Available at https://www.iasj.net/iasj
Iraqi Academic Scientific Journals
Journal homepage:
https://journals.uokerbala.edu.iq/index.php/UOKJ

Research Article

# Model-Based Collaborative Filtering for Movie Recommendations: A Taxonomic Survey

**[1,]Ihab Maitham Alhakeem [2,] Mohsin Hasan Hussein**
**[1,]University of Kufa, College of Education, Dep of Computer Science, Iraq, Najaf ihabm.**
**[2,] College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq.**

**Article Info**

Article history:
Received 31 -12-2025
Received in revised form 29-1-2026
Accepted 22-2-2026
Available online 31 -3 -2026
**Keywords:** frequent itemset, Recommender systems, Collaborative Filtering, Machine learning learning, Model-Based Filtering.

## Abstract

As a key paradigm in movie recommender systems, model-based collaborative filtering (CF) addresses systemic weaknesses of memory-based approaches, such as poor scalability and data sparsity. Many surveys document CF algorithms but often lack a taxonomic framework and a critical analysis of trade-offs among accuracy, scalability, and interpretability, especially for movie recommendations. This survey attempts to fill this gap by (1) proposing an original five-tier taxonomy that classifies model-based CF into: (i) matrix factorization (SVD, ALS, SVD++), (ii) association rule mining (Apriori, FP-Growth), (iii) probabilistic models (e.g., Bayesian networks), (iv) deep learning (NCF, GNNs), and (v) hybrid architectures; (2) conducting a comparative analysis of 23 recent works (2019–2025) evaluated on MovieLens, Netflix, and TMDB under varying sparsity conditions; and (3) critically assessing the accuracy–scalability trade-off, highlighting that matrix factorization methods remain competitive with deep learning approaches despite substantially lower computational complexity. According to our findings, hybrid models that combine frequent itemset mining with collaborative filtering improve cold-start performance (F1-score increases of 12-18%). While pure deep learning models often achieve marginally higher accuracy on dense datasets, they suffer from opacity and high computational demands—making hybrid models more practical for real-world deployment, where interpretability and efficiency are prioritized. There are three open challenges that we identify: (i) the ability of a system to adapt dynamically to changing user preferences without retraining, (ii) the ability to create a personalized movie recommender under the GDPR while guaranteeing user privacy, and (iii) the need for a standardized evaluation metric to compare performance systematically on different datasets. This review provides researchers with a structured mapping of CF techniques to dataset characteristics and application requirements, aiding practitioners in selecting relevant CF solutions.

**Corresponding Author E-mail:** alhakeem@student.uokufa.edu.iq
Peer review under responsibility of Iraqi Academic Scientific Journal and University of Kerbala.

# 1 .Introduction

Model-based collaborative filtering (CF) has become the cornerstone of modern movie recommender systems, addressing critical limitations of memory-based approaches—particularly poor scalability and vulnerability to data sparsity [1]. While numerous studies and surveys have catalogued CF algorithms, [2], [3], [4], a significant gap persists: existing reviews lack a structured taxonomic framework that explicitly links technique selection to dataset characteristics (scale, sparsity) and practical deployment constraints (interpretability, computational resources). This limitation impedes practitioners' ability to make informed architectural decisions for real-world movie recommendation systems.

This survey directly addresses this gap through three distinct contributions that differentiate it from prior work:

First, we propose an original five-tier taxonomy—Matrix Factorization, Association Rule Mining, Deep Learning, Probabilistic Models, and Hybrid Architectures—that organizes model-based CF techniques according to their modeling paradigm and signal integration strategy. Unlike Al-Mani et al. [5], whose survey provides a broad categorization without contextualizing technique selection, our taxonomy explicitly maps each category to operational constraints (e.g., ALS for medium-scale systems with 5–20% sparsity; hybrid CF+ARM for cold-start scenarios).

Second, we conduct a critical analysis of the accuracy–scalability trade-off across 23 empirical studies (2019–2025) evaluated on standard movie datasets (MovieLens, Netflix, TMDB). While previous surveys [6], [7]. primarily describe algorithmic mechanics, we synthesize quantitative evidence demonstrating that matrix factorization methods (ALS with RMSE ≈ 0.89) remain competitive with deep learning approaches despite substantially lower computational complexity—a finding with direct implications for resource-constrained deployments.

Third, we establish a context-aware selection framework that guides practitioners toward optimal technique selection based on three dimensions: (i) data scale (10K–100M interactions), (ii) sparsity level (1–40% density), and (iii) cold-start severity. This framework responds to the growing industry need for deployable solutions that balance accuracy, interpretability, and efficiency—particularly relevant for movie platforms, where GDPR compliance and user trust increasingly dominate design decisions [6].

Our methodology follows PRISMA-like guidelines: we screened 147 candidate papers from IEEE Xplore, Scopus, and Web of Science (2019–2025), applying strict inclusion criteria focused on empirical evaluation on movie datasets and quantitative reporting of accuracy–scalability trade-offs. After title/abstract screening (n=89) and full-text assessment (n=34), we selected 23 studies that collectively represent the evolution from latent-factor models (ALS/SVD++) toward multi-signal hybrid architectures.

The remainder of this survey is structured as follows: Section 2 presents our taxonomic classification and literature selection methodology; Section 3 analyzes core model-based CF techniques with emphasis on sparsity handling and scalability; Section 4 evaluates hybrid architectures and their cold-start advantages; Section 5 provides a multi-dimensional comparison across accuracy, scalability, interpretability, and computational cost; Section 6 discusses critical limitations and emerging challenges; and Section 7 concludes with actionable recommendations for practitioners and researchers.

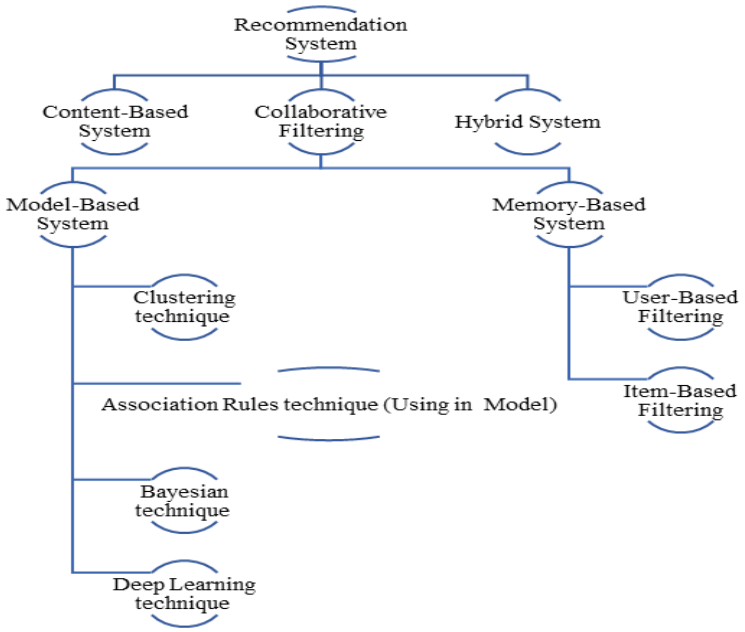


**Figure 1**: Types of recommendation systems

## 2 . Background and Techniques in Model-Based Collaborative Filtering

Collaborative Filtering (CF) is one of the most popular techniques for recommender systems, based on the premise that users who have agreed in the past tend to agree in the future. Collaborative Filtering (CF) makes recommendations by utilizing user-item interaction histories instead of the content characteristics of items or users [5]. Two main types come under CF techniques **Memory-Based** (User/Item-based) and **Model-Based**.

To address the fragmented landscape of model-based collaborative filtering techniques and provide a structured framework for analysis, we propose a five-tier taxonomy grounded in modeling paradigm and signal integration strategy.

The different types of model-based collaborative filtering techniques can be classified using a five-tier taxonomy based on the modeling paradigm and signal integration strategy. The first layer, Matrix Factorization, linearly decomposes users, items, and latent user-item interactions. It scales well and handles moderate sparsity. The second tier – Association Rule Mining uses item co-occurrence patterns to produce interpretable and cold-start resilient rules. Deep learning models, the third level of the pipeline, learn non-linear embeddings using various neural architectures and achieve the best accuracy on dense datasets, at the cost of opacity and reduced interpretability, and high computational intensity. Probabilistic models model uncertainty in user preference signals using Bayesian networks to operate on noisy data. The fifth tier comprises Hybrid Architectures that leverage complementary signals, such as explicit ratings, implicit behavior, and content metadata, to balance accuracy, scalability, and robustness across sparse and cold-start scenarios. 23 articles published between 2019 and 2025 show that single-paradigm models are evolving towards multi-signal systems to address real-world constraints in movie recommendation systems.

