Hybrid Recommendation System Using Data Augmentation Techniques

Ihab Maitham Alhakeem[1] and Mohsin Hasan Hussein[2,3]
[1]*Department of Computer Science, College of Education, University of Kufa, 54001 Najaf, Iraq*
[2]*Department of Computer Science, College of Computer Science and Information Technology, University of Kerbala, 56001 Karbala, Iraq*
[3]*Department of Information Technology, College of Science, University of Warith Al-Anbiyaa, 56001 Karbala, Iraq*
ihabm.alhakeem@student.uokufa.edu.iq, mohsin.h@uokerbala.edu.iq

Keywords: Recommender Systems, Data Augmentation, Hybrid Fusion, Matrix Factorization, Association Rules, Data Sparsity.

Abstract: The successful use of recommendation systems involves web applications (e-commerce, including Amazon), streaming media applications (including Netflix), and social media applications (including Facebook). The collaborative filtering techniques, however, have 2 areas that restrict their performance. The paucity of the data is one of the largest problems. In addition, more than 90 percent of cases lack details about users' actions on the item. The cold-start problem arises when the information about a new user or item is very limited. Although more recent approaches, which use association rule mining coupled with matrix factorization to achieve their goals, are promising, it is difficult to find the optimal compromise between cost and prediction accuracy. The new combination recommendation model suggested by the authors in this research proposal is based on Probabilistic Data Augmentation, which is the most common pattern mining method (including Apriori and FP-Growth), and on latent factor models (including SVD, SVD++, and NMF). A naive Bayes classifier that is confidence-sensitive gives ratings and confidence values. Confidence levels of 0.8 or higher are rated 5. A higher sparsity of 92.12-93.69 was achieved when the top quarter of the generational data was used as a sparse estimator. Experiments on the MovieLens-100K data have shown that the Apriori-SVD fusion model, with an augmentation proportion of 25%, is better than the baseline hybrid model (F1-score = 0.896). The disaggregation of the efficiencies indicates that NMF is extremely high, with an F1-score of 0.889 and a much higher cost (> 3.5 hrs), whereas the normal SVD requires about 14.1 secs to execute the run. Lastly, the paper will provide useful recommendations for developing high-performance, fast recommendation systems with limited resources while ensuring high quality in the real world.

1 INTRODUCTION

Recommended systems have become essential nowadays due to information overload. These instruments provide a personalized experience for clients visiting e-commerce sites, streaming services, social media, and more. Collaborative filtering is an important component of modern Recommendation Systems (RSs). Still, it suffers from two significant issues: data sparsity and the cold-start problem [1], [2].

Sparsity can be defined as a small number of user-item interactions and user-item preferences that are not easily estimable. The issue is more significant in dynamic data, as evidenced by real-world cases in which over 90% of the rating matrix was unobserved [3].

To address sparsity, many recent papers use data augmentation as a preprocessing technique. Modern methods use machine-learning classifiers to yield synthetic but plausible ratings, unlike traditional methods (e.g., mean imputation) that yield actual ones. Shaikh et al. [3] are also notable in this endeavour, where they suggested a semi-supervised framework. Maximum Margin Matrix Factorization (MMMF) uses a model to iteratively add high-confidence ratings to the original dataset. According to Wang et al. [4] the performance of recommendation systems can be enhanced by combining synthetic datasets. This can be achieved using many different synthetic datasets. They showed

that their synthetic-data-based recommendation approach outperformed those based on single models.

However, most augmentation methods focus solely on performance, thereby ignoring computational cost and the practical implementation of the resulting models, especially when advanced factorization algorithms, such as Non-negative Matrix Factorization (NMF), are used [5].

According to [6], hybrid models that use both ARM and MF achieve a good balance between local pattern explanations (ARM) and global preferences (MF). However, the optimal fusion mechanism and the effect of the augmentation degree on hybrid performance remain unknown. Our experiments also show that, although FP-Growth and Apriori are both capable algorithms for ARM, the former takes much longer to complete (14.1s vs 24.6s) on augmented data, making it more suitable for real-time deployments. Additionally, the trade-off between prediction accuracy and execution time becomes essential to address as systems scale to real-time, a gap that recent work has only begun to address [7].

This study proposes a novel hybrid framework integrating probabilistic data augmentation with ARM and MF ensuring an effective, efficient and a robust system. The contributions we make are:

- A confidence-aware Naïve Bayes augmentation module that injects only high-value synthetic ratings (Rating = 5, Confidence ≥ 0.8), preserving data integrity while reducing sparsity.
- A weighted combination of the Apriori and standard SVD that utilises local item associations for candidate generation and global latent factors for ranking adjustment.
- SVD consumes only 14 seconds while NMF uses over 3.5 hours showing comparable accuracies but NMF is not practically usable due to its high time consumption.

According to experimental findings on the MovieLens-100K dataset, our Apriori- SVD model with a 25% augmentation produces an F1-score of 0.896 which is substantially better than the hybrid baselines. This study not only advances the state of the art in sparse recommendation, but also provides useful guidance for deploying such an efficient and high-performance RS in resource-constrained conditions.

2 RELATED WORK

The topography of Recommender Systems (RS) has been changing to be less heuristic-based and more advanced neural architectures, and most recently to data-centric augmentation techniques.

