Ensemble Learning Approach for Efficient Recommendation Systems Using Semi-Supervised Learning

Submitted: 7th August 2025; accepted: 4th November 2025

Nisha Sharma, Mala Dutta

DOI: 10.14313/jamris-2026-027

Abstract:
In recommender systems, collaborative filtering (CF) is a crucial technique, but it often struggles with data sparsity, which affects recommendation accuracy. To address this challenge, we have proposed a Co-Training Ensemble Learning (CTEL) technique that integrates item-based Collaborative Filtering (CF), user-based CF, and Singular Value Decomposition (SVD) via a structured stacking methodology to improve recommendation performance. The co-training procedure, which creates pseudo-labels for unlabeled data based on a confidence threshold, is used to iteratively improve the user-based and item-based CF models after they have been originally trained. These models produce predictions for validation and test sets, in conjunction with the independently trained SVD model. These forecasts yield meta-features, including additional statistical variables such as variance and the product of predictions. The Linear Regression model is trained as the meta-learner to optimise the predictions of the base models using K-Fold cross-validation. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used to assess the final model's performance on a test set. The outcomes confirm the effectiveness of the co-training and stacking strategy, demonstrating notable increases in prediction accuracy. By leveraging the advantages of collaborative filtering and matrix-based approaches, the proposed model provides a comprehensive foundation for developing advanced recommendation systems.

Keywords: *Recommender system, Ensemble learning, Collaborative filtering, SVD, Semi-Supervised learning*

1. Introduction

Recommender systems have become indispensable to digital ecosystems such as e-commerce, entertainment, education, and social media by enabling personalized interaction between users and massive content repositories [1, 2]. These systems enhance user engagement, retention, and satisfaction by analyzing behavioral data and generating tailored product, service, or content recommendations. Collaborative filtering (CF) remains the most widely used technique in this domain, as it effectively exploits patterns of collective user–item interactions to infer individual preferences. CF approaches are generally categorized into user-based and item-based models, which leverage similarities either among users or among items to predict unseen ratings or interactions. Despite their effectiveness, these methods are constrained by data sparsity, where insufficient user–item interactions limit the system's ability to make accurate predictions. This problem is particularly severe in large-scale systems with rapidly expanding catalogs, leading to the well-known cold-start challenge [3–5]. Addressing data sparsity has become a central research objective in recommendation science. Matrix-factorization techniques such as Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), and Probabilistic Matrix Factorization (PMF) have been instrumental in revealing latent user and item features that explain observed interactions [6–8]. These models improve scalability and interpretability; however, their linear structure often limits the ability to capture complex nonlinear relationships. The integration of deep learning into recommendation models has transformed the field, enabling neural collaborative filtering, autoencoders, and graph-based learning to uncover hierarchical and non-linear dependencies [9–11]. Deep architectures such as Neural Collaborative Filtering (NCF) and Variational Autoencoders (VAE) have proven effective in learning rich feature embeddings and temporal preferences that enhance prediction accuracy. In parallel, ensemble learning has emerged as a key strategy to improve robustness and generalization in recommender systems. Ensemble frameworks such as bagging, boosting, and stacking combine multiple weak or moderately strong learners to form composite models that outperform individual predictors [12]. Stacking employs a meta-learner to integrate diverse base models, while co-training allows multiple learners trained on different feature views to iteratively refine each other using pseudo-labels generated from unlabeled data [13]. These principles have inspired semi-supervised ensemble models that effectively exploit both labeled and unlabeled data. The proposed Co-Training Ensemble Learning (CTEL) framework builds upon this foundation by integrating user-based CF, item-based CF, and SVD models within a co-training and stacking paradigm. Through iterative pseudo-labelling and meta-feature construction, CTEL improves predictive accuracy even under severe sparsity. However, with the accelerating pace of research in 2025, new paradigms have emerged that extend beyond deterministic ensemble frameworks. Contemporary advances increasingly focus on incorporating epistemic uncertainty, monotonic transformation, and multi-intent behavioral diversity into ensemble recommendation models. Recent studies introduced Bayesian deep ensemble collaborative filtering frameworks that quantify epistemic uncertainty to improve robustness under noisy or sparse data [14]. Further developments integrated orthogonal meta-learning with Bayesian optimization to achieve uncertain multi-objective recommendation, balancing competing goals such as accuracy, fairness, and diversity [15]. Other advances proposed unified monotonic ranking ensembles employing Unconstrained Monotonic Neural Networks (UMNNs) to ensure monotonic score transformations and adaptive Pareto-optimal weighting, thereby eliminating manual tuning and ensuring interpretability [16]. Additional frameworks introduced unified representation-learning architectures that model user intent as Gaussian distributions, capturing both mean preferences and behavioral uncertainty [17]. These state-of-the-art approaches represent a shift toward uncertainty-aware and monotonic ensemble architectures that explicitly model confidence and ranking consistency in recommendation outcomes [18].

