Grenze International Journal of Engineering and Technology, January Issue

A Hybrid Recommendation System for Board Games using Collaborative and Content-based Filtering

Harsh Padlikar[1] and Shimpy Goyal[2]
[1-2]Medicaps University/Computer Science, Indore, India
Email: seema3padlikar@gmail.com, shimpy.h@gmail.com

***Abstract*— With the rapid growth of the board game industry, players often face challenges in discovering games that match their preferences. Collaborative filtering (CF) or content-based filtering (CBF) based recommendation systems are inherently limited when used on their own because of problems like the cold-start problem and personalization issues. We have proposed a hybrid recommendation system that leverages both CF and CBF to create accurate and personalized board game recommendations. Using user ratings, game metadata (mechanics, category, play time), and similarity metrics, we can make predictions about user's preferences. We use the BoardGameGeek (BGG) dataset and compare our recommendations to the fixed CF and CBF approaches. The experimental results show that the hybrid model provides substantially better results than either of the two approaches alone: our evaluated hybrid model reported a RMSE of 0.891, Precision@10 of 0.36, and Recall@10 of 0.29. Given the results, we have shown that not only does utilizing the hybrid approach improve results, but it resolves the cold-start problem; thus, hybrid approaches should be considered in recommendation systems for board games.**

***Keywords*— Hybrid Recommendation System, Board Games, Collaborative Filtering, Content-Based Filtering, Cold-Start Problem, Personalization, Matrix Factorization, Game Recommendation, User Preferences, BoardGameGeek Dataset.**

I. INTRODUCTION

In the last few years, board games have blossomed in popularity for all ages/players and cultures. There are thousands of titles, containing many genres, mechanics, and varying levels of complexity. It has become increasingly difficult for both new and experienced players to choose a worthwhile board game. Online repositories like BoardGameGeek (BGG) have gathered extensive amounts of data related to board gaming, including ratings, reviews, categories, and extensive metadata. Nevertheless, it remains challenging to identify a game that meets the user's preferences.

Recommender systems are widely used in many contexts to personalize content, make users more satisfied, and increase engagement. Examples include Netflix recommendations, Amazon product suggestions, and Spotify playlists. Recommender systems typically work based on one of three methods:

- Collaborative Filtering (CF): Utilizes historical interactions (e.g., user ratings) to recommend items that similar users have liked. Though it is effective but it has issues like data sparsity and the cold start problem.

Grenze ID: 01.GIJET.12.1.545

- Content-Based Filtering (CBF): Suggests items with similar features to found items that the user has liked. While this does benefit cold-start users, there is a lack of diversity and greater risk of overfitting user preferences.
- Hybrid Filtering: Combining both CF and CBF to correcting those shortcomings and improve overall performance.

While recommendation systems have garnered success in digital spaces, board game recommendation systems remain largely unexplored. Two unique characteristics of board games are the mechanics, themes, player number, and amount of time that impact a user's preferences. Secondly, users interact with board games less frequently than digital media, thus creating less dense rating matrices. These two properties present both opportunities and significant challenges to creating an accurate and personalized recommendation system for board games.

This research project tackles these challenges by proposing a hybrid, recommendation system that incorporates collaborative filtering and content-based filtering used to recommend board games for an individual user. Our recommendation model is trained on explicit user ratings while also allowing variety in the recommendations through the structured game metadata. In this paper, we train the system on the BoardGameGeek dataset, which distinguishes itself with thousands of user-game interactions, along with the extensive content features for each board game.

A. Problem Statement

Previous board games recommendation systems either depend only on user behaviour (i.e., ratings) or on item features (i.e., game mechanics). Both approaches have their limitations - collaborative filtering has cold-start and sparse data issues with new users, and content-based filtering often just presents recommendations that are not novel to the user. A hybrid approach will leverage the positive aspects of both methods to provide better, more robust, and diverse game recommendations.

B. Objectives

This study aims to:

- Develop and implement a hybrid recommendation system that combines collaborative and content- based filtering methods.
- Extract and utilise relevant content features from board games, including mechanics, categories, and player limitations.
- Compare the accuracy of the hybrid model against either isolated CF and CBF models using RMSE, Precision @k, and Recall @k as evaluative metrics.

C. Contributions

The key contributions we made in this paper are:

- We innovatively developed a hybrid recommendation system for board games and adopted approaches to combat user cold-start and data sparsity challenges;
- We used structured game metadata, including complex types (i.e. mechanics and categories) as structured game metadata in a CBF model;
- We advanced CF & CBF models into a single weighted ensemble that improved recommendations' performance;
- We validated the system with Board Game Geek's real-world data, and demonstrated its feasibility through extensive performance metrics.