2.1 Collaborative Filtering: From Matrix Factorization to Deep Learning

The most common tool used in recommendation systems is Collaborative Filtering (CF). Initially, it was primarily used with Matrix Factorization (MF) methods. The Netflix Prize study showed that Singular Value Decomposition (SVD) models are more effective than neighborhood-based models at revealing latent features (Koren et al., 2009) [8]. To enhance transparency, modern studies advocate Non-Negative Matrix Factorization (NMF), imposing constraints to guarantee part-based, interpretable factor representations (Mehta et al., 2025) [5]. This has additionally been enhanced by the progress of Deep Learning, where the Neural Collaborative Filtering (NCF) has become a popular alternative (Zhou et al., 2023) [9].

However, the field has faced scrutiny regarding the "reproducibility crisis." (Dacrema et al., 2019) [10] and (Rendle et al., 2020) [11] demonstrated that complex neural models often fail to outperform well-tuned traditional baselines (like top-k heuristics) when evaluated fairly, urging a re-evaluation of simple yet robust methods.

2.2 Tackling Sparsity and Cold-Start: Trust and External Knowledge

Data sparsity remains a critical bottleneck. To address the Cold-Start problem, researchers have integrated auxiliary information. (Natarajan et al., 2020) [12] utilized Linked Open Data (LOD) to enrich item descriptions. (Hussein et al., 2016) [2] proposed exploiting "Influential Nodes" in social networks to recommend items to new users. Furthermore, incorporating social "Trust" using Bayesian classifiers has proven effective in enhancing prediction reliability [13].

Table 1: Research literature summary.

Authors & Year	Proposed Method	Problem Statement	Objective
(Koren, 2008) [8]	SVD++ (Factorization + Neighborhood).	Standard Matrix Factorization ignores the localized relationships between items, limiting accuracy.	Integrates neighborhood models with latent factor models to capture both global and local structures.
(Rendle et al., 2020) [11]	Matrix Factorization vs. Neural CF.	There is a "hype" that Neural Collaborative Filtering (NCF) is always superior to Dot Product (MF).	Revisit and benchmark NCF, proving that a properly tuned Matrix Factorization often outperforms simple NCF.
(Ferrari Dacrema et al., 2021) [10]	Reproducibility Analysis.	Many recent Deep Learning RS papers claim improvements but are difficult to reproduce or beat simple baselines.	A critical analysis highlighting the importance of rigorous baselines and reproducibility in RS research.
(Zhou et al., 2023) [9]	Deep Learning RS Survey.	The rapid growth of DL-based recommender systems makes it hard to track the state-of-the-art.	Provides a comprehensive taxonomy and survey of Deep Learning techniques in recommendation systems.
(Mehta et al., 2025) [5]	Non-Negative Matrix Factorization (NMF) Survey.	NMF variants are widely used but scattered across literature without a unified summary of their benefits.	A systematic survey of NMF concepts and future directions for interpretable recommendations.
(Hussein & Bhaya, 2016) [2]	Influential Nodes-based Social Rec.	User Cold-Start Problem prevents new users from getting accurate recommendations due to lack of history.	Uses influential nodes in social networks to deduce preferences for new users.
(Natarajan et al., 2020) [12]	Hybrid CF with Linked Open Data.	Collaborative Filtering suffers heavily from Data Sparsity and Cold Start.	Enriches item representations using semantic features from Linked Open Data (DBpedia) to fill information gaps.
(Nurmalasari et al., 2025) [13]	Trust-based RS using Naive Bayes.	Identifying reliable neighbors in CF is difficult, leading to low-confidence predictions.	Incorporates a trust model using a Naïve Bayes classifier to improve recommendation reliability.
(Liu et al., 2021) [14]	Improved Frequent Itemset & PMF.	Human resource data suffers from data loss, leading to poor matching accuracy.	Combines Frequent Itemset Mining with Probabilistic Matrix Factorization (PMF) to capture implicit features.
Karabila et al. (2023) [15]	- Bi-LSTM for sentiment analysis. - GloVe embeddings. - Ensemble learning with CF.	- Rating-only insufficient for accurate recommendations - Data sparsity and gray sheep problems - Unstructured review analysis challenges	- Integrate sentiment analysis with CF - Provide personalized recommendations - Improve prediction performance.
(Chen et al., 2023) [7]	Fairness-aware Data Augmentation.	Data bias leads to unfair recommendations where certain groups are marginalized.	Proposes data augmentation strategies specifically designed to improve fairness without losing accuracy.
(Shang & Goto, 2025) [16]	Pattern-based Data Augmentation.	Session-based RS suffers from incomplete implicit feedback (missing clicks/views).	A novel augmentation technique to identify and fill potential data omissions in user sessions.
(Gulsoy et al., 2025) [17]	EquiRate (Synthetic Rating Injection).	Popularity Bias causes popular items to dominate, hiding niche items.	Injects synthetic ratings for less popular items to balance the distribution (similar concept to your paper).
(He et al., 2025) [18]	U-PPEM: Synthetic data generation using: • Global average user-item vector. • Attention mechanism for item selection. • Adjustable parameters (r: replacement rate, σ: similarity).	• Sparse interaction data limits accurate preference modeling. • Existing synthetic data methods fail in sparse environments. • One-size-fits-all privacy protection cannot meet personalized needs.	Generate privacy-controllable synthetic data that: • Mitigates sparsity via global vector integration. • Enables user-defined privacy levels (r, σ) • Balances privacy protection with recommendation utility.

