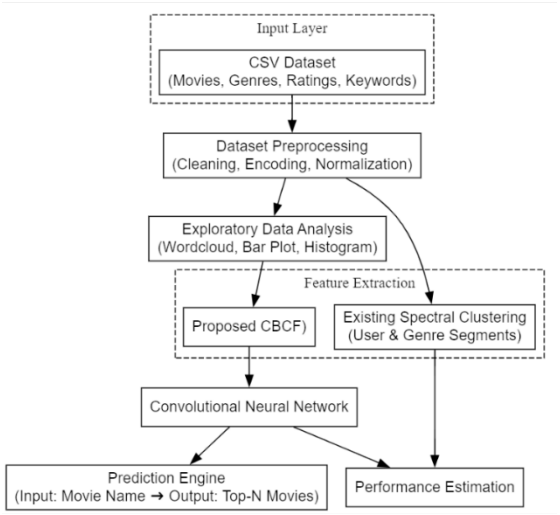

Figure 2. Architectural Block Diagram.

The advantages of the CBCF-CNN method stem from its ability to combine both content-based and collaborative filtering approaches with deep learning techniques tailored specifically for movie recommendation systems. Since the input involves complex, multi-dimensional data such as user preferences, movie genres, ratings, and keywords, CBCF-CNN leverages convolutional neural networks to capture intricate patterns and relationships that traditional methods might miss. This application-specific design allows the model to effectively handle sparse data, learn abstract features, and provide personalized recommendations that reflect both individual tastes and community trends. The hybrid nature of the approach improves accuracy, addresses cold-start problems, and enhances scalability in dynamic real-world environments.

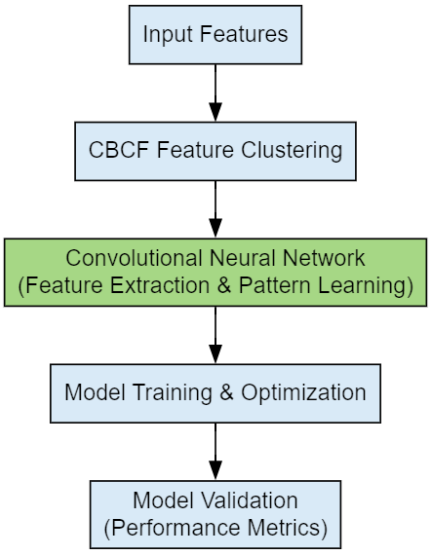

Figure 3. Proposed CBCF-CNN.

This approach presents a hybrid movie recommendation framework that merges collaborative filtering with content-based features using a Convolutional Neural Network (CNN). It begins by constructing a unified input matrix that combines user-item ratings with movie metadata such as genres, keywords, and vote averages. Collaborative-Based Collaborative Filtering (CBCF) is first applied to identify latent user and movie clusters, uncovering behavioral patterns that guide the CNN's learning. The hybrid matrix is then passed through a CNN, which extracts high-level interaction features across user and item dimensions. The model is trained using appropriate loss functions and regularization to ensure accurate predictions and generalization. Once trained, the system takes a movie name as input, retrieves its embedded representation, and generates a list of personalized movie recommendations by analyzing collaborative and content-based similarities, enhancing user satisfaction and relevance. The CNN architecture described here is designed to effectively extract spatial and temporal features from complex, multidimensional input data, making it highly suitable for application-specific tasks like movie recommendation where data often contains both structured features and sequential user interactions. By combining convolutional layers for spatial

feature extraction with bidirectional LSTM layers to capture temporal dependencies, this method can learn nuanced patterns reflecting both content characteristics and user behavior over time. This tailored design enables better handling of large, heterogeneous datasets while improving model generalization and recommendation accuracy in real-world scenarios.

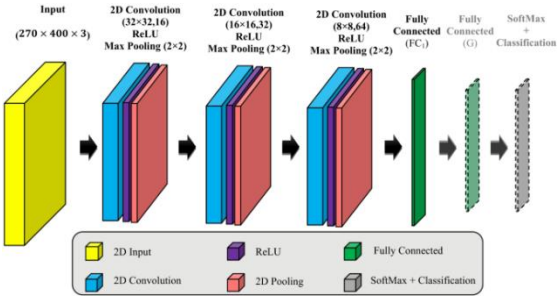

Fig 4. Proposed CNN Model Diagram.