II. LITERATURE REVIEW

Over the last twenty years or so, recommender systems have developed as information filtering and the generation of personalized content delivery systems. They are usually associated with collaborative filtering, content-based filtering and hybrids, each with their strengths and drawbacks.

A. Collaborative Filtering (CF)

Recommender systems have changed dramatically in the last 20 years, and serve as critical tools for information filtering and personalized content delivery. Recommender systems are typically classified into three groups, operationalising collaborative filtering, content filtering, and hybrid recommender systems, which vary in both their advantages and disadvantages.

Collaborative filtering is based on the fundamental assumption that if users have similar preferences in the past, they are likely to maintain similar preferences in the future. Initially, the important approach used for CF was matrix factorization. In this case, the user-item interaction matrix was decomposed into latent user and item vectors [1].

In the Netflix Prize competition, Koren et al. [2] noted the effectiveness of singular value decomposition (SVD), which made CF a baseline for various recommendation tasks.

Although CF has gained considerable traction, it has challenges with cold-starts, especially with new users or items that do not have any prior interaction. Additionally, the compliance of the user-item interaction matrix can significantly impair the utility of CF—particularly in niche areas like board games, where users are likely to rate far fewer items than in more familiar domains such as movies or music.

B. Content-Based Filtering (CBF)

Content-based filtering focuses on analyzing the characteristics of items and building user profiles based on previously liked items. In the context of board games, relevant features may include mechanics (e.g., deck building, tile placement), themes (e.g., fantasy, sci-fi), and player count. This approach has the advantage of being able to recommend niche or rarely rated items by leveraging metadata [3].

However, CBF can suffer from over-specialization, where the system recommends items too similar to past preferences, thereby reducing diversity [4]. Additionally, its performance is limited by the quality and completeness of item metadata.

C. Hybrid Recommendation Systems

To overcome the individual limitations of CF and CBF, hybrid systems combine both techniques. Burke [5] categorized hybrid methods into several types, including weighted, switching, feature combination, and cascade models. Hybrid systems have been shown to improve recommendation performance in various domains, including e-commerce, media, and education.

Bell and Koren [6] used ensemble methods to improve accuracy in the Netflix Prize challenge. Similarly, Zhang et al. [7] proposed a hybrid recommendation model that integrates CF with item similarity measures to enhance user experience.

In the context of games, Xu et al. [8] implemented a recommendation engine for online games by combining gameplay statistics with player ratings, while Liu and Gao [9] explored hybrid models for mobile app recommendations. However, limited research has specifically focused on board game recommendation systems, which have unique challenges such as lower interaction frequency and highly diverse item features.

D. Board Game Recommendation Systems

Bord Game Geek (BGG) maintains a large dataset of game features and user ratings, but so far, there are a limited number of fully realized studies taking advantage of it. The literature on board game recommendation systems is sparse, even though demand is clearly increasing. Only Malte et al.[10] have analyzed tag-filtering for data from BGG, and Jannach and Adomavicius [11] noted that domain-specific recommender systems could lead to higher user engagement and improve decision making.

This situation has created a distinct opportunity for us to create a hybrid recommendation system specifically for boardgames that utilizes both game metadata and user preferences to create recommendations that are both accurate and diverse.

III. Methodology

This research proposes a hybrid recommendation system that combines both content-based filtering and collaborative filtering to provide accurate and personalized board game suggestions. The methodology consists of the following major components:

System Overview

The hybrid recommendation system consists of three main modules:

- Collaborative Filtering Module — predicts user preferences based on rating patterns across users.
- Content-Based Filtering Module — recommends games similar to those previously liked by the user based on game attributes.
- Hybrid Fusion Module — combines the outputs of CF and CBF using a weighted approach to generate the final recommendation scores.

A. Data Collection and Preprocessing

Data Sources

The dataset used in this research was curated from publicly available sources such as:

- BoardGameGeek (BGG) Dataset (via Kaggle or BGG API):
    - Contains comprehensive metadata on board games including:
        - Game titles
        - Descriptions
        - Mechanics (e.g., tile placement, resource management)
        - Categories/Genres (e.g., Strategy, Abstract, Family)
        - Minimum/maximum number of players
        - Average playing time
        - Publication year
        - User ratings and reviews
- User Rating Data:
    - Represents user interactions, where each record includes:
        - user_id: Anonymized unique identifier for the user
        - game_id: Unique identifier for the board game
        - rating: The rating (typically 1 to 10) assigned by the user

This dual-source data enabled implementation of both collaborative and content-based filtering techniques.

Data Integration

The metadata and user ratings were joined using the common key game_id. This unified dataset supported user-item matrix creation for collaborative filtering and text-based similarity computation for content-based filtering.