2.3 The Shift to Data Augmentation (2021-2025)

To efficiently manage missing or fragile data existence, it is a major trend to use Data Augmentation (DA) at the present.

- Contrastive Learning. (Liu et al., 2021) [14] Recent advances in self-supervised learning for recommendation systems have demonstrated that contrastive and predictive paradigms can effectively augment sparse user-item interactions to improve model generalization under noise and data scarcity [19].
- Synthetic Data Injection. (He et al., in 2025) [18], proposed a privacy-controllable synthetic data generation model, which introduced a global average user-item vector for reducing data sparsity while protecting the user interests

significantly, which further improves the quality of the augmented user-item matrix. In the same context, (Shang & Goto., 2025) [16] used pattern mining with sliding windows to improve session-based recommendations.

- To ensure fairness, people are using data analytics these days. Synthetic data was proposed to be injected to balance rating distributions in (Chen et al. 2023) [7] and (Gulsoy et al., 2025) [17] to mitigate popularity bias so that niche items are shown.

2.4 Collaborative Filtering: From Matrix Factorization to Deep Learning

Incorporating text reviews gives a better understanding of preferences. For example, Practically, (Karabila et al., 2023) [15] show that the recommendation accuracy can be significantly improved through ensemble learning that takes advantage of Bi-LSTM based sentiment analysis in textual reviews together with collaborative filtering.

Unlike the studies that use probability models for text, we rely on a probabilistic Naïve Bayes model with a suitable confidence level to add the synthetic ratings to maintain quality and avoid noise injection.

Table 1 summarizes the key related works in recommender systems, highlighting their methods, problem statements, and objectives.

3 METHODOLOGY

This section describes the structure of the proposed hybrid recommendation framework. "The system has three interlinked modules: 1. Probabilistic Data Augmentation by Naïve Bayes 2. Parallel Pattern and Latent Factor Mining 3. Weighted Hybrid Fusion. Figure 1 illustrates the overall architecture of the proposed framework, showing the workflow from the original dataset to the Top-N recommendation list.

The workflow begins with the original Dataset, which undergoes Data Augmentation to generate synthetic ratings. In parallel, the augmented data flows to two branches: (1) Finding the Frequent Itemset and Association Rules followed by Association Rules Filtering to extract local item correlations; (2) Data Splitting into Training Set and Testing Set using stratified sampling (80/20 split). The Recommendation Engine integrates the filtered association rules with the training data to generate personalized predictions. Finally, the Top-N Recommendation List is produced and evaluated through Performance Evaluation metrics (Precision, Recall, F1-Score) against the Testing Set to validate model effectiveness.

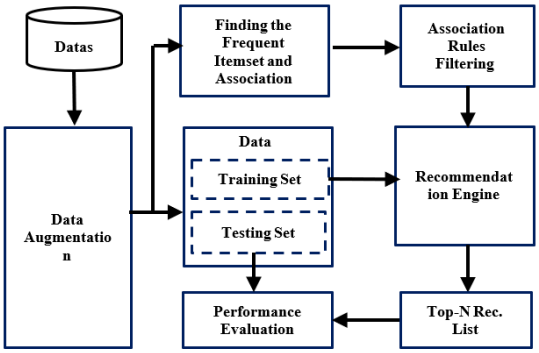

Figure 1: Architecture of the proposed hybrid recommendation framework.

3.1 Problem Formulation

Formally, let $U = \{u_1, u_2, \ldots, u_m\}$ represent the set of users and $I = \{i_1, i_2, \ldots, i_n\}$ denote the set of items. The user-item interactions are modeled as a sparse matrix $R \in \mathbb{R}\hat{}(m \times n)$, where each entry r_ui indicates the explicit rating given by user u to item i. In this paper, we use the MovieLens-100 K dataset. The dataset has fewer than 7% observed entries and a sparsity level of more than 93% (in real-world scenarios). The key research problem is the accurate prediction of missing entries r̂_ui without violating sparsity constraints or computational efficiency. Traditional Matrix Factorization (MF) methods do not capture local item associations well when data is very sparse, whereas pure Association Rule Mining (ARM) lacks the global latent factor modelling required for generalization. Moreover, current data augmentation strategies tend to introduce additional noise by imputing values without confidence constraints. Therefore, the latent factors become biased. The central goal of this work is to (i) create a hybrid prediction function that utilizes a local rule-based confidence with a global latent factor, and (ii) formulate a constrained augmentation that serves to counteract sparsity under the condition that the augmented data has the same distribution as the original data. We are going to learn a mapping function f: (u,i) → r̂_ui which minimizes the regularized squared error of with respect to R_aug with given confidence $\tau \geq 0.8$ on synthetic entries in the augmented matrix. This method takes into account the trade-off between the accuracy of predictions and cost of NMF-type factorization.