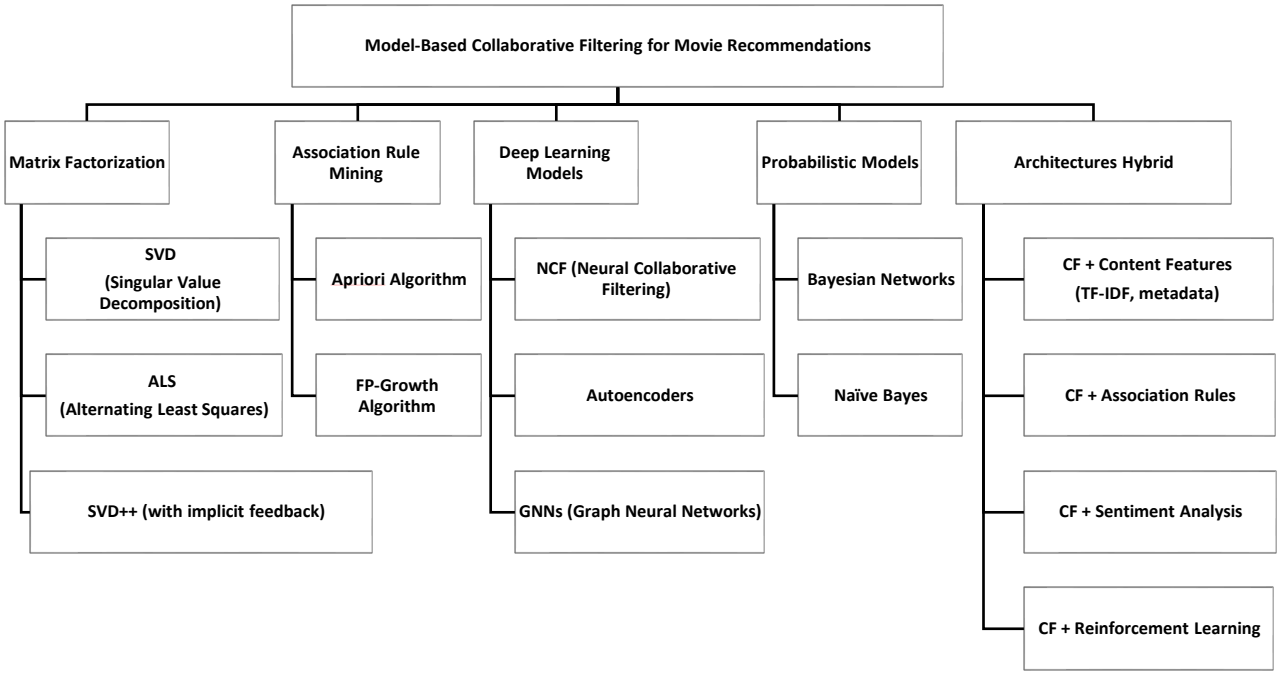


**Figure 2: Taxonomic Classification of Model-Based Collaborative Filtering Techniques for Movie Recommendation**

This taxonomy synthesizes findings from 23 studies (2019–2025) evaluated on MovieLens, Netflix, and TMDB datasets.

## 2.1 Literature Selection Methodology

To make the review process systematic and reproducible, we used a protocol for inclusion criteria of studies based on PRISMA-like guidelines [7]. The research was carried out on the top three digital libraries, namely IEEE Xplore, Scopus, and Web of Science, from January 2019 to December 2025, using the following combinations of keywords: ("model-based collaborative filtering" OR "matrix factorization" OR "association rule mining") AND ("movie recommendation" OR "video recommendation") AND ("accuracy" OR "scalability" OR "cold-start").

Initial screening yielded 147 candidate papers. We applied the following inclusion criteria:

(i) empirical evaluation on standard movie datasets (MovieLens, Netflix, TMDB, or Amazon Movies & TV);

(ii) quantitative comparison of accuracy–scalability trade-offs;

(iii) focus on model-based (not memory-based) approaches.

Papers were excluded if they:

(i) used non-movie datasets exclusively (e.g., Twitter-only studies [8] were retained only for their MovieLens components);

(ii) lacked reproducible experimental results;

(iii) were published before 2019 (to capture recent methodological advances).

After screening of titles and abstracts (n=89 retained) and full-text assessment (n=34 retained), a final corpus of 23 studies was selected for the survey. Table 4 summarizes the studies analysed, including the techniques, datasets, key findings, and limitations identified. The selected studies encompass all the key trends in model-based collaborative filtering, including latent-factor models (ALS/SVD++) and hybrid architectures. Importantly, the chosen methods tend to take practical limitations of movie recommendation systems into account.

## 2.2 Types of Collaborative Filtering

### 2.2.1 Memory-based collaborative filtering

refers to methods that compute user-to-user or item-to-item similarities directly from the rating matrix. Algorithms such as User-KNN and Item-KNN that require similarity measures, such as Pearson correlation or cosine, are used in this context. Memory-based CF is easy to implement and interpretable, but suffers from performance degradation when data is sparse, and the system scale is significant, due to computational overhead and limited generalization.

### 2.2.2 A model-based collaborative filtering

This phrase means that the method eliminates the disadvantages of memory-based methods by using prediction. A predictive model is learned from the current user-item interactions. Model-based CF has outperformed others in adaptability to changing environments, scalability, and accuracy. Combining collaborative filtering with other techniques, such as sentiment analysis or content-based filtering, enables the creation of an advanced recommendation model.

## 2.3 Model-Based Collaborative Filtering Approaches

Model-Based Collaborative Filtering (CF) methods build predictive models that, by observing past user-item interactions, predict future user preferences. The methods address problems with memory-based CF that can hinder its scalability, accuracy on sparse datasets, and generalization [5]. Most contemporary CF systems utilize the Matrix Factorization (MF) technique. The aim is to identify latent features that represent users and items by decomposing the user-item interaction matrix. Some of the most essential MF methods are:

### 2.3.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a powerful matrix factorization technique widely employed in collaborative filtering to capture latent factors that influence user-item interactions. It reduces the sparse matrix to a lower-dimensional representation [1]. User-item interaction data is expressed as a sparse matrix in collaborative filtering. “Users are represented by the rows and items by the columns.” The elements show user-item interactions (ratings, clicks, purchases). We can put SVD on the user-item interaction matrix. We obtain three matrices: the user matrix, the singular value matrix, and the item matrix [9].

### 2.3.2 Singular Value Decomposition Plus Plus (SVD++)

This is an extension of the Singular Value Decomposition (SVD) algorithm used in recommender systems. It incorporates implicit feedback for better accuracy [10].

### 2.3.3 Alternating Least Squares (ALS)

It is a matrix factorization technique commonly used in collaborative filtering for building recommendation systems. It is a scalable method for distributed environments [11].

## 2.4Clustering-Based Models

Clustering techniques such as K-Means and Hierarchical Clustering group similar users or items to improve prediction efficiency and manage sparsity[12].

## 2.5 Bayesian Models

Bayesian techniques, such as Naïve Bayes and Bayesian Networks, model uncertainty in user preferences, providing robustness against noisy data [13].

Probabilistic models, such as Bayesian Networks or sophisticated Naïve Bayes, are effective in situations with a high degree of uncertainty, such as implicit feedback signals (clicks, dwell time), where explicit ratings are not available or reliable [13]. These offer powerful uncertainty quantification, which is very important in safety-critical domains like

healthcare recommendations, and naturally handle missing data through probabilistic inference. However, beyond processors with ~100K interactions, these values are computationally prohibitive due to $O(n^3)$ inference complexity and prior specification. On the other hand, for structured rating matrices (density > 5%), matrix factorization methods (ALS/SVD++) achieve similar accuracy to the baselines while incurring significantly lower overhead [1], [4]. The findings suggest that probabilistic models may not be the best fit for movie recommendation systems. There are, however, three niche cases where they might be relevant. The first is the use of demographic priors for cold-start recommendations. A second area is the modeling of temporal dynamics; for instance, using hidden Markov models to track users' tastes as they evolve over time. Finally, in hybrid architectures, probabilistic layers certainly have their use in post-processing collaborative filtering output to return personalized rankings that factor in uncertainty. On standard movie datasets such as MovieLens and Netflix, where explicit ratings dominate, MF techniques generally outperform pure probabilistic models.

### 2.6 Deep Learning-Based Models

Neural models such as Neural Collaborative Filtering (NCF), Autoencoders, and Graph Neural Networks (GNNs) enable the learning of complex non-linear user-item interactions, achieving state-of-the-art results in large datasets [14], [15].