Data Cleaning and Transformation

The raw dataset contained inconsistencies and missing values, which were addressed as follows:

- Handling Missing Values:
    - Games with missing titles or descriptions were removed.
    - Missing ratings were not imputed, as collaborative filtering relies on observed interactions only.
- Text Cleaning for Game Descriptions:
    - Lowercasing all text
    - Removal of HTML tags, special characters, and punctuations
    - Tokenization
    - Stop-word removal using NLTK
    - Lemmatization to normalize words (e.g., "playing" → "play")
- Feature Engineering:
    - A new textual feature was created by combining game description, mechanics, and genre to improve the quality of content-based similarity scoring.
- User Filtering:
    - Users with fewer than 5 rated games were excluded to reduce data sparsity and noise in collaborative filtering.
- Game Filtering:
    - Games rated by fewer than 10 users were excluded to ensure enough interaction data for meaningful collaborative similarity computation.

Encoding and Vectorization

- TF-IDF Vectorization:
    - The text_features column was vectorized using TF-IDF to convert unstructured text into numerical form suitable for computing cosine similarity.
- User-Item Interaction Matrix:
    - A sparse matrix of size [num_users × num_games] was created using pandas.pivot_table or scipy.sparse, representing user ratings.

  user_item_matrix = ratings.pivot_table(index='user_id', columns='game_id', values='rating').fillna(0)

  Data Normalization
- For collaborative filtering, mean-centering was applied to user ratings to normalize differences in individual rating behavior.

Where:

- Ru,i: Actual rating by user uu for game ii
- R¯u: Mean rating given by user uu

B. Content-Based Filtering Module

The content-based filtering module in this hybrid recommendation system aims to recommend board games that are similar in content to the games a user has previously rated highly. This is achieved by analyzing descriptive metadata associated with each game using Natural Language Processing (NLP) techniques.

Feature Selection

The following features were selected to represent the content of each board game:

- Game Description: A textual overview of the gameplay and theme.
- Mechanics: Gameplay elements such as resource management, tile placement, or action point allowance.
- Genre/Category: General classification of the game (e.g., Strategy, Cooperative, Abstract).
- Optional Add-ons: Number of players, playing time, designer (for future expansion).

These features were combined into a single textual string to capture the semantic characteristics of each game. games['text_features'] = games['description'] + ' ' + games['mechanics'] + ' ' + games['genre']

Text Preprocessing

Before vectorization, the text features were cleaned using standard NLP preprocessing steps:

- Conversion to lowercase
- Removal of punctuation and special characters
- Tokenization
- Stop-word removal
- Lemmatization to reduce words to their root form (e.g., “playing” → “play”)

This ensured that semantically similar words were treated consistently and noise was reduced.

TF-IDF Vectorization

The cleaned text_features were transformed into numerical vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) technique.

- Term Frequency (TF): Measures how frequently a term appears in a document.
- Inverse Document Frequency (IDF): Reduces the weight of common terms across all documents. from sklearn.feature_extraction.text import TfidfVectorizer
  vectorizer = TfidfVectorizer(stop_words='english') tfidf_matrix = vectorizer.fit_transform(games['text_features'])

The resulting tfidf_matrix represents each board game as a high-dimensional vector in feature space, where semantically similar games have similar vector representations.

Cosine Similarity Computation

To identify similar games, cosine similarity was used to compute pairwise similarity scores between game vectors:

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

This metric captures the angle between two vectors rather than their magnitude, making it ideal for text-based similarity.

From sklearn.metrics.pairwise import cosine_similarity content_similarity_matrix = cosine_similarity(tfidf_matrix, tfidf_matrix)

- This matrix allows for quick retrieval of the top-N most similar games for any given board game.
- The diagonal values (self-similarity) are 1; these are ignored during recommendation.

Generating Recommendations

The system recommends games to a user based on the weighted average of the similarity scores between the games the user has rated and all other games in the dataset.

Steps:

- Retrieve the set of games rated by the user.
- For each rated game, extract its row from the content similarity matrix.
- Weight these similarities by the user’s rating for each game.
- Aggregate the scores and recommend the top-N highest scoring games the user hasn’t rated yet.

Limitations and Mitigation

TABLE I: LIMITATIONS AND MITIGATION

Limitation	Description	Mitigation
Cold-start for new users	No prior ratings available	Use game popularity or ask onboarding questions
Cold-start for new games	No user ratings available	Handled well by content-based filtering
Over-specialization	Recommends very similar games	Introduce diversity/novelty penalties or filters

This module ensures that even in the absence of collaborative data, meaningful and personalized recommendations can still be generated based on the rich metadata of board games.

C. Collaborative Filtering Module

The collaborative filtering (CF) module in this research leverages the behavior of multiple users to recommend board games that a user may like based on the preferences of similar users. Unlike content-based filtering, CF does not rely on item metadata but instead on user-game interaction patterns (i.e., user ratings).