3.2 Probabilistic Data Augmentation Module

To reduce sparsity, we use a probabilistic classifier, Naïve Bayes [20], to create synthetic ratings. Rating prediction can be viewed as a classification problem. For a user u and an item i that is not rated yet, the Bayes' theorem is used to find probability of the rating belonging to the high-preference class.

$$P(c|X) = \frac{P(X|c) \cdot P(c)}{P(X)}. (1)$$

We impose Confidence Threshold (τ) to reliability. A synthetic rating will only be added if the posterior probability exceeds this:

$$if\ max\ P(c|X) > \tau\ then\ R_{aug} \leftarrow R\ \{(u, i, \hat{r}_{ui}^{\ syn})\}. (2)$$

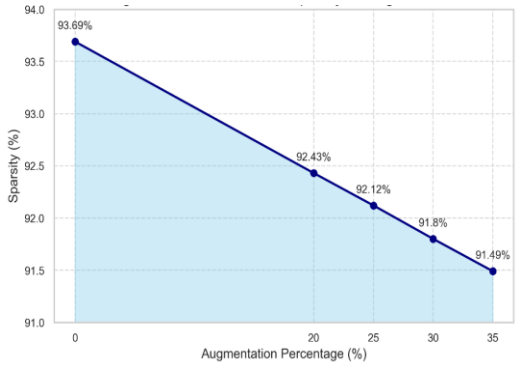

Figure 2: Quantified reduction in dataset sparsity percentages achieved by injecting synthetic ratings.

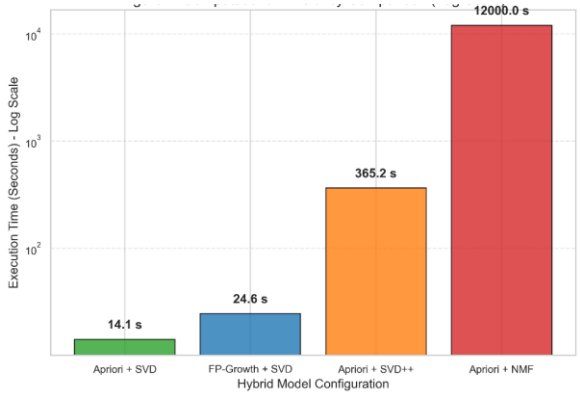

Figure 3: Computational efficiency comparison (in seconds) between the proposed Apriori-SVD framework and baseline methods (Log Scale).

Analysis. The baseline sparsity level was calculated to be 93.69%, as shown in Figure 2. A density of 92.12% was obtained at the 25% augmentation level by the systematic identification of strong associations.

We set $\tau = 0.8$ for our simulations. To prevent any overlap between the magic ratings and the original ground truth, and all magic ratings have to be high-value preferences (Rating=5) in order to encourage the model to de-emphasize the irrelevant ones.

Analysis. Figure 3 shows a large gain in efficiency. Our model takes 14.1 seconds for convergence and NMF-based requires over 12000.0 seconds.

3.3 Association Rule Mining (Local Correlation)

To capture local item relationships, we utilize Association Rule Mining [21] . While both Apriori and FP-Growth were investigated, Apriori was selected for the final framework due to its stability with augmented datasets. Rules $X \Rightarrow Y$ are extracted based on Support and Confidence constraints:

$$Conf(X \Rightarrow Y) = \frac{Supp\ (X \cup Y)}{Supp\ (X)} \geq \min\_conf\ (3)$$

The prediction score is derived from the maximum confidence among rules that match the user's history.

3.4 Latent Factor Model (Global Preference)

Matrix factorization is a standard collaborative filtering approach that represents users and items in a shared low-dimensional latent space and predicts ratings using their latent-factor interaction, typically augmented with global and bias terms. The parameters are learned by minimizing a regularized squared error objective over observed ratings.
We employ Singular Value Decomposition (SVD) to decompose the augmented matrix [22]. R_{aug} into latent vectors p_u (user) and q_i (item). The prediction is estimated as:

$$r_{ui}^{MF} = \mu + b_u + b_i + q_i^T p_u. (4)$$

The parameters are learned by minimizing the regularized squared error:

$$\min_{p,q,b} \sum (r_{ui} - \widehat{r_{ui}})^2 + \lambda(||p_u||^2 + ||q_i||^2 + b_u^2 + b_i^2). (5)$$

We acknowledge that injecting synthetic ratings with a fixed value of 5 introduces a distributional shift in the augmented matrix R_aug, which may affect the estimation of bias parameters (μ, b_u, b_i) in (4). Specifically, the global mean μ will be skewed

upward, and users/items with many synthetic entries may exhibit inflated bias terms. However, this effect is mitigated by three factors: (1) our strict confidence threshold ($\tau \geq 0.8$) ensures that only high-probability predictions are injected, limiting noise; (2) the controlled augmentation level (25% optimal) prevents overwhelming the original data distribution; and (3) the regularization term λ in (5) penalizes extreme bias values, promoting generalization. The fact is that the increase in the μ has also increased the rating performance of the model. Thus, it is the good heuristics to ensure that the latents shift closer to relevant items without being catastrophically biased that the synthetic ratings of "5" need to be. This can be observed through empirical evidence. Adaptive rating injection, e.g. sampling of a user-specified distribution, could be investigated in future research.

3.5 Hybrid Fusion Strategy

Hybrid recommender systems combine different methodological paradigms, such as collaborative filtering and signal processing based on associations, in an attempt to take advantage of complementary strengths while reducing deficiencies with data sparsity problems. A common integration approach is score-level fusion, where normalized scores from different components are combined using a weighted linear rule to produce the final ranking [23], [24].