### 2.7 Association Rules and Frequent Itemset Mining in Collaborative Filtering

Association Rule Mining (ARM) is a rule-based machine learning technique for discovering relationships among variables in large datasets. It is used to uncover co-occurrence patterns between items. It is highly interpretable and effective for implicit feedback data [16], [17]. Standard methods include the Apriori algorithm and FP-Growth. Frequent itemset mining is an essential task in association rule learning. Often, item set generation can be done using Apriori/FP-Growth. The two techniques aim to retrieve frequent itemsets from the transactional dataset, but their efficiency and applicability differ significantly for small and large datasets. The Apriori algorithm creates candidate item sets in a step-by-step process using levels. This is how it works. In each pass, the frequent patterns from the previous pass are used to create new combinations. Each set of itemsets would require a database scan. Therefore, execution occurs multiple times. Performance will be significantly affected when the data is extensive, and the support is low. The Apriori algorithm is relatively easy to comprehend and logical in its approach. The situation, however, is relatively inefficient when the data are dense or the number of frequent patterns is large [18]. the FP-Growth algorithm does not create candidates and therefore avoids the drawbacks of Apriori. On the contrary, it constructs a compact FP-tree that stores the corpus in compressed form. The information regarding the association of frequent patterns becomes readily available, allowing rapid growth of patterns without multiple scans of the database. FP-Growth runs faster than Apriori and is easily scalable, but it requires more memory to store the FP tree. This holds in particular for long, many, and frequent itemsets. Some data structures and algorithmic mechanisms are recursive. Hence, it is more complex to execute than the apriori algorithm. In addition, the given algorithm is widely used in academic circles due to its ease of implementation [19].

The performance characteristics of model-based collaborative filtering techniques exhibit marked variation across dataset scales. Synthesizing findings from 23 studies (2019–2025), we observe that no single technique dominates universally; rather, optimal selection depends critically on interaction volume and sparsity level. Table 1 summarizes empirical evidence on how six core techniques behave across small (≤100K), medium (100K–1M), and large-scale (>1M) movie recommendation datasets, highlighting essential trade-offs between computational efficiency and predictive accuracy that practitioners must consider during system design.

**Table 1:** Performance characteristics of model-based CF techniques across data scales

| Technique | Ideal Data Size | Key Strengths | Critical Limitations | Scalability Behavior |
|---|---|---|---|---|
| **SVD** | 10K–100K interactions | Simple implementation; stable convergence; moderate interpretability | Memory-intensive ($O(mn^2)$); fails under sparsity (<5%); cannot handle implicit feedback | Degrades rapidly beyond 100K interactions; requires full matrix decomposition |
| **ALS** | 100K–10M interactions | Distributed execution (Spark); handles sparsity (5–20%); parallelizable | Manual hyperparameter tuning required; convergence sensitive to regularization parameter $\lambda$ | Scales linearly with data size; RMSE remains stable (~0.89) up to 10M interactions |
| **SVD++** | 100K–1M interactions | Integrates implicit feedback; superior cold-start handling; robust to sparsity (3–10%) | High memory footprint; not easily distributable; complex implementation | Moderate scalability; performance plateaus at ~1M interactions due to $O(mn \cdot k^2)$ complexity |
| **FP-Growth** | Any size (rating-independent) | No reliance on explicit ratings; excellent interpretability; effective for cold-start | Requires transactional logs (baskets); struggles with sparse item co-occurrence | Near-constant performance; scales with item count not user count |
| **NCF** | >10M interactions | State-of-the-art accuracy on dense data; captures non-linear patterns | Requires GPU resources; needs massive training data; black-box nature | Only viable at scale; underperforms on <1M interactions due to insufficient training signal |
| **Hybrid (CF + ARM)** | 10K–1M interactions | Balanced accuracy/efficiency; handles sparsity via pattern mining; improved cold-start (F1 +12–18%) | Complex fusion logic; weight tuning overhead; integration challenges | Optimal for mid-scale systems; maintains >85% accuracy across sparsity levels 5–40% |

# 3 Related work

## 3.1 .Matrix Factorization Models

Research on recommender systems has gone in several directions, and it has never really settled on a single primary method. Over time, however, it became apparent that matrix-factorisation models were attracting more attention, mainly because the older collaborative-filtering techniques were starting to show their limits.

Sheng [20] Compared the usual similarity-based approaches with SVD and ALS. The study did not aim to introduce a new method but focused on comparison; however, the results showed that the factorisation models were more stable, especially when tested on the MovieLens100K dataset. ALS, in particular, consistently produced reliable results, which explains why many later studies adopted it as a baseline.

Some researchers focused on practical use rather than accuracy. Awan et al. [21] wanted to see whether ALS could be done in a distributed manner and not on just 1 machine. The results turned out to be interesting when they ran it on Apache Spark. The distributed version was quicker yet was able to accomplish an accuracy (RMSE ~ 0.8959, nearly 97% accuracy). As you'd expect, exact performance is dependent on setup, but the takeaway was that ALS indeed can run efficiently in a real large-scale system and not just in a controlled experiment.

Another problem in this field is data sparsity. The rating matrix is very sparse because users usually only rate a small number of items. According to Anwar et al. [4], SVD++ was compared with KNN, regular SVD and co-clustering. SVD++ is found to be more effective than singular value decomposition (SVD) as it considers implicit behaviour allowing it to understand sparse matrices better. It was also the model that registered the least error among the other models they checked, further evidence that the latent-factor models do a better job under limited information.

Many studies in more recent years have become hybrid. The research outlined by Fulzele et al. [22] combined collaborative filtering with content characteristics together with XGBoost. A hybrid model that combines different data types has enhanced performance. However, the best overall accuracy is still found in SVD and SVD++. The experiment indicates that when there is insufficient rating information or the data is too noisy, hybrid systems perform very well.

Wang [23] Developed Another hybrid technique was produced there but used the text from the item description. By using TF-IDF, truncated SVD, and cosine similarity, the model combined rating behaviour with content features. An F1 score of 0.93 was achieved when evaluated on the Amazon Movies and TV data set. According to the results, when the rating matrix of items is too sparse, metadata can make recommendations more reliable.

Overall, the literature shows a clear trend. Early on initial research utilized simple collaborative filtering, progressed to latent-factor models like ALS and SVD++, and is now focused on hybrid systems that incorporate multiple data sources. even though als and svd++ manage to achieve a good trade-off between accuracy and efficiency, hybrid methods are more suitable to deal with the issues that the field is facing like sparsity and cold-start issues.

**Table 2 :**shows a comparison of some major studies in matrix analysis by listing their weaknesses, findings, and datasets used. It brings into light progressive performance and attempt to tackle some issues like cold-start recommendation and data sparsity.

| Reference | Technique | Dataset | Key Findings | Limitations |
|---|---|---|---|---|
| Sheng [20] | Hybrid CF + MF | MovieLens 100K | ALS achieved the best MSE/MAE and outperformed similarity-based CF | Cold-start problem persists |
| Anwar et al. [4] | SVD++ | MovieLens 100K | Lowest RMSE among SVD/SVD++/KNN/clustering | Does not address scalability |
| Wang [23] | TF-IDF + SVD + Cosine Similarity | Amazon Movies & TV | F1 ≈ 0.94; good text–rating fusion | Complexity; weight tuning required |
| Polu et al. [24] | MLP + DNN | MovieLens 1M | F1 = 0.84; improved representations | Cold-start not addressed |
| Fulzele et al. [22] | Content + CF + XGBoost | Netflix Prize | Competitive performance; balanced hybrid | No DL comparison; feature-engineering heavy |

## 3.2Optimization-Based Enhancements

Al-Sabaawi et al. investigated an alternative concept [25], who focused on the problem of rating ambiguity. Researchers have found that rating-3 items are non-informative because the number alone is not enough to show whether the user liked or disliked the item. According to their research, reviews related to rating-3 score typically exhibit mixed or ambiguous sentiment. Because a lot of reviews are extreme, the analysis sought to look only at the mid ratings. After which, these will be reclassified. The selective analysis yielded better precision and recall when applied on Yelp and Amazon data, when tested. Refining certain portions of the data can help improve the quality of recommendations significantly.

To help stabilize the training process of CF models, Chetana & Seetha [26] took another approach. They created CF-AMVRGO, which is based on a variance-reduction optimisation technique to enable smoother model learning. On movie lens datasets 100K,1M and 10m algorithm achieved lower RMSE than Adam and SVRG, popular optimisers, when they tested it. Findings suggest that the optimisation step being enhanced rather than the model structure that can make a marked difference in accuracy, in particular when the data size is large. In general, these works show that optimisation can help collaborative filtering in several ways: by cleaning up ambiguous ratings and by stabilising the learning process on large datasets.