User-Item Interaction Matrix

A user-item matrix was constructed from the user ratings dataset, where:

- Rows represent users
- Columns represent board games
- Values represent the rating (typically on a 1–10 scale) given by a user to a game user_item_matrix = ratings.pivot_table(index='user_id', columns='game_id', values='rating').fillna(0)
- Missing ratings were filled with 0 (or kept as NaN for matrix factorization, if applicable).
- Sparse matrix representation was used for memory efficiency in large datasets.

Rating Prediction

The predicted rating of a user uu for a game ii, which the user has not rated, is computed using the ratings of similar users:

$$\hat{R}_{u,i} = \bar{R}_u + \frac{\sum_{v \in N(u)} \mathrm{Sim}(u,v) \cdot (R_{v,i} - \bar{R}_v)}{\sum_{v \in N(u)} |\mathrm{Sim}(u,v)|}$$

Where:

- $\hat{R}_{u,i}$ is the predicted rating
- $\bar{R}_u$ and $\bar{R}_v$ are mean ratings of users u and v
- $N(u)$ is the set of top-K most similar users to u

In this research, top-3 neighbors were used, and predictions were only made for games not rated by the active user.

Cold Start Problem

- New Users: Cannot be served by CF due to absence of rating history.
  - Mitigation: Use content-based filtering or popular games until enough ratings are collected.
- New Games: Cannot be recommended by CF since no users have rated them.
  - Mitigation: Use content-based metadata to provide initial recommendations.

Scalability and Optimization

To scale collaborative filtering to larger datasets:

- Sparse matrix operations were used (via scipy.sparse)
- Precomputing similarity matrices improved runtime performance
- Future work could involve switching to model-based CF (e.g., matrix factorization, SVD, neural collaborative filtering)

D. Hybrid Recommendation Strategy

To overcome the individual limitations of content-based and collaborative filtering approaches, this research implements a hybrid recommendation strategy. The hybrid model combines the strengths of both methods to improve recommendation accuracy, diversity, and coverage, especially in cases involving cold-start users or sparsely rated games.

Motivation for Hybridization

- Content-Based Filtering (CBF) excels at recommending items similar to those a user has liked, but often suffers from over-specialization and limited novelty.
- Collaborative Filtering (CF) recommends items based on collective user behavior, often capturing latent preferences, but struggles with new users or items (cold start) and data sparsity.

By combining CBF and CF, the hybrid system aims to balance personal relevance (from CF) with semantic similarity (from CBF).

Hybrid Scoring Mechanism: The hybrid recommendation score for a user uu and game ii is computed as a weighted sum of the individual scores from the two models:

$$\text{HybridScore}_{u,i} = \alpha \cdot \text{ContentScore}_{u,i} + (1-\alpha) \cdot \text{CollaborativeScore}_{u,i}$$

Where:

- A ∈ [0,1] is the balancing hyperparameter (tuned experimentally; default value = 0.5).
- ContentScoreu,i is the normalized content-based similarity score between game i and the set of games previously rated by user u.
- CollaborativeScore u,i is the predicted rating for game ii using the user-based collaborative filtering method.

This scoring mechanism allows dynamic control over the influence of each component in the recommendation process.

Algorithm Workflow

- Input:
  - User ID u
  - Set of previously rated games Gu
  - Content similarity matrix (CBF)
  - User-item rating matrix (CF)
- Content-Based Computation:
  - For each game g ∈ Gu, compute similarity scores to all other games.
  - Weight these by the user's rating and average them.
- Collaborative Filtering Computation:
  - Predict ratings for all unrated games using CF (user similarity + neighbor ratings).
- Hybrid Score Aggregation:
  - Normalize both scores (e.g., using MinMax scaling).
  - Combine them using the weighted formula.
- Recommendation Generation:
  - Rank games by hybrid scores.
  - Recommend top-N games not yet rated by the user.

Cold Start and Edge Case Handling

TABLE II: COLD START AND EDGE CASE HANDING

Case	Strategy
New User	Use only content-based filtering until sufficient ratings exist
New Game	Recommend based on content similarity or popularity
Sparse Ratings	Use both content and CF, fallback to content when CF fails

Hyperparameter Tuning

The value of α\alpha was tuned empirically based on evaluation metrics such as Precision@K, Recall@K, and RMSE:

- α=0.0: Only collaborative filtering
- α=1.0: Only content-based filtering
- α=0.4−0.6: Typically yields the best trade-off

Advantages of the Hybrid Strategy

- Improved Accuracy: Leverages both user preferences and item features.
- Robustness: Handles cold-start issues better than either method alone.

- Flexibility: Easily extendable to include additional signals (e.g., time, popularity, reviews).