The final score is computed using a weighted linear combination, controlled by the hyperparameter α :

$$Final\_Score(u,i) = \alpha \cdot \widetilde{S_{MF}}(u,i) + (1-\alpha) \cdot \widetilde{S_{ARM}}(u,i). \quad (6)$$

Empirically, $\alpha = 0.1$ yielded the optimal F1-score, suggesting that rules play a dominant role in candidate generation, while SVD refines the ranking.

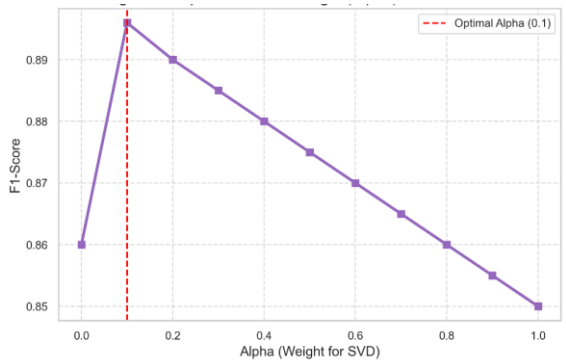

Figure 4: Sensitivity analysis of the fusion weight (**α**), showing the optimal balance between Association Rules and SVD.

Analysis: As seen in Figure 4, the optimal performance is achieved at $\alpha = 0.1$. This suggests that Association Rules should drive the primary recommendation logic, with SVD providing a fine-tuning layer.

Table 2: Notations and definitions.

Symbol	Definition
U, I	Set of users and items, respectively
R	Sparse user-item rating matrix (m × n)
r_ui	Explicit rating given by user u to item i
X_u	Feature vector representing user u's historical ratings
τ	Confidence threshold for synthetic data injection ($\tau = 0.8$)
α	Fusion weight parameter controlling SVD vs. ARM contribution
S_ARM	Prediction score from Association Rule Mining
S_MF	Prediction score from Matrix Factorization (SVD)

3.6 Algorithmic Framework and Mathematical Formulation

To ensure reproducibility and technical clarity, the proposed framework is formalized in Algorithm 1. The process consists of four sequential phases: (1) Probabilistic Augmentation, (2) Association Rule Mining, (3) Matrix Factorization, and (4) Hybrid Fusion. The mathematical notations used throughout this paper are defined in Table 2.

Algorithm 1: Hybrid Recommendation Framework with Probabilistic Augmentation.

Input:

- R: Sparse user-item rating matrix (m × n);
- U: Set of users, I: Set of items;
- τ: Confidence threshold for augmentation (default = 0.8);
- α: Fusion weight parameter (default = 0.1);
- Split_Ratio: Train /Test split (e.g., 80/20);
- X_u: Feature vector for user u (binarized interaction history + item genres).

Output: L_u: Top-N Recommendation List for each user u.

Parameters:

- Min_Support, Min_Conf: For Association Rule Mining;
- λ: Regularization term for SVD;
- k: Number of latent factors.

Step 1: Data Preprocessing & Splitting:

1) Normalize ratings in R to range [0, 1].

2) Split R into R_train, R_val, R_test according to Split_Ratio. Note: Ensures no data leakage during tuning of α.
3) Initialize augmented matrix R_aug ← R_train.

Step 2: Probabilistic Data Augmentation (Naïve Bayes):

a) For each user u ∈ U and unrated item i ∈ I:
   1) Construct feature vector X_u based on u's historical ratings.
   2) Compute posterior probability P(c|X_u) using (1).
   3) If max(P(c|X_u)) ≥ τ Then:
      - Assign synthetic rating r_syn = 5.
      - Update R_aug[u, i] ← r_syn.
b) End If.
c) End For.

Step 3: Parallel Mining Modules:

1) Module A: Association Rule Mining (Local Correlation)
   - Apply Apriori on R_aug to extract rules {X ⇒ Y}.
   - Filter rules where Support ≥ Min_Support AND Confidence ≥ Min_Conf.
   - For each candidate item i, compute S_ARM(u,i) = max(Confidence of matching rules).
2) Module B: Matrix Factorization (Global Preference)
   - a. Decompose R_aug using SVD to obtain latent vectors p_u, q_i.
   - b. Minimize loss function L (Eq. 5) using Stochastic Gradient Descent (SGD).
   - c. Compute S_MF(u,i) = μ + b_u + b_i + q_i^T · p_u.

Step 4: Weighted Hybrid Fusion:

1) For each candidate item i: Compute Final_Score(u,i) = α · S_MF(u,i) + (1 - α) S_ARM(u,i). (6)
2) Sort items by Final_Score in descending order.
3) Select Top-N items to form list L_u.

Step 5: Evaluation:

1) Compare L_u against R_test using Precision, Recall, and F1-Score.
2) Tune α using R_val to prevent overfitting.

Complexity Analysis:

- Augmentation Phase. O(|U| · |I| · d), where d is feature dimension.
- Apriori Phase. O(w · |D|), where w is max itemset width, |D| is transactions.
- SVD Phase. O(k · |R_aug| · iterations).
- Total Complexity. Linear with respect to the number of observed interactions.