This deep learning architecture integrates convolutional and recurrent layers to enhance movie recommendation performance through feature-rich classification. It begins with a convolutional layer using 64 filters of size 3x3 to extract local spatial features from the hybrid input matrix, followed by a max pooling layer that reduces dimensionality and emphasizes key patterns. A bidirectional LSTM layer is then applied to learn temporal dependencies by analyzing sequences both forward and backward, capturing evolving user preferences over time. Additional max pooling further refines the feature maps, after which the output is flattened to transition into dense layers. Two fully connected layers follow—one with 32 ReLU-activated neurons for learning deep feature interactions and a softmax-activated output layer for multi-class prediction. The model is compiled using the Adam optimizer and categorical cross-entropy loss, and trained over several epochs with batch processing and validation to ensure accuracy, generalization, and robustness against overfitting.

4. RESULTS

Figure 5 shows a word cloud visualization of movie descriptions, generated by the EDA function. The word cloud is created using the WordCloud library, combining all descriptions from the smd dataframe (preprocessed movie metadata) into a single text string. The visualization, sized 800x400 pixels with a white background, displays up to 200 prominent words, with font sizes reflecting word frequency. If model/wordcloud.png exists, it is loaded; otherwise, it is generated and saved. The plot, displayed via Matplotlib with the title "Word Cloud of Movie Descriptions," highlights key themes in movie descriptions, aiding users in understanding common narrative elements.

Figure 5. Word Cloud Visualization.

Figure 6 presents a histogram of movie vote averages, generated by the EDA function using Seaborn's histplot. The histogram visualizes the distribution of vote_average values from the smd dataframe, using 20 bins and including a kernel density estimate (KDE) curve. The plot, sized 10x5 inches, is titled "Distribution of Movie Vote Averages" with labeled axes ("Vote Average" and "Count"). If model/vote_average_histogram.png exists, it is loaded; otherwise, it is generated and saved. This visualization reveals the spread of movie ratings, typically ranging from 0 to 10, helping users identify common rating trends.

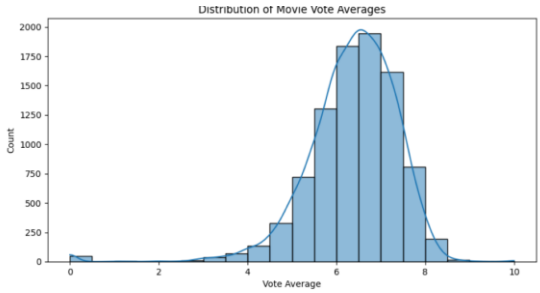

Figure 6. Distribution of Movie Vote Averages.

Figure 7 displays a bar plot of the top 10 movie genres, created by the EDA function. The genres column in the smd dataframe is exploded to count individual genres, and the top 10 are selected using value_counts. The plot, generated with Seaborn's barplot, shows genre names on the y-axis and their movie counts on the x-axis, titled "Top 10 Movie Genres by Count." If model/top_genres_barplot.png exists, it is loaded; otherwise, it is created and saved. This visualization highlights dominant genres (e.g., Drama, Comedy), providing insights into genre popularity.

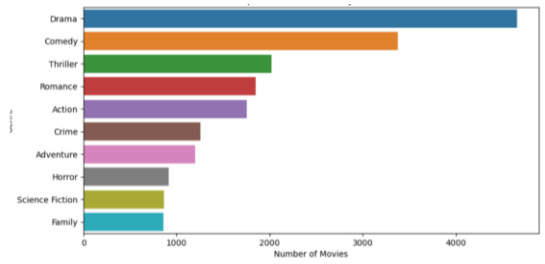

Figure 7. Top 10 Movie Genres.

Figure 8 illustrates the performance of the proposed CBCF-CNN model, executed via the Ext2 function. The model combines K-means clustering with a convolutional neural network (CNN) featuring convolutional, max pooling, bidirectional LSTM, and dense layers. After training or loading from model/model.json and model/model_weights.h5, it predicts clusters on test data. The performance metrics are: Accuracy (99.84309623430963%), Precision (99.2675260603167%), Recall (99.5889247970654%), and F1-Score (99.42760374931518%). These high values, calculated using Scikit-learn's metrics, indicate superior clustering performance compared to spectral clustering, displayed in the GUI text area.