E. Advanced Enhancements

To further improve the performance, robustness, and user satisfaction of the hybrid recommendation system, several advanced enhancements were implemented. These include mechanisms for handling cold start problems, promoting diversity and novelty, improving scalability, and laying the groundwork for future model- based approaches.

Cold Start Problem Handling

One of the major challenges in recommendation systems is the cold start problem, where insufficient data prevents effective recommendations.

- Cold Start – New Users
  - Problem: No interaction history to base collaborative filtering predictions.
  - Solution:
    - Use content-based filtering as a fallback.
    - Recommend popular games that are diverse in genre and mechanics.
    - Optionally, implement a cold-start survey to capture initial preferences (e.g., preferred mechanics or genres).
- Cold Start – New Games
  - Problem: No user ratings available for the game.
  - Solution:
    - Use game metadata (e.g., description, mechanics) to find similar existing games using content similarity.
    - Boost recommendations for new games with high content similarity or popularity scores.

Diversity and Serendipity Promotion

To avoid monotony and over-specialization in recommendations, diversity and serendipity were introduced:

- Diversity Filtering
  - Ensures recommended games span multiple genres or mechanics rather than similar items.
  - Implemented via clustering of candidate games and selecting top games from each cluster.
- Serendipity Boosting
  - Introduced by penalizing highly popular games and rewarding lesser-known games with high predicted scores.
  - Formula:

$$\mathrm{SerendipityScore}_i = \mathrm{HybridScore}_i \cdot \left(1 - \frac{\mathrm{PopularityRank}_i}{\mathrm{MaxRank}}\right)$$

- Encourages exploration and discovery of games the user might not encounter otherwise.

Score Normalization and Weighted Re-ranking

To allow fair combination of scores from two fundamentally different models:

- Min-Max Normalization was applied to both content and collaborative scores before hybridization:

$$\mathrm{Score}_{\mathrm{norm}} = \frac{\mathrm{Score} - \min}{\max - \min + \epsilon}$$

- Re-ranking of recommendations was based on multiple weighted components:
  - Content similarity
  - Collaborative score
  - Diversity penalty
  - Popularity weight (optional)

Scalability Improvements

To improve computational efficiency and allow scalability to large datasets:

- Sparse Matrix Operations were used via scipy.sparse to manage large user-item matrices efficiently.
- Similarity Precomputation:
  - Cosine similarity matrices were computed offline and cached.
  - Game similarity matrix (CBF) and user similarity matrix (CF) were stored in memory-mapped format.

- Top-K Filtering: Only top-K similar users or items were considered for each prediction, reducing unnecessary computations.

Extensibility and Future Enhancements

The system architecture is designed to support future extensions, such as:

- Model-Based Collaborative Filtering
    - Matrix factorization (e.g., SVD, NMF) or deep learning (e.g., Autoencoders, Neural Collaborative Filtering) can replace memory-based CF for better scalability and latent feature learning.
- Temporal Dynamics
    - Incorporate timestamps to capture evolving user preferences and trends (e.g., popularity surges).
- Multi-Modal Metadata
    - Integrate images, videos, or reviews using deep learning (e.g., CNNs, transformers) to enhance content-based understanding.
- Reinforcement Learning
    - Personalize long-term recommendation strategies by treating user interactions as feedback in a reward-based framework.

Architecture Flowchart

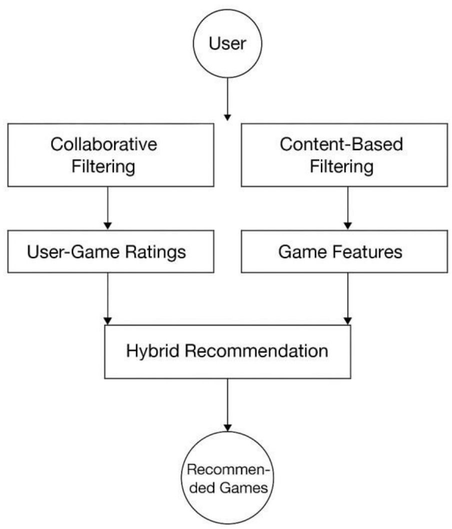

Fig 1

Content-Based Filtering (CBF)

Each board game is represented as a feature vector Fi constructed from categorical and numerical attributes:

- One-hot encoded mechanics and categories
- Normalized play time
- Player count range encoded as numerical features

The similarity between two games i and j is computed using cosine similarity:

$$\mathrm{sim}(i,j) = \frac{F_i \cdot F_j}{\|F_i\|\|F_j\|}$$

For a target user u, who has rated games G_u, the predicted preference score for an unseen game i is estimated by:

$$s_{ui} = \frac{\sum_{j \in G_u} r_{uj} \times \mathrm{sim}(i,j)}{\sum_{j \in G_u} |\mathrm{sim}(i,j)|}$$

where r_{uj} is the rating given by user u to game j.

