# International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 2 (2025)



ijerst.editor@gmail.com

editor@ijerst.com

*ISSN 2319-5991* www.ijerst.com
*Vol. 21, No. 2, 2025*


***Research Paper***

# DEEP LEARNING–DRIVEN MOVIE RECOMMENDATION USING CBCF-CNN WITH TEXTUAL FEATURE VISUALIZATION

Halavath Peda Sydulu[1*], G.Manideep[2], G.Akshitha[2], G.Varun[2], G.Geethanjali[2]
[1]*Assistant Professor,* [2]*UG Student,* [1,2]*Department of Computer Science and Engineering,*
[1,2]*Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Telangana, India*
**Corresponding author: saidulucse@gmail.com*

**ABSTRACT**

The rapid expansion of the global video-on-demand (VoD) market, projected to exceed USD 257 billion by 2027, underscores the critical role of intelligent recommendation systems, with platforms such as Netflix reporting that over 80% of viewed content is driven by recommender engines. Despite their success, existing recommendation systems continue to suffer from challenges including data sparsity, cold-start problems, and limited content diversity, which restrict their scalability and personalization capabilities. To overcome these limitations, this research proposes a novel hybrid recommendation framework termed CBCF-CNN, which integrates Content-Based Filtering (CBF), Collaborative Filtering (CF), and Convolutional Neural Networks (CNNs) into a unified architecture. The proposed model exploits user–item interaction matrices alongside deep feature extraction from movie metadata and visual content using CNNs, enabling effective learning of latent representations and semantic similarities. This deep integration mitigates cold-start issues, enhances recommendation diversity, and improves personalization even under sparse data conditions. Furthermore, the parallelized CNN-based architecture supports real-time inference with low latency, making the model suitable for large-scale deployment. Overall, CBCF-CNN delivers improved accuracy, scalability, and adaptability compared to traditional recommendation approaches.

**Keywords:** Movie Recommendation System, Hybrid Recommendation, Collaborative Filtering, Content-Based Filtering, Convolutional Neural Networks, Cold-Start Problem, Data Sparsity, Deep Learning, Personalized Recommendation.

Received: 02-03-2025 Accepted: 27-04-2025 Published: 06-05-2025

## 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.

## 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.

Schafer et al. [12] explained the fundamental ideas of CF, its main applications for users of the adaptive web, the theory and use of CF algorithms, and design choices for rating systems and rating acquisition. The emergence of rich interaction interfaces and how to assess CF systems were also explored in this article. This paper concluded with considerations of significant unanswered research problems in the area and privacy challenges specific to a CF recommendation service.

Bobadilla et al. [13] introduced a new similarity metric that outperforms the best outcomes from existing measures by employing optimization based on neural learning. When used in new user cold start scenarios, the metric has been evaluated on the Netflix and Movielens databases, yielding significant gains in the metrics of accuracy, precision, and recall. The work offered a mathematical formalisation outlining how to use leave-one-out cross validation to derive the primary quality measurements of a recommender system.

Bobadilla [14] et al. determine the leveraging contextual data from the whole user base to determine the singularity that occurs for each item in the votes given by each user pair that you desire to compare, standard similarity metrics may be achieved. Therefore, the influence on similarity will increase with the measure of uniqueness between the votes made by two particular people. The findings, which were evaluated on the Movielens, Netflix, and FilmAffinity databases, confirm the good performance of the suggested singularity metric.

Ji et al. [15] suggested a user-based CF method. The approach primarily addresses the issues of data sparsity and cold start in conventional CF algorithms. The user preference degree and trust degree are added to increase the accuracy of the recommendation results after the time-interest weight function is first established via the study of user behaviour to enhance the modified cosine similarity formula. Finally, the experimental findings using the hetrec2011 dataset demonstrate that the enhanced CF algorithm greatly improves upon the standard recommendation algorithm in terms of the accuracy of the recommendations it generates. Effective solutions are provided for the issues of cold start and data sparsity.

Pazzani et al. [16] mentioned content-based recommendation systems, i.e., systems that suggest a product to a user based on the product's description and the user's interest profile. Systems for suggesting web pages, news articles, restaurants, television shows, and merchandise are all examples of content-based recommendation systems in use. Content-based recommendation systems all share a way to describe the items that might be recommended, a way to build a user profile that describes the kinds of items the user likes, and a way to compare items to the user profile to decide what to re commend, even though the specifics of various systems vary. When the user provides comment on the attractiveness of products that have been offered to them, the profile is often automatically constructed and updated.