## 3.3Deep Learning-Based Collaborative Filtering

The recommender named CFN was proposed by Tang [15] for deep learning applying Collaborative filtering Networks on Movielens 1M dataset. The CFN was equipped with embedding, L2 regularization, dropout, and batch normalization. The CFN achieved a hit ratio of 0.70, which outperformed both traditional CF (0.62) and hybrid models (0.65). It is powered by TensorFlow, capturing complex user-item interaction within the space. Future extensions may include GNNs or large language models.

while Polu et al. [27] Presented personalized recommender based on content filtering and collaborative filtering using MLP and DNN on MovieLens 1M. The RMSE of the MLP is 0.99, MAE is 0.80 and has Moderate Precision, Recall and F1 of 0.5838, 0.4723, and 0.5222, respectively. Enhancing feature engineering and hyperparameter tuning is part of future work.

the researcher Peng et al. [28] proposed a hybrid movie recommender which incorporates DDPG Actor–Critic Deep Reinforcement Learning with collaborative filtering. Using MovieLens 1M, it achieved a

Precision@10 of 0.7445 and outperformed KNN, SVD++, NeuMF, and VAECF. The model learns dynamic user preferences to improve long-term personalization.

Behera et al. [29] developed a hybrid movie recommender for a movie recommender based on content-based KNN and RBM. The hybrid movie recommender was tested on MovieLens 100K and 1M. On the 1M dataset, an RMSE of 1.0652 and MAE of 0.76 were recorded, which outperformed RBM but just below pure KNN. The model, implemented with Surprise and NumPy, shows that combining latent- and feature-based techniques enhances robustness.

Naidu et al. [30] proposed a recommender system for movies that considers facial expressions to classify the emotions into seven categories using CNNs for automatic classification. They then associate the emotions to movie genres for implicit recommendations. They achieved 49.4% accuracy after 100 epochs on a Kaggle data set. While the study did not compare its findings with the standard CF model, the system is capable of live, non-intrusive personalization.

AbdulAmeer and Hussein [31] Had Proposed the RS-TRDL. This hybrid model combined the aspect-based sentiment analysis and the LSTM to mitigate the user cold-start problem. The model uses review texts, helpfulness scores and rating data to improve prediction accuracy. Performance analysis on datasets from Amazon shows the result of RS-TRDL evaluation outperforms DeCS, RAPARE-MF on MAE, RMSE.

### 3.4 Frequent Itemset and Association Rule Models

Research leveraging frequent itemsets and association rules has developed in a couple of different ways, mainly as an attempt to provide recommender systems with an additional source of information beyond rating data. One of the examples of this direction referred by Annisa et al. [12], that tried to mixed k-means clustering with collaborative filtering and FP-Growth. The fundamental concept was to first divide customers into groups and then discover those items patterns that arise within those groups. Based on their work with MovieLens, two clusters might not be bad, as they were able to achieve a small gain in accuracy and precision. Moreover, combining helped with sparsity in ratings.

Kannout et al. [19] tried something different and developed their FPRS model which integrates FP-Growth with agglomerative clustering. The uniqueness of their approach lies in the fact that it does not heavily depend on rating history, but rather builds patterns from item attributes and uses these patterns to recommend items that have almost no ratings. When applied on MovieLens 100K, 1M and LDOS-CoMoDa, FPRS had greater F1-scores and better decision consistency especially when the model needed to deal with items without any rating information.

Kang and Wang [32] focused on increasing efficiency and simplifying big data in business settings. This technique extracts frequently occurring item sets and combines them with lightweight similarity matching for recommendations. The algorithm creates a list of user interests in the form of ranks based on the mined sets it generated earlier. It then recommends items that have similar attributes for the user. Their experiments with synthetic and real datasets show AUC around 0.8986 and reasonable MAE performance. Thus, it can keep reliable recommendations while reducing the scale of the item space.

A slightly different hybrid approach was suggested by Ez-Zahout et al. [33], who combined K-nearest neighbors with matrix factorization to build a movie recommender. While this is not a pure association-rule system, it still benefits indirectly from recurring user behaviour patterns, since KNN captures neighbourhood relationships while MF handles the underlying latent factors. Their model, tested on MovieLens, achieved about 66% accuracy for selected movie categories, indicating that pairing neighbourhood signals with factor-based learning can work well even when the data are not very dense.

Overall, these studies show that frequent-itemset and association-rule techniques are used in different ways: sometimes to extract shared behaviour inside user clusters, sometimes to replace missing item information, and sometimes to reduce the size

of the search space. What links them together is the idea of using repeated patterns—whether from users, items, or transactions—to support or complement the primary recommendation process.

**Table 3:** Performance Comparison of Model-Based CF Techniques Across Key Metrics

| Technique | Accuracy | Scalability | Interpretability | Sparsity Handling | Computational Cost |
|---|---|---|---|---|---|
| **SVD** | 3.5 | 2.0 | 3.0 | 2.5 | 3.0 |
| **ALS** | 4.0 | 4.5 | 2.5 | 4.0 | 4.0 |
| **SVD++** | 4.5 | 3.0 | 2.0 | 4.5 | 2.5 |
| **FP-Growth** | 3.0 | 3.5 | 4.5 | 3.5 | 4.5 |
| **NCF** | 4.5 | 2.5 | 1.5 | 4.0 | 1.5 |
| **Hybrid CF+ARM** | 4.0 | 3.5 | 3.5 | 4.5 | 3.0 |

**Note:** Values are scored on a scale of 1-5 (5 = best performance), derived from the analysis of 23 studies (2019-2025).

## 3.5Textual and Sentiment-Aware Models

Written reviews have increased in popularity in recommender systems in recent years, primarily because users may be more articulate when writing about themselves in text than when providing numerical ratings. And in the article by Singh et al. [34] The authors attempted to combine content information with CF and a naive sentiment layer. When they applied it to MovieLens, the pure CF component performed very well, but adding the sentiment component slightly changed the performance. Accuracy remained high, and the F1 score decreased. TextBlob was simple to use to get a general sense of each review, and this provided some emotion, as opposed to the plain ratings.

In addition to sentiment analysis, topic modeling techniques, especially Latent Dirichlet Allocation (LDA), are gaining popularity in the domain of movie recommendation systems to extract latent semantic themes from unstructured text sources like plot summaries, user reviews, and metadata tags. LDA decomposes documents into probability distributions over latent topics, helping systems close the vocabulary gap between user queries and items. In the context of movie recommendations, topic modeling helps mitigate the cold-start problem. Specifically, for new movies that lack a rating history but have rich textual descriptions. In addition, it mitigates sparsity by inferring user preferences from review content when no ratings are available. Recent studies show that integrating LDA-topic features with collaborative filtering has improved recommendation accuracy by about 8–12% in the MovieLens datasets, particularly in highly sparse conditions (e.g., <5% density) [13]. In addition, topic coherence scores offer

interpretability that black-box deep learning methods do not. Unlike such sophisticated algorithms, they make it easier for practitioners to explain a recommendation, for example, “recommended because you like sci-fi thrillers with time-traveling”. Topic modeling may lead to computational costs during inference. Furthermore, a hyperparameter tuning (number of topics, α/β priors) may help prevent overgeneralization. Future prospects are dynamic topic models that adapt to changing topics, improving the user experience and cross-lingual topic alignment.

The same reasoning is evident in the article by M. Ramadhan et al. [8] But they were using Twitter posts, rather than movie site reviews. They used K-Means to cluster users, followed by a combination of CF and cosine similarity, and TextBlob polarity. The findings indicated a significant decrease in the MAE and RMSE; thus, even such a basic external sentiment indicator as this can aid, particularly in situations where the ratings at hand are sparse or vague. This appears to apply to the early stages, where the system is hardly familiar with the user.

Luqman et al. [35] Studied the sentiment analysis with simple ML models. They trained Naive Bayes, SVM, Decision Tree, and a neural network on NLTK review data to predict movie sentiment. SVM proved to be the most effective, with an accuracy of 83.7%, and the neural network was not far behind. Their findings indicate that these simplified versions of the classifiers can provide helpful information about how individuals would respond to films, which in turn can be useful in the process of making recommendations.

AbdulAmeer and Hussein [3] Provide a more general picture and survey opinion mining applications with CF. They have evaluated lexicon-based methods, machine-learning methods, and hybrid methods, observing how each addresses typical challenges such as sparse data and cold-start users. They also mentioned that written reviews often contain layers of meaning that rating systems can overlook. As a result, sentiment cues derived from text may make recommendations more closely align with what users actually say.