Against this evolving backdrop, CTEL distinguishes itself as a semi-supervised, deterministic ensemble model that emphasizes interpretability and computational efficiency. While it may be considered an incremental contribution relative to the Bayesian and monotonic paradigms, CTEL provides a practical bridge between classical CF-based learning and emerging uncertainty-aware ensemble frameworks. The approach remains particularly relevant for large-scale applications where model transparency, low latency, and adaptability are paramount. Furthermore, CTEL's architecture can be naturally extended to incorporate Bayesian regularization or monotonic fusion layers, aligning with ongoing efforts to create interpretable and reliable next-generation recommender systems. This research contributes to the evolving discourse on ensemble-based recommender systems by presenting a hybrid semi-supervised framework—CTEL—that addresses data sparsity through co-training and meta-learning integration. It complements recent innovations such as Bayesian deep ensembles and monotonic ranking transformations, laying the foundation for the future development of uncertainty-aware and explainable recommendation architectures.

Recent studies emphasize hybrid strategies integrating semi-supervised learning, ensemble learning, and uncertainty-aware modelling to exploit both labelled and unlabeled data while improving interpretability and robustness [19]. In addition, graph neural networks (GNNs) and attention-based transformers have expanded the representational depth of recommender models. GNNs propagate relational information through user–item interaction graphs, enabling higher-order connectivity learning, while transformers dynamically capture temporal and contextual cues to enhance sequential and session-based recommendation [20, 21]. Explainability and reliability are emerging as equally critical dimensions, with attention-based interpretability mechanisms [22] and reliability-aware hybrid models integrating epistemic uncertainty into recommendation pipelines. Furthermore, semi-supervised ensemble frameworks continue to evolve, leveraging agreement-driven pseudo-label generation and consistency regularization for large-scale sparse datasets.

2. Issues and Challenges

3. Methods and Materials

3.1. Datasets used for testing our proposed model

The MovieLens 100K dataset, which consists of 100,000 user ratings for 1,682 movies from 943 users, is a commonly used benchmark in the field of recommendation systems. The University of Minnesota's GroupLens Research Project gathered the dataset [46]. Every entry in the collection includes a timestamp, a user ID, a movie ID, and a rating representing the user's assessment of the film at that moment. These are not the only features; there are also additional features for user and movie information. Features for user information include age, gender, occupation, and Zip code, and features for movie information include movie ID, title, release date, video release date, IMDb URL, and genres. Predicting a user's movie rating based on their past ratings and other users' ratings is usually the aim. A description of the dataset's core features is provided in Table 3.

Another popular dataset for assessing recommendation systems is FilmTrust, which is gathered from the FilmTrust social network [47]. Users' movie evaluations and the trust ties between them are included in the dataset. It contains variables like ratings, item IDs (movies), and user IDs. By modelling and forecasting user preferences, these factors provide insights that can improve the accuracy of recommendations. The dataset's description is given in Table 4.

3.2. Collaborative algorithms

Collaborative filtering is a method used in recommendation systems to predict a user's interests by collecting preferences from many users. There are two main types of collaborative filtering algorithms: user-based and item-based. Additionally, matrix factorization techniques such as Singular Value Decomposition (SVD) are often used to improve the performance of collaborative filtering systems.