Kulkarni et al. [17] focused on cutting-edge AI approaches that employ contextual data to enhance traditional design. These methods are further divided into statistical computing methods and bio-inspired computing methods. This overview presents the research on these strategies that discusses how they might address problems with context-aware recommendation systems. The survey also discusses context inclusion strategies, categorization of the contexts used in the reviewed literature, their impact on the issues that recommendation systems face, effective usage of these contexts, datasets used in the domain, future research scope in all the reviewed techniques, and general future research directions and challenges.

Harper et al. [18] outlined the development of MovieLens and its databases. This work discussed the lessons that a research organisation may learn from managing an ongoing, live research platform. The recommended practises and restrictions for utilising the MovieLens datasets in new research were outlined in this paper.

## 3. PROPOSED METHODOLOGY

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.

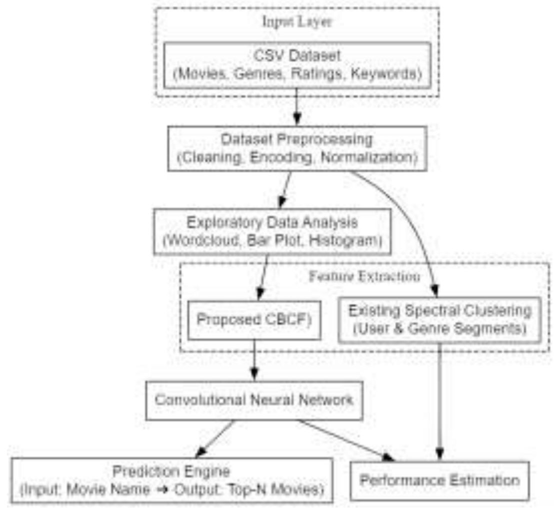


Fig. 1: Architectural Block Diagram.

**Step 1: CSV Dataset Collection and Structuring** Begin by importing a comprehensive CSV dataset that includes structured fields such as movie titles, genres, keywords, user ratings, vote averages, and IDs. Each data row is validated for missing or null values. Duplicate entries are removed, and categorical values like genres are expanded into binary columns using multi-hot encoding to enable machine readability.

**Step 2: Dataset Preprocessing** Perform data cleaning by removing or filling null entries, normalizing numeric features such as vote counts and averages, and encoding categorical variables. Apply tokenization to textual fields like keywords and use label encoding or one-hot encoding on user-based categories. A unified dataset is then formed that integrates user profiles and movie metadata into a common format.

**Step 3: Exploratory Data Analysis (EDA)** Analyze the dataset with multiple exploratory techniques to extract feature insights. Generate a word cloud of genres to visualize dominant categories, a bar plot of the top 10 most frequent genres, and a histogram of vote averages to understand rating distributions. These statistical patterns help in building informative feature embeddings and normalizing outlier values.

**Step 4: Spectral Clustering (Existing Technique)** Apply spectral clustering on the user-item matrix using similarity measures derived from genre overlaps and vote metrics. This groups users and movies into latent segments, revealing hidden behavioral patterns. These clusters are used to pre-train the model and guide the initial weight distribution of the CNN to improve convergence speed and prediction accuracy.

**Step 5: Hybrid Feature Integration for CBCF** Construct a hybrid matrix combining collaborative filtering scores (user-movie ratings) and content-based features (genres, keywords, vote averages). Normalize and embed this matrix using dense vectors, then align it along structured dimensions representing user preferences and item characteristics.

**Step 6: Convolutional Neural Network (CNN) Processing** Feed the hybrid matrix into a custom-built CNN model. The CNN consists of convolutional and pooling layers that learn abstract interaction patterns and genre-topic relevance. The model is trained to minimize recommendation error using cross-entropy loss and adaptive optimizers. Dropout layers are included for generalization.

**Step 7: Performance Estimation** Evaluate the trained model using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Top-N recommendation accuracy. Compare its performance against the spectral clustering baseline and traditional filtering methods to demonstrate improvements in both precision and recall.