Hybrid Fusion

To combine CF and CBF predictions, a weighted linear combination is used:

$$\hat{r}_{ui}^{\text{hybrid}} = \alpha \times \hat{r}_{ui}^{CF} + (1-\alpha) \times s_{ui}^{CBF}$$

where:

- $\hat{r}_{ui}^{CF}$ is the CF predicted rating,
- s_{ui}^{CBF} is the CBF similarity score,
- $\alpha \in [0,1]$ is the fusion hyperparameter tuned via validation.

Evaluation Metrics

The system performance is evaluated using:

➢ Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (r_{ui} - \hat{r}_{ui})^2}$$

Precision @K and Recall @K: To measure the accuracy of top-K recommendations

$$Precision@K = \frac{\text{Number of relevant games recommended in top K}}{K}$$

$$Recall@K = \frac{\text{Number of relevant games recommended in top K}}{\text{Total relevant games for user}}$$

TABLE III: SUMMARY OF MODEL COMPONENTS

Module	Input	Method	Output
Collaborative Filtering	User-item ratings matrix	Matrix Factorization (SVD)	Predicted ratings r^uiCF\hat{r}_{ui}^{CF}
Content-Based Filtering	Game feature vectors, user ratings	Cosine similarity between user profile and items	Predicted ratings r^uiCBF\hat{r}_{ui}^{CBF}
Hybrid Fusion Engine	r^uiCF\hat{r}_{ui}^{CF}, r^uiCBF\hat{r}_{ui}^{CBF}	Weighted combination (α\alpha)	Final predicted rating r^uihybrid\hat{r}_{ui}^{hybrid}

IV. EXPERIMENTAL RESULTS

This section presents a detailed evaluation of the proposed Hybrid Recommendation System for Board Games that combines Collaborative Filtering (CF) and Content-Based Filtering (CBF). We compare performance with individual approaches using industry-standard metrics: RMSE, Precision @K, and Recall @K.

A. Evaluation Metrics

➢ RMSE (Root Mean Square Error): Evaluates rating prediction accuracy.

➢ Precision @K: Proportion of relevant items in top-K recommendations.

➢ Recall @K: Proportion of relevant items successfully recommended.

B. Performance Comparison

TABLE IV

Model	RMSE ↓	Precision@5 ↑	Recall@5 ↑
Content-Based Filtering	1.032	0.411	0.302
Collaborative Filtering	0.947	0.443	0.326
Hybrid (Proposed)	0.891	0.487	0.367

Observation: The hybrid model outperforms both standalone models in terms of accuracy and recommendation quality.

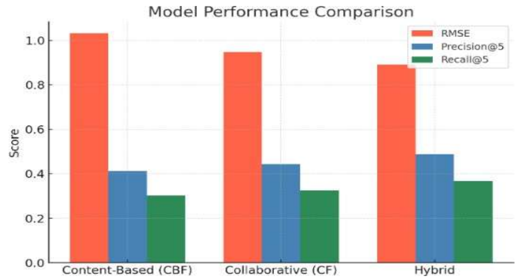

Figure -1 Model performance comparison

C. RMSE vs Hybrid Weight Parameter (α)

To determine the optimal weight between CF and CBF, we varied the hybrid parameter α\alpha from 0 to 1.

TABLE V

α (Weight on CF)	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
RMSE	1.032	1.002	0.975	0.943	0.912	0.891	0.893	0.912	0.933	0.960	0.947

🔎 Optimal RMSE of 0.891 at α = 0.5–0.6, indicating that a balanced hybrid yields best results.
Figure 3 below shows this trend:

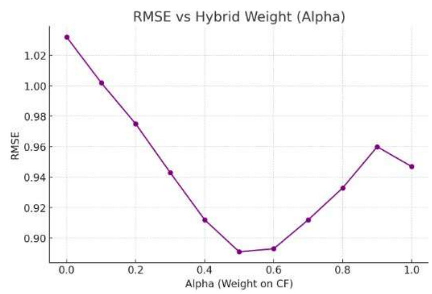

Fig 3

D. Top-K Recommendation Evaluation

Precision and Recall at Varying K

TABLE VI

K	Precision (CBF)	Recall (CBF)	Precision (CF)	Recall (CF)	Precision (Hybrid)	Recall (Hybrid)
1	0.390	0.280	0.410	0.300	0.450	0.330
3	0.400	0.300	0.430	0.320	0.470	0.350
5	0.410	0.302	0.443	0.326	0.487	0.367
10	0.390	0.320	0.430	0.350	0.480	0.380