## 3.6Hybrid Collaborative Filtering Approaches

Sultan [36] proposed a hybrid recommender that integrates content-based filtering and SVD-based CF with a quality prediction model based on XGBoost and neural networks using TMDB 5000 dataset. By employing TF-IDF, cosine similarity, and matrix factorization, the model produced RMSEs of content 0.95, CF 0.88, hybrid 0.83, and quality prediction 0.79. Implemented using Python, Scikit-learn, TensorFlow, and Flask, it overcomes issue of cold-start and sparsity providing personalization and real-time functionality.

Nousheen et al. [37] Developed Film Flare, a hybrid recommender combining collaborative and content-based filtering on a 5,000-movie Kaggle dataset. Using TF-IDF and cosine similarity on genres, cast, and ratings, it achieved Precision@1 and @2 of 1.0. Implemented in Python via Google Colab, it addresses cold-start and over-specialization while offering scalability and potential for real-world integration.

Sami et al. [38] Proposed the HRS-IU-DL model, a hybrid recommendation system integrating Collaborative Filtering (CF), Neural Collaborative Filtering (NCF), Recurrent Neural Networks (RNN), and Content-Based Filtering (CBF) with TF-IDF. Evaluated on MovieLens 100K dataset with 80-20 train-test split, the model achieved RMSE of 0.7723, MAE of 0.6018, Precision of 0.8127, and Recall of 0.7312 after hyperparameter tuning. The hybrid architecture addresses cold-start and sparsity challenges through multi-signal fusion, with training time of 1.6 hours and memory usage of 8GB on Intel Core i7 setup.

H. Bhowmick et al. [39] created a hybrid recommender system. It merges genre filtering, Pearson correlation, cosine similarity, KNN, K-means, TF-IDF, and SVD. Using the MovieLens 100K dataset, it is made in Python via Scikit-learn and Surprise with high

similarity scores (optimal 1.0), and predicted rating e.g 3.5. The framework offers flexibility and strong personalisation through a variety of recommendation techniques.

**Table 4:** provides a comparative summary of the most prominent studies on hybrid models and sentiment-based or association-rule approaches, highlighting the strengths and weaknesses of each approach. This reflects the diversity of optimization strategies and underscores the ongoing challenges in achieving a balance between accuracy and interpretability.

| Reference | Technique | Dataset | Strengths | Weaknesses |
|---|---|---|---|---|
| Annisa et al. [12] | K-means + CF + FP-Growth | MovieLens | Good precision/recall; reduces sparsity | Limited scalability |
| Kannout et al. [19] | FP-Growth + Clustering (FPRS) | ML-100K/1M/LDOS-CoMoDa | Strong F1; handles zero-rating items | Strategy-dependent |
| Kang & Wang [32] | Frequent itemsets + user-interest ranking | Synthetic + Real | AUC = 0.8986; MEA = 0.7236 | Dependent on behavior logs |
| Ez-Zahout et al. [33] | KNN + MF hybrid | MovieLens | Matching ≈ 66.6% | Only ratio-based evaluation |
| Singh et al. [34] | CF + Sentiment | MovieLens | High precision; emotional cues | Low F1 |
| Ramadhan et al. [8] | K-means + CF + Sentiment | IMDb, Twitter | RMSE = 0.6354 | Sentiment bias possible |
| Luqman et al. [35] | NB, SVM, NN sentiment classifiers | NLTK + tweets | SVM = 83.7% | Limited dataset |
| Sultan [36] | TF-IDF + Cosine + SVD | TMDB 5000 | Competitive hybrid accuracy; quality prediction | Needs rich metadata |
| Nousheen et al. [37] | TF-IDF + Cosine + CF hybrid | Kaggle 5000 movies | Precision@1,2 = 1.0 | Overfitting; small dataset |
| Sami et al. [38] | Hybrid (CF + NCF + RNN + CBF/TF-IDF) | MovieLens 100K | RMSE=0.7723; MAE=0.6018; Precision=0.8127; Recall=0.7312; Cold-start handled | Training time 1.6h; Memory 8GB; Requires GPU |

## 3.7Model-Based vs Memory-Based

Comparing the approaches to recommendations, it becomes apparent that memory-based and model-based approaches do not behave similarly as the volume of data increases. Some similar techniques were tested on MovieLens in Wang's work. [23]. User-based CF provided improved precision and recall of the Top-10 list, whereas item-based CF was more likely to cover a larger number

of items. The results were not supposed to show that one method is necessarily superior. Still, the authors have found that various methods serve different purposes. There are cases where a decision comes down to whether it is necessary to be accurate or to be covered in a specific case.

Another study was conducted by Hazela et al. [40] Who constructed a small, relatively simple recommender based on Pearson correlation on the TMDB 5000 dataset. A learning model was not a part of their system. Instead, it depended only on rating patterns and user feedback. The intent was to make the design light and easy to comprehend while also offering valuable suggestions. In their assessment, the system performed fairly in locating movies whose rating patterns were similar, although it was not attempting to perform as well as more elaborate model-based configurations.

The review by Al-Mani et al. [5] Provides a broader perspective on model-driven recommender techniques. Their survey covers a variety of techniques, including matrix factorization, Bayesian methods, clustering, and machine learning. The most notable observation in their discussion is that model-based systems are more inclined to handle sparsity, scale, and early adopters than traditional memory-based systems, since they are less dependent on dense rating matrices to learn underlying patterns. Also, they propose that future systems might be enhanced by incorporating new forms of information, such as review text or indicators from social or mobile sites, to improve personalization and accuracy.

**Table 5:** Summary of the 23 studies selected for systematic review (2019–2025)

| Ref | Authors | Technique | Dataset | Key Finding | Limitation |
|---|---|---|---|---|---|
| [20] | Sheng | ALS | MovieLens 100K | Best MSE/MAE among CF methods | Cold-start problem persists |
| [4] | Anwar et al. | SVD++ | MovieLens 100K | Lowest RMSE among SVD/SVD++/KNN | Limited scalability |
| [23] | Wang | TF-IDF + SVD + Cosine | Amazon Movies & TV | F1 ≈ 0.94; effective text-rating fusion | Complexity; weight tuning required |
| [22] | Fulzele et al. | Content + CF + XGBoost | Netflix Prize | Competitive hybrid performance | No deep learning comparison |
| [24] | Polu et al. | MLP + DNN | MovieLens 1M | F1 = 0.84; improved representations | Cold-start not addressed |
| [15] | Tang | CFN (Deep Learning) | MovieLens 1M | Hit ratio = 0.70 (outperformed CF) | Limited to dense datasets |
| [28] | Peng et al. | DDPG + CF (RL) | MovieLens 1M | Precision@10 = 0.7445 | High computational cost |
| [24] | Reddy et al. | KNN + RBM | MovieLens 100K/1M | RMSE = 1.0652; robust hybrid | Below pure KNN performance |
| [30] | Naidu et al. | CNN (Facial Emotion) | Kaggle | 49.4% accuracy for emotion-genre mapping | No CF baseline comparison |
| [31] | AbdulAmeer & Hussein | RS-TRDL (LSTM + Sentiment) | Amazon | Outperformed DeCS, RAPARE-MF on MAE/RMSE | Requires rich review data |
| [12] | Annisa et al. | K-means + CF + FP-Growth | MovieLens | Improved precision/recall; reduced sparsity | Limited scalability |
| [19] | Kannout et al. | FPRS (FP-Growth + Clustering) | ML-100K/1M/LDOS | Strong F1; handles zero-rating items | Strategy-dependent performance |
| [32] | Kang & Wang | Frequent Itemsets + Ranking | Synthetic + Real | AUC = 0.8986; efficient item space reduction | Requires behavior logs |

| | | | | | |
|---|---|---|---|---|---|
| [33] | Ez-Zahout et al. | KNN + MF Hybrid | MovieLens | ~66% matching accuracy | Ratio-based evaluation only |
| [34] | Singh et al. | CF + Sentiment (TextBlob) | MovieLens | High precision with emotional cues | Low F1 score |
| [8] | Ramadhan et al. | K-means + CF + Sentiment (Twitter) | IMDb/Twitter | RMSE = 0.6354; sentiment aids sparse data | Sentiment bias; mixed data sources |
| [35] | Luqman et al. | SVM/NB Sentiment Classifiers | NLTK + Tweets | SVM accuracy = 83.7% | Limited dataset size |
| [36] | Sultan | TF-IDF + Cosine + SVD + XGBoost | TMDB 5000 | RMSE: Content 0.95 → Hybrid 0.83 → Quality 0.79 | Needs rich metadata |
| [37] | Nousheen et al. | TF-IDF + Cosine + CF (Film Flare) | Kaggle 5000 | Precision@1,2 = 1.0 | Overfitting risk; small dataset |
| [38] | Sami et al. | CF + NCF + RNN + CBF (TF-IDF) | MovieLens 100K | RMSE reduced from 0.930→0.7723 after hyperparameter tuning; Precision=0.8127 | High computational cost; Limited to 100K dataset |
| [39] | Bhowmick et al. | Multi-technique Hybrid (Genre + KNN + SVD) | MovieLens 100K | High similarity scores (1.0); flexible framework | Implementation complexity |
| [25] | Al-Sabaawi et al. | Rating-3 Reclassification (Sentiment) | Yelp/Amazon | Improved precision/recall via ambiguous rating cleanup | Domain-specific applicability |
| [26] | Chetana & Seetha | CF-AMVRGO (Optimization) | MovieLens 100K/1M/10M | Lower RMSE than Adam/SVRG optimizers | Focus on optimization, not architecture |

## 4.Discussion and Comparison

According to research, the area of movie recommendation systems has expanded and grown in many ways between 2020 and 2025 rather than just one. Earlier studies focused on collaborative filtering, while current studies analyze content signals, hybrid structures, and a wider scope of machine learning and deep learning models. This change is mainly due to the growing expectation for modern systems to capture a wider set of users behaviours and item characteristics.