3.2.1. User-Collaborative approach (U-col)

This method predicts a user's interest in an item based on ratings from similar users. The similarity between two users, u and v, can be calculated using the Pearson correlation coefficient:

$$Sim\,(u,\,v) = \frac{\sum_{i\in Iuv}(r_{ui} - \underline{r}_u)(r_{vi} - \underline{r}_v)}{\sqrt{\sum_{i\in Iuv}(r_{ui} - \underline{r}_u\ )^2}\sqrt{\sum_{i\in Iuv}(r_{vi} - \underline{r}_v\ )^2}}$$

Table 1. Comparison of commonly used semi-supervised methods

Method	Description	Advantages	Applications
Co-training	Uses multiple models trained on different views of data to iteratively improve predictions	Effective use of unlabeled data, robustness to noisy data	Text categorization, recommender systems
Self-training	Iteratively labels unlabeled data using a model trained on labeled data	Simple and intuitive, easy implementation	Text classification, image recognition
Tri-training	Extension of Co-training with three classifiers, enhancing model robustness	Improved performance with three different views	Sentiment analysis, social network analysis
Label propagation	Propagates labels from labelled to unlabeled instances based on similarity	Utilizes local information effectively, scalable	Community detection, recommendation systems
Graph-based methods	Uses graph structure to propagate labels and capture relationships	Captures complex relationships, robust to noise	Social network analysis, recommendation systems
Semi-supervised SVM	Applies SVM with labelled and unlabeled data to learn decision boundaries	Utilizes margin maximization, effective for non-linear boundaries	Image recognition, text classification
Expectation Maximization	It iteratively estimates the parameters of a probabilistic model with hidden variables	Handles missing data, robust to noise	Clustering, anomaly detection
Transudative SVM	SVM variant that learns from both labelled and unlabeled data simultaneously	Utilizes unlabeled data for decision boundary optimization	Classification, pattern recognition
Generative Models	Models that generate data based on learned probability distributions	Provides insights into data distribution, scalable with large datasets	Data generation, anomaly detection
Semi-supervised Deep Learning	Deep learning models trained with both labelled and unlabeled data	Captures intricate patterns, effective for large-scale data	Natural language processing, image recognition

where r_{ui} is the rating of user u for item i, r_v is the average rating of user u, and I_{uv} is the set of items rated by both users u and v. The prediction for user u for item i is given by:

$$\hat{r}_{ui} = \underline{r}_u + \frac{\sum_{v \in N(u)} Sim(u,\ v)(r_{vi} - \underline{r}_v)}{\sum_{v \in N(u)} \left|Sim(u,\ v)\right|}$$

where N(u) is the set of top k similar users.

3.2.2. Item-based collaborative approach (I-col)

Item-based collaborative filtering predicts a user's interest in an item based on the similarity of the item to other items the user has rated. The similarity between two items i and j can be calculated using cosine similarity:

$$Sim\ (i,\ j) = \frac{\sum_{u \in U_{ij}} r_{ui}\, r_{uj}}{\sqrt{\sum_{u \in Uij} r_{ui}^2}\sqrt{\sum_{u \in Uij} r_{uj}^2}}$$

where r_u, is the rating of user u for item i and U_{ij} is the set of users who have rated both items i and j. The prediction for user u for item i is given by the following formula, where N(i) is the set of top k similar items.

$$\hat{r}_{u,j} = \frac{\sum_{j \in N_{(i)}} Sim(i,\ j) r_{u,\ j}}{\sum_{j \in N_{(i)}} \left|Sim(i,\ j)\right|}$$

3.3. SVD approach

Matrix factorization techniques, such as SVD, reduce dimensionality by decomposing the rating matrix R into three lower-dimensional matrices U, Σ, and V such that R≈UΣVTR, where U is an m×k user-feature matrix, Σ is a k×k diagonal matrix of singular values, and V is an n×k item-feature matrix.

The predicted rating $r_{u,i}$ for user u and item i is given by:

$$\hat{r}_{u,i} = \bigcup_u \sum v_i$$

where u is the u-th row of U and vi is the i-th row of V. The optimization problem for matrix factorization is to minimize the regularized squared error:

$$\min \sum \left(r_{u,i} - U_u^T V_i\right)^2{}_{u,v(u,i) \in K} + \lambda(\|U_u\|^2 + \|v_i\|^2)$$

The overall structure of our model is divided into three distinct stages, each playing a critical role in the recommendation process as shown in Figure 1.

1. Initial Stage: This includes data preprocessing and data splitting.
2. Semi-Supervised Stage: In this stage, we employ the co-training approach to enhance user-based and item-based collaborative filtering.
3. Ensemble Stage: In this stage, we used the stacking technique to combine the enhanced user-based and item-based collaborative approach with the SVD model for final prediction.