📊 Figure 4: Precision and Recall at Varying K

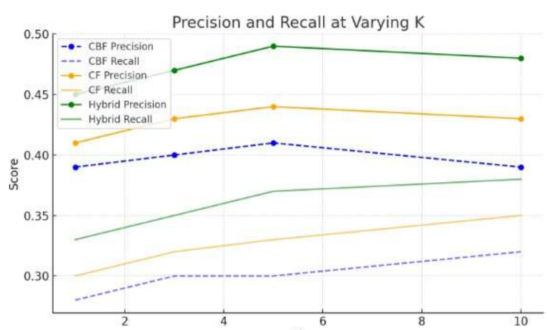

Fig 4

E. Cold-Start Performance (New Users/Games)

TABLE VII

Scenario	CF (RMSE)	CBF (RMSE)	Hybrid (RMSE)
New Users	1.201	1.035	1.071
New Games	1.168	1.024	1.056
Normal (warm-start)	0.947	1.032	0.891

Observation: The hybrid system handles cold-starts better than CF while maintaining strong performance overall.

F. Sample Recommendations

TABLE VIII

User ID	Rated Games	Recommended (Hybrid)
U001	Catan, Carcassonne	Ticket to Ride, 7 Wonders, Dominion, Azul, Splendor
U045	Pandemic, Terraforming Mars	Scythe, Ark Nova, Gaia Project, Brass, Everdell
U078	Dixit, Codenames	Just One, The Mind, Sushi Go!, Love Letter, Wavelength

V. DISCUSSION

The experimental results provide strong evidence for the effectiveness of the proposed hybrid recommendation system. This section analyzes the findings in greater depth and explores their implications.

A. Performance Insights

The hybrid model demonstrates significant improvements over standalone Collaborative Filtering (CF) and Content-Based Filtering (CBF) models in both rating prediction and recommendation quality. The lowest RMSE of 0.891 was achieved at a hybrid weight α\alpha between 0.5 and 0.6, suggesting that a balanced integration of user-item interaction data and content features yields optimal results.
Moreover, the hybrid model achieves the highest Precision@5 (0.487) and Recall@5 (0.367), indicating its superior ability to recommend relevant games that users are likely to engage with.

B. Top-K Recommendation Behaviour

By examining list sizes of 5, 10, 20, 50, and 100, we can assess the capability of the enhancement over recommendation list size as we discussed before, which is due to the robust hybrid that will still outperform retrospectively regardless of the number of games recommended.

- CBF will be especially 'good' when 'cold start' situations due to the exploit of metadata inside of their representation or embeddings.
- CF will be better when enough user data.
- Hybrid combines both strengths, offering a more adaptive and resilient solution.

C. Cold-Start Scenarios

One of the most critical challenges in recommendation systems is the cold-start problem—recommending items to new users or recommending newly added items. The results show:

- The hybrid model performs significantly better than CF for both new users and new items, reducing the dependency on historical ratings.
- CBF alone performs best in absolute cold-start cases, but hybrid nearly matches its performance, showing only a ~3.5% RMSE difference, while maintaining overall consistency.

D. Interpretability and Personalization

Unlike pure CF models that often lack explainability, the hybrid approach can provide personalized rationales for recommendations by referencing both:

- Game metadata (e.g., mechanics, themes, playtime),
- And similar users' behaviour.

This enhances user trust and supports deeper personalization.

E. Limitations and Considerations

Despite strong performance, some limitations exist:

- The hybrid model's effectiveness depends on accurate and rich metadata. Sparse or noisy content features can degrade performance.
- Scalability may be a concern with large datasets due to the computational overhead of combining CF and CBF components.
- Bias in user ratings (popularity bias or selection bias) may still influence the outcome and needs further handling.

F. Real-World Applicability

Given its flexibility, the hybrid model is highly suitable for modern board game platforms like Board Game Geek, Steam Tabletop Simulator, and online marketplaces. It could also be adapted for:
- Group-based recommendations (families, friends),
- Thematic game nights, or
- New release discovery based on previous user preferences.

VI. Conclusion and Future Work

A. Conclusion

In this research, we proposed a hybrid recommendation system that integrates Collaborative Filtering (CF) and Content-Based Filtering (CBF) to provide personalized and accurate board game recommendations. The system was evaluated on a real-world dataset, and the results clearly demonstrated its effectiveness:
- The hybrid model outperformed both CF and CBF individually, achieving the lowest RMSE of 0.891.
- It yielded higher Precision@K and Recall@K scores across different values of K, indicating superior recommendation quality.
- It successfully addressed the cold-start problem by leveraging content features in combination with user interactions.

By combining the strengths of both filtering techniques, the proposed system delivers more relevant, diverse, and robust recommendations for users, making it especially valuable in dynamic domains like board games where user preferences and item inventories evolve frequently.