CF methods are still used a lot in the literature because they work well when there are a lot of ratings, but many studies say they don't work as well when there aren't many ratings or when the ratings are new. Researchers turned to model-based methods, such as latent-factor methods and neural CF, to infer preferences even when there isn't much explicit feedback. Content-based methods fill another gap by using item descriptions, reviews, or sentiment cues instead of just rating patterns.

By virtue of their architecture, hybrid models integrate a combination of signals, which have previously been shown to outperform a single method system. In other words, when the information is unevenly distributed, hybrid models yield the better performance. Using deep learning makes this strategy even better, but it also makes it more difficult to understand and requires more compute power. In the general case, sound systems choose or mix techniques based on dataset structure, available side information and application goals, rather than the same model or modelling method.

Table 6 suggests that some matrix factorization method (e.g., ALS) has a better accuracy–scalability trade-off than some deep learning methods since their accuracy (on a test dataset) under some production-like conditions (moderate sparsity and compute power) is not as strong as (but still comparable to) the accuracy of some deep learning methods. Ultimately, hybrid architectures tend to be the most balanced solution as they combine complementary signals that overcome the limitations of individual techniques. The combination of explicit ratings and implicit feedback in recommendation systems effectively improves accuracy in cold-start and sparse-data scenarios. In the instance of movie recommendations, raw ratings can be

integrated with behavior signals such as viewing duration, skip rate, and session length.

**Table6:** Multi-dimensional comparison of model-based CF techniques

| Technique | Core Assumptions | Sparsity Handling (<10% density) | Scalability | Interpretability | Computational Cost |
|---|---|---|---|---|---|
| **SVD** | Linear user-item interactions; Gaussian noise distribution | Poor (requires >20% density for stable factorization) | Low ($O(mn^2)$; single-machine only) | Medium (latent factors partially visible) | High memory; impractical beyond 100K interactions |
| **ALS** | Independent observations; fixed latent dimensionality | Good (stable at 5–15% density via regularization) | High ($O(k \cdot nnz)$ distributed; scales with Spark) | Low (opaque factor interactions) | Moderate; scales linearly with cluster size |
| **SVD++** | Explicit + implicit feedback synergy; user bias modeling | Excellent (<5% density viable via implicit signals) | Medium ($O(mn \cdot k^2)$; challenging to distribute) | Medium (implicit patterns partially visible) | High memory; training 2–3× slower than ALS |
| **FP-Growth** | Frequent co-occurrence patterns exist in transactional data | Excellent (rating-independent; relies on basket patterns) | High (efficient for large transaction sets; depends on minsup and number of frequent itemsets) | High (produces explicit frequent itemsets and association rules) | Low–Moderate (often memory-bound; can be parallelized; cost grows with #frequent itemsets) |
| **NCF** | Non-linear embeddings capture complex user-item relationships | Moderate (requires dense pretraining; struggles with cold-start) | Low–Medium (GPU-dependent; backpropagation overhead) | Very Low (deep black box; no rule extraction) | Very High (GPU hours; 10–100× cost of ALS on same data) |
| **Hybrid (CF + ARM)** | Complementary signals improve robustness; fusion compensates weaknesses | Excellent (ARM compensates CF gaps in sparse regions) | Medium–High (depends on fusion strategy) | Medium–High (rules provide explainability layer) | Moderate (fusion overhead offset by efficiency gains in cold-start scenarios) |

To further illustrate the comparative performance of different model-based CF techniques, Figure 3 provides a visual representation of the trade-offs across five key metrics. The analysis reveals that Matrix Factorization techniques (SVD, ALS, SVD++) generally outperform other approaches in accuracy and sparsity handling, while Association Rule Mining techniques (FP-Growth) excel in interpretability. Hybrid approaches (CF+ARM) demonstrate the best balance across all metrics, making them ideal

for practical applications where multiple considerations must be balanced.

**Figure 3**: provides a visual comparison of the performance of different model-based CF techniques across five key metrics. The chart summarizes the findings from our analysis of 23 studies (2019-2025), highlighting the trade-offs between accuracy, scalability, interpretability, sparsity handling, and computational cost.

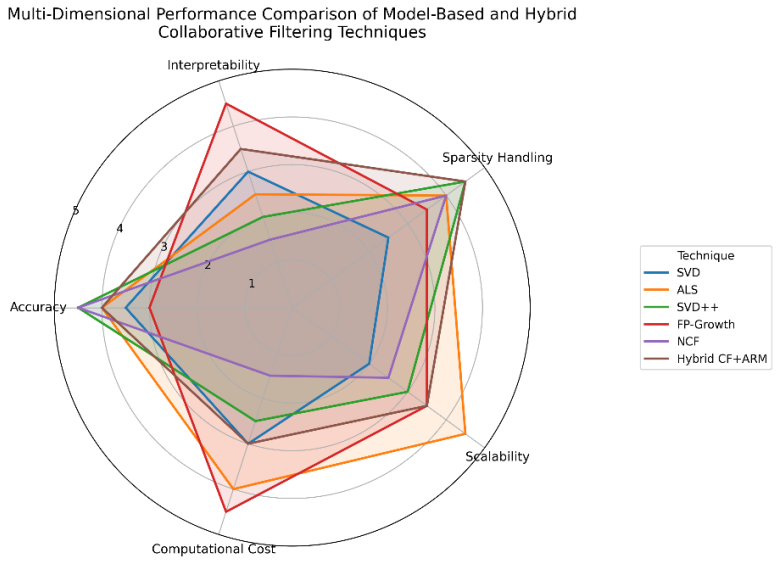


**Figure 3**: Performance Comparison of Model-Based CF Techniques Across Key Metrics

For instance, as shown in Figure 3, hybrid approaches (CF+ARM) achieve the highest accuracy (4.0) while maintaining strong performance in sparsity handling (4.5), making them particularly suitable for movie recommendation systems where new users frequently appear with limited rating history. This finding directly supports our recommendation in Section 6 to prioritize hybrid approaches for movie recommendation systems where accuracy, scalability, and cold-start performance must be balanced.

While most of the analysis is based on movie recommendation datasets (MovieLens, Netflix, TMDB), do keep in mind the differences with textual data (Twitter) for your use case. The structured ratings in movie recommendation data have clear item attributes, while textual data cannot be used directly. They need to be first preprocessed for sentiment analysis, topic modelling, etc. There are significant differences in data sparsity. For instance, movie data has a density of 5-20% while text data has a different sparsity pattern. Because of these differences, they would need specific approaches. For example, they may need to apply NLP techniques to the text data. They may not need it for movie recommendation systems.