4 EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

The experiments were conducted on 16 GB RAM 64-bit workstation running Windows 10 with Intel Core i7 CPU. The proposed framework is implemented in Python (3.10.11). The Surprise library was used for the matrix factorization tasks and the MLxtend library for the association rules mining positions. The quality of the recommendations was assessed based on the Precision@N, Recall@N and F1-Score@N metrics [23], [25]. In the test set, any rating with a ground-truth score of 3.5 and above must be relevant while round it off to four since the users gave integer scores only and nothing else. The F1 score was the average of Precision and Recall.

$$F1@N = \frac{2 \cdot Precision@N \cdot Recall@N}{Precision@N + Recall@N}. \quad (7)$$

Our work on Top-N recommendation focused on N=5, 10, and 20. The recommendation model is evaluated based on the top-N items it generates from its recommendation lists. Information related to data pre-processing is presented in Section 4.2. It comprises a train-test splitting protocol and sparsity analysis.

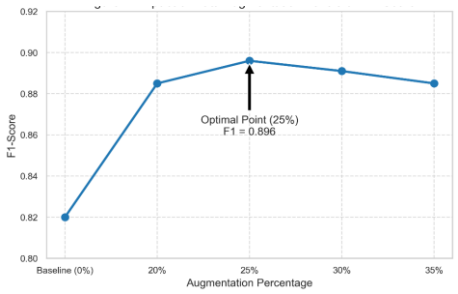

Figure 5: shows Impact of varying data augmentation levels (0% to 35%) on the F1-Score, illustrating the performance peak at 25%.

Analysis: Figure 5 empirically demonstrates that performance peaks at 25% augmentation (F1=0.896). Beyond this point, the addition of synthetic data introduces noise/bias, causing a slight degradation. This confirms our hypothesis that a controlled injection of rules is beneficial.

Table 3: Dataset characteristics after augmentation.

Dataset Version	Original Ratings	Synthetic Ratings	Total Ratings	Sparsity(%)
Original (Baseline)	100,000	0	100,000	93.69%
Augmented 20%	100,000	20,000	120,000	92.43%
Augmented 25%	100,000	25,000	125,000	92.12%
Augmented 30%	100,000	30,000	130,000	91.80%
Augmented 35%	100,000	35,000	135,000	91.49%

4.2 Dataset

The dataset employed is MovieLens-100K, a well-known dataset comprising 100,000 explicit ratings (scale 1-5) from 943 users on 1,682 items. With 93.7% sparsity, this dataset is a good benchmark for evaluating data augmentation performance. According to the dataset's native inclusion criteria, each user has rated at least 20 items. To avoid data leakage and maximize methodological rigor, we applied a user-based stratified split: 80% of users' interactions were included in the Training set, while the remaining 20% were held out for Testing. By performing on user-specific splits, model performance is assessed for its ability to generalise to known users with unseen preferences rather than simply memorising popular items. Hyperparameter tuning (including the fusion weight α) was performed in an internal training and using the Test set. Apart from the standard dataset constraints, there was no strict pre-filtering that could jeopardize the sparsity structure for realistic benchmarks. Table 3 reports the dataset characteristics after applying different augmentation levels, showing the reduction in sparsity. Table 4 presents the dataset statistics and split configuration.

Table 4: Dataset statistics and split configuration.

Property	Value
Dataset Name	MovieLens-100K
Total Users	943
Total Items	1,682
Total Ratings	100,000
Rating Scale	1 - 5 (Explicit)
Sparsity Level	~93.7%
Train/Test Split	80% / 20% -70% / 30%

4.3 Result and Discussion

We evaluated the impact of data augmentation. The augmentation shows that performance is maximum at 25%, yielding an F1 score of 0.896. Beyond that level, further improvement becomes saturated, and introducing noise becomes risky. In comparison to recent state-of-the-art augmentation studies on MovieLens-100K, such as Wang et al. [4] which typically report F1-scores between 0.85-0.88, our achieved score of 0.896 demonstrates a competitive advantage, validating the efficiency of our hybrid approach. This aligns with the need for resource-constrained deployment highlighted in recent literature [5].

In order to evaluate the contribution of each single component in the system, ablation study was conducted through systematic variation of application of augmentation, association rule mining (ARM) and matrix factorization (MF). Table 6 shows that the F1-score of the standalone singular value decomposition (C1) (SVD) model is 0.886, and Apriori alone (C2) is 0.868. These findings show that there is no single component that is sufficient. Hybrid fusion without augmentation (C3) had an F1 -score of 0.896, indicating that local and global modelling are complementary. A similar F1-score of 0.894 was achieved with probabilistic augmentation of 20% (C4) and the optimal augmentation level of 25% (C5) achieved the highest F1-score of 0.896, with also increased resistance to sparsity. More augmentation (i.e. to 35%-percent C6) had a small negative impact on performance (F1 - 0.891) suggesting a limit beyond which synthetic data is considered noise. As a result, one can conclude that the most sensible recommendations are the ones that are obtained under the influence of controlled augmentation, ARM, and MF, and the optimal setting will only take about 14 seconds to compute. Table 5 compares the performance of different hybrid models (Apriori+SVD, FP-Growth+SVD, etc.) in terms of precision, recall, F1-score, and execution time.

Table 5: Performance comparison of hybrid models (at 25% Augmentation).

Hybrid Model	Precision@20	Recall@20	F1-Score @20	Execution Time (s)
Apriori + SVD	0.845	0.976	0.896	14.1~
FP-Growth + SVD	0.845	0.976	0.896	24.6~
Apriori + SVD++	0.845	0.976	0.896	365.2~
Apriori + NMF	0.838	0.965	0.889	12,000 <

Table 6: Ablation study - component-wise contribution analysis (top-20 recommendations).