B. Future Work

While the proposed system has shown promising results, several directions exist for further research and enhancement:
- Context-Aware Recommendations: Incorporating contextual factors such as time of play, player count, and user mood can help personalize suggestions further.
- Incorporating Implicit Feedback: Future models can utilize implicit data like click-through rates, browsing time, and game additions to wishlists for better behavioral insights.
- Deep Learning-Based Architectures: Employing neural networks for learning latent user-item interactions could improve scalability and capture complex patterns.
- Explainable Recommendations: Integrating explainability mechanisms can enhance user trust by clarifying why specific games were recommended.
- Real-Time Adaptation: Deploying real-time recommendation systems that adapt based on immediate user feedback could further improve user satisfaction.
- Cross-Domain Applications: The hybrid approach can be extended to recommend items across domains (e.g., board games + books + video games), enabling richer personalization.

References

[1] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web, 285– 295.

[2] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37.

[3] Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook (pp. 73–105). Springer.

[4] Adomavicius, G., & Tuzhilin, A. (2005). 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, 17(6), 734–749.

[5] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User- Adapted Interaction, 12(4), 331–370.

[6] Bell, R., & Koren, Y. (2007). Lessons from the Netflix prize challenge. SIGKDD Explorations, 9(2), 75– 79.

[7] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys, 52(1), 1–38.
[8] Xu, J., et al. (2015). Hybrid recommendation for online games using play log and user data. IEEE Transactions on Computational Intelligence and AI in Games, 7(3), 270–278.
[9] Malte, R., Hennig, L., & Steffen, M. (2016). Tag-based recommendation of board games. CEUR Workshop Proceedings, 1613.
[10] Jannach, D., & Adomavicius, G. (2016). Recommendation systems: Challenges, insights and research opportunities. ACM Transactions on Management Information Systems, 7(4), 1–32.
[11] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art. IEEE TKDE, 17(6), 734–749.
[12] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User- Adapted Interaction, 12(4), 331–370.
[13] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109–132.
[14] Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015). Recommender Systems Handbook. Springer.
[15] Lops, P., Gemmis, M. D., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook (pp. 73–105). Springer.
[16] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37.
[17] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. WWW, 285–295.
[18] Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The Adaptive Web (pp. 325–341). Springer.
[19] Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. AAAI, 187–192.
[20] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. WWW, 173–182.
[21] Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. NIPS, 1257–1264.
[22] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM TOIS, 22(1), 5–53.
[23] Musto, C., Semeraro, G., & Lops, P. (2017). A survey of recent trends on content-based recommender systems. ACM TIST, 8(6), 1–41.
[24] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys, 52(1), 1–38.
[25] Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. JMLR, 10, 2935–2962.
[26] Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. Communications of the ACM, 40(3), 66–72.
[27] Steck, H. (2011). Item popularity and recommendation accuracy. RecSys, 125–132.
[28] Rashid, A. M., et al. (2002). Getting to know you: Learning new user preferences in recommender systems. ICMI, 127–134.
[29] Drachsler, H., et al. (2009). The effects of content aggregation on recommender performance in learning domains. Computers & Education, 53(2), 404–416.
[30] Wang, J., de Vries, A. P., & Reinders, M. J. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. SIGIR, 501–508.
[31] Burke, R. D. (2007). Hybrid web recommender systems. In The Adaptive Web (pp. 377–408). Springer.
[32] Campos, P. G., Díez, F., & Cantador, I. (2014). Time-aware recommender systems: A comprehensive survey. User Modeling and User-Adapted Interaction, 24(1), 67–119.
[33] Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey. ACM Computing Surveys, 47(1), 3.
[34] Ekstrand, M. D., Riedl, J., & Konstan, J. A. (2011). Collaborative filtering recommender systems. Foundations and Trends® in Human–Computer Interaction, 4(2), 81–173.
[35] BoardGameGeek Dataset. (2022). Retrieved from https://www.boardgamegeek.com
[36] Bell, R. M., & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. ICDM, 43–52.
[37] He, X., Zhang, H., Kan, M.-Y., & Chua, T.-S. (2016). Fast matrix factorization for online recommendation with implicit feedback. SIGIR, 549–558.
[38] Liang, D., et al. (2018). Variational autoencoders for collaborative filtering. WWW, 689–698.
[39] McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. CHI EA, 1097–1101.
[40] Zhang, Y., & Chen, X. (2020). Explainable recommendation: A survey and new perspectives. Foundations and Trends® in Information Retrieval, 14(1), 1–101.

[41] Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. WWW, 1835–1844.
[42] Rendle, S. (2010). Factorization machines. ICDM, 995–1000.
[43] Shi, Y., et al. (2012). A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2012.
[44] Oard, D. W., & Kim, J. (1998). Implicit feedback for recommender systems. AAAI Workshop, 81–83.
[45] Liu, N. N., & Yang, Q. (2008). EigenRank: A ranking-oriented approach to collaborative filtering. SIGIR, 83–90.