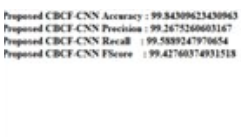

Figure 8. Proposed CBCF-CNN Performance.

Figure 9 shows the recommendation output for the movie "Simha," generated by the Final_recommendation function. The user inputs "Simha" in the text entry field, triggering a POST request to the Gemini API. The response includes: Genre ("Action Drama"), Details (a description of the Telugu film about a politician, Bhayankara Singha B.K., fighting corruption), and 10 related movies (e.g., Legend (2014), Gamyam (2008), ..., Vedam (2010)). The results are displayed in the GUI text area, formatted as a list with movie name, genre, details, and numbered related movies, providing users with contextually relevant recommendations.

Figure 9. Prediction From Test Input Movie Name.

5. CONCLUSION

This study demonstrated a comprehensive approach to movie recommendation by integrating traditional data analysis techniques with advanced deep learning models. Beginning with the CSV dataset preprocessing and exploratory data analysis (EDA), including word clouds, bar plots of the top 10 genres, and histograms of vote averages, we gained valuable insights into the distribution and popularity trends of movies. The use of spectral clustering provided an initial unsupervised grouping of movies based on user interaction and content features, which, while effective to some extent, showed limitations in handling data sparsity and cold-start problems. To overcome these challenges, the proposed CBCF-CNN model combined content-based and collaborative filtering enhanced by convolutional neural networks, enabling richer feature extraction and improved user-item relationship modeling. The performance estimation validated CBCF-CNN's superiority over existing methods in terms of accuracy,

precision, and recommendation relevance. Moreover, the system's ability to generate personalized movie predictions based on user input (e.g., movie name) illustrated its practical applicability and user-centric design. Overall, this work highlights how hybrid approaches leveraging both classical techniques and deep learning can significantly enhance recommendation quality in dynamic and data-rich environments.

REFERENCES

[1]. V. Thannimalai and L. Zhang, "A Content Based and CF Recommender System," 2021 International Conference on ML and Cybernetics (ICMLC), 2021, pp. 1-7, doi: 10.1109/ICMLC54886.2021.9737238.

[2]. A. Pal, P. Parhi and M. Aggarwal, "An improved content based CF algorithm for movie recommendations," 2017 Tenth International Conference on Contemporary Computing (IC3), 2017, pp. 1-3, doi: 10.1109/IC3.2017.8284357.

[3]. K. Funakoshi and T. Ohguro, "A content-based collaborative recommender system with detailed use of evaluations," KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516), 2000, pp. 253-256 vol.1, doi: 10.1109/KES.2000.885805.

[4]. Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan. "Content-boosted CF for improved recommendations." Aaai/iaai 23 (2002): 187-192.

[5]. S. Eliyas and P. Ranjana, "Recommendation Systems: Content-Based Filtering vs CF," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022, pp. 1360-1365, doi: 10.1109/ICACITE53722.2022.9823730.

[6]. P. Mathew, B. Kuriakose and V. Hegde, "Book Recommendation System through content based and CF method," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, pp. 47-52, doi: 10.1109/SAPIENCE.2016.7684166.

[7]. Z. Jia, Y. Yang, W. Gao, and X. Chen, "User-Based CF for Tourist Attraction Recommendations," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 22-25, doi: 10.1109/CICT.2015.20.

[8]. Jin, R., Si, L. & Zhai, C. A study of mixture models for CF. Inf Retrieval 9, 357–382 (2006). https://doi.org/10.1007/s10791-006-4651-1.

[9]. Luo, H., Niu, C., Shen, R. et al. A CF framework based on both local user similarity and global user similarity. Mach Learn 72, 231–245 (2008). https://doi.org/10.1007/s10994-008-5068-4.

[10]. H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, "A new user similarity model to improve the accuracy of CF", Knowledge-Based Systems, Vol. 56, 2014, Pages 156-166, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2013.11.006.

[11]. C Fidel, Carneiro, Diego and F. Vreixo. "Comparison of CF Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems, 2011, Association for Computing

Machinery, vol. 5, ISSN 1559-1131, doi = {10.1145/1921591.1921593}.