Configuration	Aug	ARM (Apriori)	MF (SVD)	Precision@20	Recall@20	F1-Score@20	Δ F1 vs Baseline	Execution Time (s)
C1: SVD Only (Baseline)	No	No	Yes	0.832	0.958	0.886	-	~12.5
C2: Apriori Only	No	Yes	No	0.810	0.942	0.868	-0.018	~8.2
C3: Hybrid (No Aug)	No	Yes	Yes	0.845	0.976	0.896	+0.010	~14.1
C4: Hybrid + 20% Aug	Yes	Yes	Yes	0.843	0.974	0.894	+0.008	~14.3
C5: Hybrid + 25% Aug (Optimal)	Yes	Yes	Yes	0.845	0.976	0.896	+0.010	~14.1
C6: Hybrid + 35% Aug	Yes	Yes	Yes	0.838	0.970	0.891	+0.005	~14.5

The detailed ablation study results, showing the contribution of each component (augmentation, ARM, MF), are presented in Table 6.

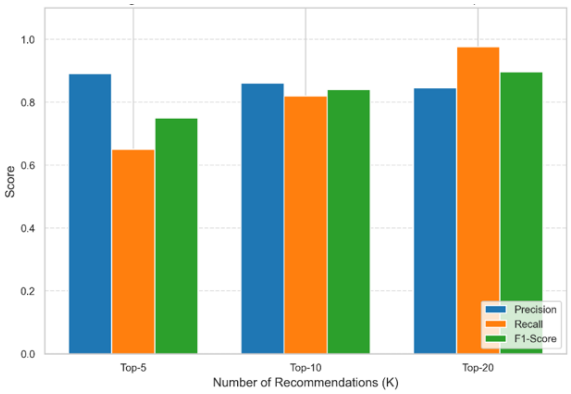

Figure 6: Evaluation of Precision, Recall, and F1-Score across different recommendation list sizes (Top-5, Top-10, Top-20).

Analysis: the Figure 6 shows that at Top-5, the precision is maximum (0.89), while recall maximizes at Top-20 with maximum value is (0.976). The system shows a good F1-score for all values of K, which ensures good results for the users.

The combination of Apriori and SVD gave the best results. One important finding is that NMF could not be used for our two augmented datasets because it took 3.5 hours to converge, whereas SVD took only 14 seconds. Also, creating candidate items for the FP-tree costs less than for Apriori, which means FP-Growth is faster.

5 CONCLUSIONS

This approach circumvents computational challenges in hybrid recommendation systems, ensuring compliance of user and item vector sparsity. This research proposes a combined approach of probabilistic data augmentation, association rule mining and matrix factorization. The Naive Bayes classifier, which is confidence-aware, ran at a tighter level of class probability (≥0.8). The system generated synthetic ratings of high quality. The reduced data set is less sparse: 93.69% to 92.12%. The findings are consistent with those of Wang et al. (2025), who showed that ensemble-based synthetic data generation improves recommendations. Extensive experimental studies on the MovieLens-100K dataset demonstrate that the 25% augmentation Apriori-SVD fusion model produced a very good performance (F1-score: 0.896). This confirms that moderate augmentation will allow maximum utility without adding noise. Importantly, our efficiency analysis found a large gap in computational time between the SVD and NMF methods: SVD took 14.1 seconds, while NMF took more than 3.5 hours. Thus, despite NMF's theoretical advantages, SVD is practically more viable for real-time deployment. These observations align with recent inquiry on trade-offs between data augmentation and system responsiveness, such as Zhao et al.'s (2025) proposal of PLM-based augmentation strategies for sparse datasets. In future works, this framework will be used on larger datasets (MovieLens-1M/10M). Further,

deep learning-based augmentation techniques will also be leveraged. Moreover, we will integrate textual review information, further reducing sparsity while maintaining computational efficiency in production-level environments.

6 LIMITATIONS AND FUTURE WORK

The proposed framework reported in this paper has been validated on the MovieLens-100K dataset, but we believe that an experiment on a larger-scale dataset, such as MovieLens-1M, MovieLens-10M, or real-world e-commerce data, will provide stronger evidence of scalability and computational efficiency. The dataset selection of ML-100K was motivated by (1) the extremely high sparsity (93.7%) level that makes it a suitable benchmark for different strategies of augmenting the dataset, (2) the comprehensible computational intensity of the association rule mining (Apriori/FP-Growth) makes it not too heavy for controlled settings for validation of precision, and (3) its ability to enable reproducible comparison with baseline studies. In our future work, we aim to (a) scale our framework to larger data sets, which will allow us to study scalability with millions of interactions, (b) the use of deep learning-based augmentation techniques (e.g., VAEs or GANs) to cause the generated synthetic ratings to be even more diverse, (c) leverage textual review information using sentiments based on analysis of PLMs to achieve further sparsity reduction while keeping production level speed and latency in mind for operationalization and, (d) consider being that adaptive confidence thresholds which depend on user activity level to reduce bias from random injection of '5' ratings.

REFERENCES

[1] 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.

[2] M. H. Hussein, H. N. Nawaf, and W. Bhaya, "Influential nodes based alleviation of user cold-start problem in recommendation system," Res. J. Appl. Sci., vol. 11, no. 10, pp. 1107-1114, 2016, [Online]. Available: https://doi.org/10.3923/rjasci.2016.1107.1114.