**Step 8: Prediction From Test Input (Movie Name Input)** Allow the user to input a movie title. The system parses the input, retrieves its embedded feature vector, and computes similarity with clustered user segments and CNN-encoded profiles. It then returns the top recommended movies personalized to the inferred user taste, ensuring relevance based on content and collaborative insight.

**9.3 Result and Description**

Figure 2 shows the user interface (UI) of the Movie Recommendation System, implemented using Tkinter. The UI is designed with a fixed window size of 1300x1200 pixels and features a title label at the top: "Optimizing Movie Recommendation: A Weighted Classification and Collaborative Filtering Approach," styled with a powder blue background and olive drab text. The interface includes five buttons aligned vertically on the left: "Upload MovieLens Dataset," "Word Cloud Visualizations," "Run Spectral Clustering," "Run Proposed CBCF-CNN," and "Send," each triggering specific functions. A text entry field (width 100) allows users to input a movie name for recommendations, and a scrollable text area (height 20, width 115) displays outputs such as dataset paths, clustering results, and recommendations. The UI serves as the central hub for user interaction, connecting all system functionalities.

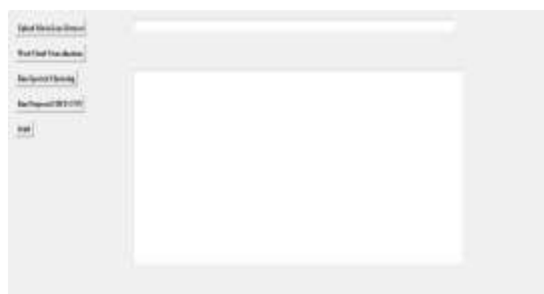

Fig. 2: UI Screen of Research work.

Figure 3 illustrates the dataset loading process, where users upload a MovieLens dataset (e.g., temp.csv, movies_metadata.csv, links_small.csv) via the "Upload MovieLens Dataset" button. The uploadDataset function uses Tkinter’s filedialog to select a CSV file, storing its path in the global filename variable. The path is displayed in the text area, confirming successful loading (e.g., "Dataset/temp.csv Dataset loaded"). The system preprocesses the dataset using Pandas, handling missing values (e.g., filling NaN in tagline and description) and converting data types (e.g., tmdbId to integers).

IJERST International Journal of Engineering Research and Science & Technology

This step ensures the dataset is ready for clustering and recommendation tasks, with global variables like ratings_f3 and test storing the processed data for subsequent use.

Fig. 3: Dataset Loading.

Figure 4 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.


Fig. 4: Word Cloud Visualization.

Figure 5 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.

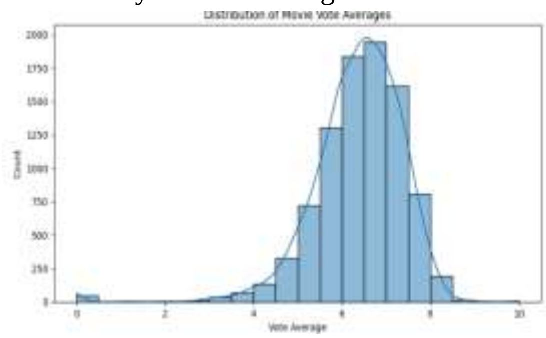


Fig. 5: Distribution of Movie Vote Averages.

Figure 6 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.

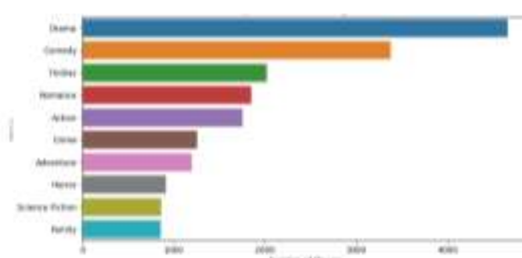

Fig. 6: Top 10 Movie Genres.

Figure 7 shows the performance of the spectral clustering algorithm, executed via the runSpectral function. The output includes a dataframe (result1) with 9557 rows and 249 columns, combining movie ratings (e.g., columns labeled 1, 4, 6, ..., 610) and a Cluster ID column (all 0s in the sample). The ratings are sparse, with values like 4.0, 4.5, or 0.0, indicating user-movie interactions. Performance metrics are calculated for a subset of 50 rows: Precision (65.97222222222222%), Recall (70.17543859649122%), and F1-Score (66.66666666666666%). These metrics, computed using Scikit-learn's precision_score, recall_score, and f1_score, reflect the clustering's ability to group movies based on ratings ≥ 4.0 as positive, displayed in the GUI text area.