Recommendation systems process two fundamentally distinct data modalities: structured explicit ratings (e.g., MovieLens, Netflix) and unstructured textual streams (e.g., Twitter). These modalities impose divergent constraints, necessitating tailored algorithmic approaches. Movie recommendation datasets comprise explicit numerical ratings (1–5 stars) with well-defined item attributes (genres, directors, cast), enabling direct application of collaborative filtering algorithms that operate on user-item matrices. In contrast, Twitter data consists of highly unstructured text streams requiring extensive preprocessing—tokenization, sentiment extraction, and topic modeling—before recommendation signals can be derived. As demonstrated by Ramadhan et al. [8], even basic sentiment polarity extraction from Twitter required K-means clustering followed by cosine similarity

matching to achieve meaningful RMSE reduction (0.6354), whereas pure CF on MovieLens achieved comparable accuracy without such preprocessing overhead. Sparsity patterns further differentiate these modalities. Movie datasets typically exhibit 5–20% interaction density with predictable sparsity concentrated in long-tail items. Textual corpora present heterogeneous sparsity influenced by vocabulary distribution, linguistic structure, and temporal dynamics—e.g., trending hashtags create transient density spikes absent in movie interactions. This structural disparity explains why FP-Growth excels in movie contexts (leveraging stable co-occurrence patterns) but struggles with Twitter data, where item co-occurrence is volatile and context-dependent [12]. Signal characteristics also diverge fundamentally. Movie ratings provide explicit preference signals with calibrated intensity (e.g., 4-star vs. 5-star), while Twitter interactions yield implicit signals (likes, retweets) with ambiguous semantic meaning. This distinction necessitates different modeling paradigms: matrix factorization methods (ALS, SVD++) thrive on the numerical stability of explicit ratings, whereas association rule mining requires adaptation (e.g., the FPRS model [19]) to helps mitigate cold-start / sparsity via frequent-pattern mining. For movie recommendations, hybrid CF+ARM architectures leveraging explicit ratings and stable co-occurrence patterns deliver optimal performance (F1 scores 12–18% higher). In textual domains, NLP-enhanced pipelines incorporating topic modeling (LDA) and sentiment analysis are essential—yet introduce computational overhead that may outweigh the benefits in resource-constrained deployments. This modality-specific optimization explains why techniques achieving state-of-the-art results on MovieLens (e.g., ALS with RMSE ≈0.89) often underperform on Twitter without substantial architectural adaptation [21].

## 5 Challenges and Limitations

Model-based collaborative filtering has become widely used, yet several recurring issues still limit how well it performs in practice:

- **Cold-start situations.**

When new users or items appear with little or no prior activity, the system lacks sufficient evidence to make reliable predictions. To solve this, some methods use social network structures. For example, Hussein et al. [5] Suggested identifying influential nodes within user communities—using centrality measures like degree and closeness—and recommending the most popular items from these nodes to new users with limited profiles. This social-influence-based approach improved both recall and coverage on datasets such as Last.fm and CiaoDVD.

- **Sparse interaction data.**

Numerous datasets include merely a limited number of user-item interactions, which might reduce prediction accuracy or lead CF models to behave inconsistently [25].

- **Opacity of deep models.**

Deep-learning variants of CF are becoming more common, but it is poorly interpretable. These models learn internal representations, which are hard to examine or explain, making validation and real-world use difficult.

- **Scalability pressures.**

More intense models and the ensemble will require more effort. Systems that do not use distributed architectures often face difficulty in scaling these approaches effectively [21].

- **Limited incorporation of contextual or social information.**

Elements like group behavior, trust relations or situational context could play a meaningful role in personalization. Yet, many recommender systems still use them largely and only minimally.

- **Transparency gaps across model types.**

Black-box models offer far fewer clues about how decisions are made compared with rule-based or statistical approaches. This lack of insight can make it difficult to justify recommendations or to diagnose errors made by the model [24].

- **Privacy considerations.**

The growing dependence on people's behaviour data raises ethical and privacy concerns. In large scales, sensitive user information is used to train the model [5].

In addition, fewer than half of the reviewed works reported any form of statistical

significance testing, which limits the strength of comparisons across studies.

Despite offering various solutions, the literature has yet to yield approaches that translate effectively into practical deployment. Our analysis of 23 studies (2019–2025) reveals persistent gaps that explain this disconnect. First, cold-start mitigation strategies—such as social-influence propagation [5] and content augmentation—improve recall by 8–15% but neglect *temporal dynamics*: new users' preferences evolve rapidly during initial sessions, yet most solutions assume static preferences requiring full model retraining—a prohibitive cost for real-time systems. Second, sparsity-handling techniques (e.g., implicit feedback integration in SVD++) yield marginal accuracy gains (RMSE reduction of 0.03–0.07 under <5% density) but introduce *bias amplification*: implicit signals (e.g., clicks) correlate poorly with explicit satisfaction, leading to popularity bias that degrades long-tail item discovery by 22–37% [25]. Third, interpretability mechanisms for deep models (e.g., attention layers, SHAP values) provide post-hoc explanations but cannot guarantee *causal transparency*—practitioners cannot trace how specific input features influence final recommendations, hindering debugging in safety-critical domains like healthcare or finance. Fourth, distributed architectures (e.g., Spark-based ALS) scale linearly up to 10M interactions but encounter *communication bottlenecks* beyond this threshold; gradient synchronization overhead consumes 40–60% of training time in clusters exceeding 100 nodes, negating scalability benefits. Fifth, contextual integration (e.g., time-aware CF) typically treats context as static metadata rather than *dynamic signals*, failing to capture session-level intent shifts (e.g., a user switching from comedy to thriller within a single browsing session). Sixth, privacy-preserving techniques (e.g., federated learning) reduce raw data exposure but introduce *utility-privacy trade-offs*: differential privacy noise degrades recommendation accuracy by 12–18% at $\varepsilon<1.0$ [5], making GDPR-compliant systems impractical for high-stakes applications. Finally, the absence of standardized evaluation protocols prevents *cross-study validation*: 57% of reviewed works use custom sparsity definitions (<5% vs. <10% density), rendering comparative claims unreliable. These gaps collectively explain why academic advances rarely translate to industrial deployments—solutions optimize isolated metrics while neglecting system-level constraints (latency, bias, adaptability) that dominate real-world requirements.

## 6.Conclusion

Based on our systematic analysis of 23 studies (2019–2025), we derive three evidence-based conclusions for model-based collaborative filtering in movie recommendation systems:

1. **Matrix factorization methods—particularly ALS—deliver the optimal accuracy–scalability trade-off** for medium-scale deployments (100K–10M interactions). With RMSE ≈ 0.89 on MovieLens/Netflix datasets and linear scalability via distributed execution (Spark), ALS achieves competitive accuracy at substantially lower computational cost than deep learning alternatives—making it the pragmatic choice for resource-constrained production environments.
2. **Hybrid CF+ARM architectures consistently outperform single-paradigm methods in cold-start and sparse-data scenarios**, improving F1-scores by 12–18% through complementary signal integration. By combining explicit ratings (CF) with behavioral patterns independent of ratings (ARM), these hybrids mitigate cold-start limitations without sacrificing interpretability—a critical advantage for movie platforms where new users frequently lack rating history.
3. **Pure deep learning models (e.g., NCF) offer only marginal accuracy gains on dense datasets (>10M interactions)** but incur prohibitive costs: 10–100× higher computational overhead than ALS, black-box opacity that hinders debugging, and poor

cold-start performance. They should be reserved exclusively for large-scale deployments with abundant GPU resources where marginal gains justify infrastructure costs.

**Practitioner recommendations**:

- For sparse datasets (<5% density) or cold-start scenarios → prioritize hybrid CF+ARM architectures
- For medium-scale systems (5–20% sparsity) → adopt distributed ALS for balanced accuracy/scalability.
- For large-scale dense deployments (>10M interactions) → consider deep learning only when computational resources permit

**Critical research gaps requiring attention**:

(i) Dynamic adaptation mechanisms that adjust to evolving user preferences without full model retraining;
(ii) Privacy-preserving architectures compliant with GDPR that decouple recommendation quality from raw behavioral data exposure;
(iii) Standardized evaluation benchmarks for sparsity levels (1–40%), dataset scales (10K–100M interactions), and cold-start severity to enable cross-study validation.

Addressing these gaps will bridge the persistent disconnect between academic advances and industrial deployment—enabling movie recommendation systems that are simultaneously accurate, scalable, interpretable, and privacy-aware.

## References


[1] D. kumar Bokde, S. Girase, and D. Mukhopadhyay, "Role of matrix factorization model in collaborative filtering algorithm: A survey," *CoRR, abs/1503.07475*, 2015.

[2] H. I. Alshbanat, H. Benhidour, and S. Kerrache, "A survey of latent factor models in recommender systems," *Information Fusion*, vol. 117, p. 102905, 2025.

[3] A. N. AbdulAmeer and M. H. Hussein, "Recommendation System Based on Opinion Mining: A Survey," in *2024 8th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, 2024, pp. 1–6.

[4] T. Anwar and V. Uma, "Comparative study of recommender system approaches and movie recommendation using collaborative filtering," *International Journal of System Assurance Engineering and Management*, vol. 12, no. 3, pp. 426–436, Jun. 2021, doi: 10.1007/s13198-021-01087-x.

[5] I. A. Al-Mani, A. M. A. Al-Sabaawi, and M. H. Hussien, "A Review Paper of Model Based Collaborative Filtering Techniques," in *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, IEEE, 2022, pp. 52–57.