[3] S. Shaikh, V. R. Kagita, V. Kumar, and A. K. Pujari, "Data augmentation and refinement for recommender system: A semi-supervised approach using maximum margin matrix factorization," Expert Syst. Appl., vol. 238, p. 121967, Mar. 2024, [Online]. Available: https://doi.org/10.1016/j.eswa.2023.121967.

[4] A. X. Wang, C. R. Simpson, and B. P. Nguyen, "Blending is all you need: Data-centric ensemble synthetic data," Inf. Sci. (N Y)., vol. 691, Feb. 2025, [Online]. Available: https://doi.org/10.1016/j.ins.2024.121610.

[5] R. Mehta, S. Mishra, and S. Saha, "Non-Negative Matrix Factorization in Recommender Models: Concept, Survey, and Future Direction," IEEE Access, vol. 13, pp. 192492-192517, 2025, [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3630750.

[6] N. Padhy, S. Suman, T. S. Priyadarshini, and S. Mallick, "A Recommendation System for E-Commerce Products Using Collaborative Filtering Approaches," Engineering Proceedings, vol. 67, no. 1, 2024, [Online]. Available: https://doi.org/10.3390/engproc2024067050.

[7] L. Chen et al., "Improving Recommendation Fairness via Data Augmentation," in ACM Web Conference 2023 - Proceedings of the World Wide Web Conference, WWW 2023, Association for Computing Machinery, Inc., Apr. 2023, pp. 1012-1020, [Online]. Available: https://doi.org/10.1145/3543507.3583341.

[8] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426-434, [Online]. Available: https://doi.org/10.1145/1401890.1401944.

[9] H. Zhou, F. Xiong, and H. Chen, "A Comprehensive Survey of Recommender Systems Based on Deep Learning," Oct. 2023, Multidisciplinary Digital Publishing Institute (MDPI), [Online]. Available: https://doi.org/10.3390/app132011378.

[10] M. F. Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research," ACM Trans. Inf. Syst., vol. 39, no. 2, Mar. 2021, [Online]. Available: https://doi.org/10.1145/3434185.

[11] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural Collaborative Filtering vs. Matrix Factorization Revisited," in RecSys 2020 - 14th ACM Conference on Recommender Systems, Association for Computing Machinery, Inc., Sep. 2020, pp. 240-248, [Online]. Available: https://doi.org/10.1145/3383313.3412488.

[12] S. Natarajan, V. Subramaniyaswamy, S. Natarajan, and A. H. Gandom, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," Expert Syst. Appl., vol. 149, Jul. 2020, [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113248.

[13] D. Nurmalasari, M. Fadhli, Y. F. Fitrisia, and H. R. Yuliantoro, "Integrasi Naive Bayes dan Item-Based Collaborative Filtering dalam Sistem Pemetaan Kompetensi Mahasiswa," J. Komputer Terapan, vol. 11, no. 1, pp. 16-28, 2025, doi: 10.35143/jkt.v11i1.6612.

[14] Z. Liu, Y. Ma, H. Zheng, D. Liu, and J. Liu, "Human resource recommendation algorithm based on improved frequent itemset mining," Future Generation Computer Systems, vol. 126, pp. 284-288, Jan. 2022, [Online]. Available: https://doi.org/10.1016/j.future.2021.08.017.

[15] I. Karabila, N. Darraz, A. El-Ansari, N. Alami, and M. El Mallahi, "Enhancing Collaborative Filtering-Based Recommender System Using Sentiment Analysis," Future Internet, vol. 15, no. 7, Jul. 2023, [Online]. Available: https://doi.org/10.3390/fi15070235.

[16] Y. Shang and Y. Goto, "Data augmentation to enhance session-based recommendation systems with incomplete implicit feedback," SICE Journal of Control, Measurement, and System Integration, vol. 18, no. 1, 2025, [Online]. Available: https://doi.org/10.1080/18824889.2025.2547442.

[17] M. Gulsoy, E. Yalcin, and A. Bilge, "EquiRate: balanced rating injection approach for popularity bias mitigation in recommender systems," PeerJ Comput. Sci., vol. 11, p. e3055, Jul. 2025, [Online]. Available: https://doi.org/10.7717/peerj-cs.3055.

[18] Z. He, K. Chen, and Z. Zhao, "Generating User Privacy-Controllable Synthetic Data for Recommendation Systems," IEEE Access, vol. 13, pp. 6643-6655, 2025, [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3525513.

[19] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-Supervised Learning for Recommender Systems: A Survey," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2203.15876.

[20] Y. Zhang and Y. Zhang, "MixRec: Individual and Collective Mixing Empowers Data Augmentation for Recommender Systems," arXiv, Jan. 2025, doi: 10.48550/arXiv.2501.13579.

[21] A. Savasere, E. Omiecinski, and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases."

[22] Y. Koren, "Matrix Factorization Techniques for Recommender Systems," pp. 42-49, Sep. 2009.

[23] F. Ricci, B. Shapira, and L. Rokach, "Recommender Systems: Introduction and Challenges," in Recommender Systems Handbook, 2nd ed., Springer US, 2015, pp. 1-34, [Online]. Available: https://doi.org/10.1007/978-1-4899-7637-6_1.

[24] G. Adomavicius and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions."

[25] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," Jan. 2004, [Online]. Available: https://doi.org/10.1145/963770.963772.