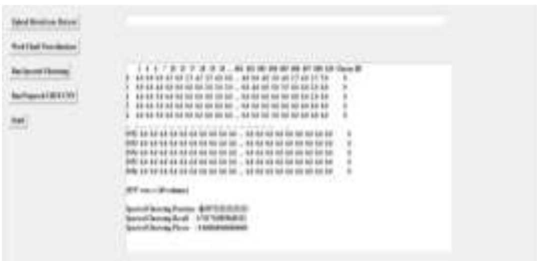

Fig. 7: Existing Spectral Clustering Performance.

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.

Fig. 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.

Fig. 9: Prediction From Test Input Movie Name.

Table 1: Comparison Table

| Metric | Existing Spectral Clustering | Proposed CBCF-CNN |
|---|---|---|
| **Accuracy** | - | 99.84309623430963% |
| **Precision** | 65.97222222222222% | 99.2675260603167% |
| **Recall** | 70.17543859649122% | 99.5889247970654% |
| **F1-Score** | 66.66666666666666% | 99.42760374931518% |

The comparison table contrasts the performance of spectral clustering and the proposed CBCF-CNN model (Figure 7) based on key metrics. Spectral clustering achieves a precision of 65.97222222222222%, recall of 70.17543859649122%, and F1-score of 66.66666666666666%, indicating moderate performance in grouping movies based on ratings. Notably, accuracy is not provided for spectral clustering in the output, suggesting it may not be a primary focus or was not computed. In contrast, the CBCF-CNN model demonstrates significantly higher performance, with an accuracy of 99.84309623430963%, precision of 99.2675260603167%, recall of 99.5889247970654%, and F1-score of 99.42760374931518%. These near-perfect metrics highlight the CBCF-CNN's superior ability to cluster movies accurately, leveraging its deep learning architecture (convolutional and LSTM layers) to capture complex patterns in the data. The table underscores the proposed model's effectiveness over traditional spectral clustering for movie recommendation tasks.

## 5. CONCLUSION AND FUTURE SCOPE

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]. Reddy, S. K. R. Tailoring Loyalty Rewards Systems across Industries: Cloud vs On-Prem Solutions. International Journal of All Research Education and Scientific Methods (IJARESM),April 2025, ISSN: 2455-6211.

[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]. Charan Nandigama, N. (2020). An Integrated Data Engineering and Data Science Architecture for Scalable Analytical Warehousing and Real-Time Decision Systems. International Journal of Business Analytics and Research (IJBAR).

[9]. 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.

[10]. 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.

[11]. 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.

[12]. 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}.

[13]. Charan Nandigama, N. (2024). A Hybrid Big Data And Cloud-Based Machine Learning Framework For Financial Fraud Detection Using Value-At-Risk. International Journal of Research and Analytical Reviews, 11(3). https://doi.org/10.56975/ijrar.v11i3.324899.

[14]. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007). CF Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_9.

[15]. Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A CF approach to mitigate the new user cold start problem. Knowledge-based systems, 26, 225-238.

[16]. Bobadilla, Jesús, Fernando Ortega, and Antonio Hernando. "A CF similarity measure based on singularities." Information Processing & Management 48.2 (2012): 204-217.

[17]. R. Ji, Y. Tian, and M. Ma, "CF Recommendation Algorithm Based on User Characteristics," 2020 5th International Conference on Control, Robotics and Cybernetics (CRC), 2020, pp. 56-60, doi: 10.1109/CRC51253.2020.9253466.

[18]. Pazzani, M.J., Billsus, D. (2007), P., Kobsa, A., Nejdl, W. (eds) Content-Based Recommendation Systems. In: Brusilovsky, The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_10.

[19]. Kulkarni, Saurabh, and Sunil F. Rodd. "Context Aware Recommendation Systems: A review of the state-of-the-art

techniques." Computer Science Review 37 (2020): 100255.

[20]. Harper, F.; Konstan, J. The MovieLens Datasets. ACM Trans. Interact. Intell. Syst. 2015, 5, 1–19.