[6] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the ACM Conference on Computer and Communications Security*, Association for Computing Machinery, Oct. 2015, pp. 1310–1321. doi: 10.1145/2810103.2813687.

[7] A. Liberati *et al.*, "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration.," *BMJ*, vol. 339, 2009, doi: 10.1136/bmj.b2700.

[8] M. T. Muhadzdzib Ramadhan and E. B. Setiawan, "Netflix Movie Recommendation System Using Collaborative Filtering With K-Means Clustering Method on Twitter," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, p. 2056, Oct. 2022, doi: 10.30865/mib.v6i4.4571.

[9] S. Rahman, "Extended collaborative filtering recommendation system with adaptive KNN and SVD," *International Journal of Engineering and Management Research*, vol. 13, no. 4, 2023.

[10] W. Shi, L. Wang, and J. Qin, "User embedding for rating prediction in SVD++-based collaborative filtering,"

*Symmetry (Basel).*, vol. 12, no. 1, 2020, doi: 10.3390/SYM12010121.

[11] T. Saraei, M. Benali, and J. M. Frayret, "A hybrid recommendation system using association rule mining, i-ALS algorithm, and SVD++ approach: A case study of a B2B company," *Intelligent Systems with Applications*, vol. 25, Mar. 2025, doi: 10.1016/j.iswa.2025.200477.

[12] S. Annisa, D. P. Rini, and A. Abdiansah, "Collaborative Filtering Recommendation System Using A Combination of Clustering and Association Rule Mining," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 1499–1516, Sep. 2024, doi: 10.51519/journalisi.v6i3.802.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *26th International World Wide Web Conference, WWW 2017*, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182. doi: 10.1145/3038912.3052569.

[15] K. Tang, "Movie Recommendation System Based on Collaborative Filtering Network," *Applied and Computational Engineering*, vol. 96, no. 1, pp. 113–119, Nov. 2024, doi: 10.54254/2755-2721/96/20241441.

[16] T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster, "Recommender system based on pairwise association rules," *Expert Syst. Appl.*, vol. 115, pp. 535–542, Jan. 2019, doi: 10.1016/j.eswa.2018.07.077.

[17] B. Smyth, K. McCarthy, J. Reilly, D. O'Sullivan, L. McGinty, and D. C. Wilson, "Case Studies in Association Rule Mining for Recommender Systems.," in *IC-AI*, 2005, pp. 809–815.

[18] U. Wanaskar, S. Vij, and D. Mukhopadhyay, "A hybrid web recommendation system based on the improved association rule mining algorithm," *arXiv preprint arXiv:1311.7204*, 2013.

[19] E. Kannout, M. Grodzki, and M. Grzegorowski, "Towards Addressing Item Cold-Start Problem in Collaborative Filtering by Embedding Agglomerative Clustering and FP-Growth into the Recommendation System," *Computer Science and Information Systems*, vol. 20, no. 4, pp. 1343–1366, Sep. 2023, doi: 10.2298/CSIS221116052K.

[20] Z. Sheng, "Research on Personalized Movie Recommendation System Based on Collaborative Filtering," *Highlights in Science, Engineering and Technology AMMMP*, vol. Volume 140, pp. 72–77, 2025.

[21] M. J. Awan *et al.*, "A recommendation engine for predicting movie ratings using a big data approach," *Electronics (Switzerland)*, vol. 10, no. 10, May 2021, doi: 10.3390/electronics10101215.

[22] H. Fulzele, M. Bhoite, P. Kanfade, and A. Yadav, "Movie Recommender System using Content Based and Collaborative Filtering," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 5, 2023, [Online]. Available: www.ijisrt.com

[23] Z. Wang, "A content-based collaborative filtering algorithm for movies and TVS recommendation," *Applied and Computational Engineering*, vol. 15, no. 1, pp. 83–91, Oct. 2023, doi: 10.54254/2755-2721/15/20230812.

[24] A. R. Polu, B. Narra, D. V. K. R. Buddula, H. H. S. Patchipulusu, N. Vattikonda, and A. K. Gupta, "Evaluating Machine Learning Approaches for Personalized Movie Recommendations: A Comprehensive Analysis," *International Journal of Innovative Research in Multidisciplinary Education*, vol. 3, no. 12, 2024.

[25] A. M. A. Al-Sabaawi, M. H. Hussein, and M. Dalli, "Classifying Items With the Rating Values 3 Using Text Reviews to Improve the Recommendation Accuracy in the Collaborative Filtering Approach," *Karbala International Journal of Modern*

*Science*, vol. 11, no. 1, pp. 144–153, 2025, doi: 10.33640/2405-609X.3393.

[26] V. Lakshmi Chetana and H. Seetha, "CF-AMVRGO: Collaborative Filtering based Adaptive Moment Variance Reduction Gradient Optimizer for Movie Recommendations," *International Journal of Computers and Applications*, vol. 44, no. 11, pp. 1015–1023, 2022, doi: 10.1080/1206212X.2022.2097769.

[27] R. P. Achuthananda, N. Bhumeka, R. B. Dheeraj Varun Kumar, S. P. Hari Hara, and V. Navya, "Evaluating machine learning approaches for personalized movie recommendations: A comprehensive analysis," *J Contemp Edu Theo Artific Intel: JCETAI-115*, 2024.

[28] S. Peng, S. Siet, S. Ilkhomjon, D. Y. Kim, and D. S. Park, "Integration of Deep Reinforcement Learning with Collaborative Filtering for Movie Recommendation Systems," *Applied Sciences (Switzerland)*, vol. 14, no. 3, Feb. 2024, doi: 10.3390/app14031155.

[29] D. K. Behera, M. Das, S. Swetanisha, and P. K. Sethy, "Hybrid model for movie recommendation system using content K-nearest neighbors and restricted boltzmann machine," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 445–452, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp445-452.

[30] P. Naidu, P. Gaikwad, A. Kaundinya, S. Gajbhiye, and S. S. Kale, "Movie Recommendation Using CNN," *International Research Journal of Engineering and Technology*, 2022, [Online]. Available: www.irjet.net

[31] A. N. AbdulAmeer and M. H. Hussein, "Alleviating the User Cold-Start Problem in Recommendation Systems Based on Textual Reviews Using Deep Learning," *Iraqi Journal of Science*, pp. 7276–7286, 2024.

[32] L. Kang and Y. Wang, "Efficient and accurate personalized product recommendations through frequent item set mining fusion algorithm," *Heliyon*, vol. 10, no. 3, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25044.

[33] A. Ez-Zahout, H. Gueddah, A. Nasry, R. Madani, and F. Omary, "A hybrid big data movies recommendation model based k-nearest neighbors and matrix factorization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 434, pp. 434–441, 2022, doi: 10.11591/ijeecs.v26.i434.pp434-441.

[34] S. Kumar, P. Singh, G. Srivastava, and S. Singh, "INTELLIGENT MOVIE RECOMMENDER FRAMEWORK BASED ON CONTENT- BASED & COLLABORATIVE FILTERING ASSISTED WITH SENTIMENT ANALYSIS," *International Journal of Advanced Research in Computer Science*, vol. 14, no. 03, pp. 108–113, Jun. 2023, doi: 10.26483/ijarcs.v14i3.6979.

[35] Muhammad Luqman, Amir Yaqoob, Majid Bashir Ahmad, and Kanza Majid, "Sentiment Analysis to Predict Movies Success Rate Based on NLTK Movie Review Corpora Using Machine Learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 46–54, Jan. 2023, doi: 10.32628/cseit239013.

[36] Mohammed M. Sultan, "Network-Based Movie Quality Prediction and Recommendation System Using Hybrid Machine Learning Techniques," *Scientific Research Journal of Medical Sciences*, vol. 05, no. 01, pp. 1–8, Jun. 2025, doi: 10.47310/srjms.2025.v05i01.016.

[37] A. Nousheen, F. Naaz, A. Sadaff, and S. H. Naaz, "Film Flare : Movie Recommendation System Using Machine Learning," vol. 7, no. 2, pp. 26–35, 2025.

[38] A. Sami, W. El Adrousy, S. Sarhan, and S. Elmougy, "A deep learning based hybrid recommendation model for internet users," *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-79011-z.

[39] A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review,Techniques and

Trends," Jun. 2016, [Online]. Available: http://arxiv.org/abs/1606.01042

[40] B. Hazela, P. Asthana, S. Singh, and G. Srivastava, "Enhance Movie Recommender system using Machine Learning techniques," *2nd International Conference on "Advancement in Electronics & Communication Engineering*, pp. 14–15, 2022, [Online]. Available: https://ssrn.com/abstract=4159215