3.4. Initial stage

Data Loading and Preprocessing

The MovieLens dataset is first imported from a CSV file into a DataFrame. After that, StandardScaler is used to normalize the ratings in the dataset so that their mean is 0 and their standard deviation is 1. This helps to stabilize the machine learning model's learning process. The movie ID and user ID columns are then encoded as integer codes and transformed into category data types. The models require this encoding because they process numerical inputs.

Table 2. Issues and challenges of existing approaches

Issue/Challenge	Description	References
Quality of Pseudo-labels	The quality of pseudo-labels, inferred labels assigned to unlabeled data based on model predictions, is pivotal in semi-supervised learning for recommendation systems. Incorrect or noisy pseudo-labels can degrade model performance by introducing bias or inconsistencies. High-quality pseudo-label generation often requires robust methods for handling label noise and uncertainty. Ensuring the quality of pseudo-labels through techniques such as self-training or confidence-based filtering is crucial to improving the effectiveness of recommendation systems.	[23–25]
Data Distribution	Ensuring alignment in the distribution of labeled and unlabeled data is essential to prevent bias in models. Differences in data distribution can lead to models that do not generalize well, affecting the accuracy and effectiveness of recommendations across diverse user preferences and item characteristics.	[26,27]
Model Overfitting	Preventing overfitting is critical, especially when using ensemble techniques with limited labelled data. Ensemble models, which combine multiple base learners, can potentially memorize noise in the training data, leading to poor generalization on unseen data. Proper regularization and validation strategies are necessary to mitigate this risk.	[28–30]
Scalability	Managing computational resources effectively is crucial, particularly with large-scale datasets common in recommendation systems. Ensemble methods can be computationally intensive due to the need to train and integrate multiple models. Scalable implementation and optimization are necessary to ensure efficient processing and deployment in real-world applications.	[31,32]
Algorithm Complexity	Implementing and tuning ensemble algorithms requires expertise and computational resources. Ensemble methods involve integrating diverse algorithms or models, each with its own parameters and configurations. Optimizing these parameters and ensuring compatibility across different techniques requires advanced knowledge and careful experimentation.	[33,34]
Evaluation Metrics	Developing metrics that accurately assess recommendation quality beyond traditional metrics like MAE and RMSE is challenging. Recommendation systems aim to enhance user satisfaction and engagement, which may not be fully captured by standard metrics. Developing and adopting metrics that align with user preferences and business objectives is essential for comprehensive evaluation.	[35,36]
Robustness to Concept Drift	Adapting models to changes in user preferences or item popularity over time is crucial for maintaining recommendation accuracy. Concept drift occurs when the underlying relationships between users and items evolve, requiring continuous model adaptation. Ensuring robustness to concept drift involves monitoring data changes and updating models accordingly to provide relevant recommendations.	[37,38]
Interpretability	Ensuring transparency in decision-making processes within complex ensemble models is challenging. Ensemble methods often combine diverse models or algorithms, making it difficult to interpret how decisions are made. Enhancing interpretability helps build user trust and facilitates debugging and refinement of recommendation systems.	[39–41]
Data Privacy and Security	Addressing privacy concerns when using unlabeled data, especially in sensitive domains, is paramount. Unlabeled data may contain sensitive information about users or items, raising privacy risks if not handled properly. Implementing data anonymization techniques and adhering to privacy regulations are essential to protect user confidentiality and trust.	[42–44]

Table 3. Description of the core features of the MovieLens 100K dataset

Sl. No.	Feature	Data type	Description
1	userId	Numeric	User ID
2	movieId	Numeric	Movie ID
3	rating	Numeric	Rating given by the user
4	timestamp	Numeric	Timestamp of the rating

3.5. Data Splitting

The data is then divided into test, validation, and training sets to make sure that each user's data is represented in each split and to enable customized recommendations. Initially, the data is divided into 20% temporary sets and 80% training sets for each user. Another division of the temporary set is made into 50% test and 50% validation sets. By ensuring that every user's data is included throughout the whole model training and evaluation process, this method helps the models better understand and anticipate the preferences of specific users.

Table 4. Description of the core features of the FilmTrust dataset

Sl. No.	Feature	Data type	Description
1	userId	Numeric	User ID
2	itemId	Numeric	Movie ID
3	rating	Numeric	Rating given by user (0.5-4.0)

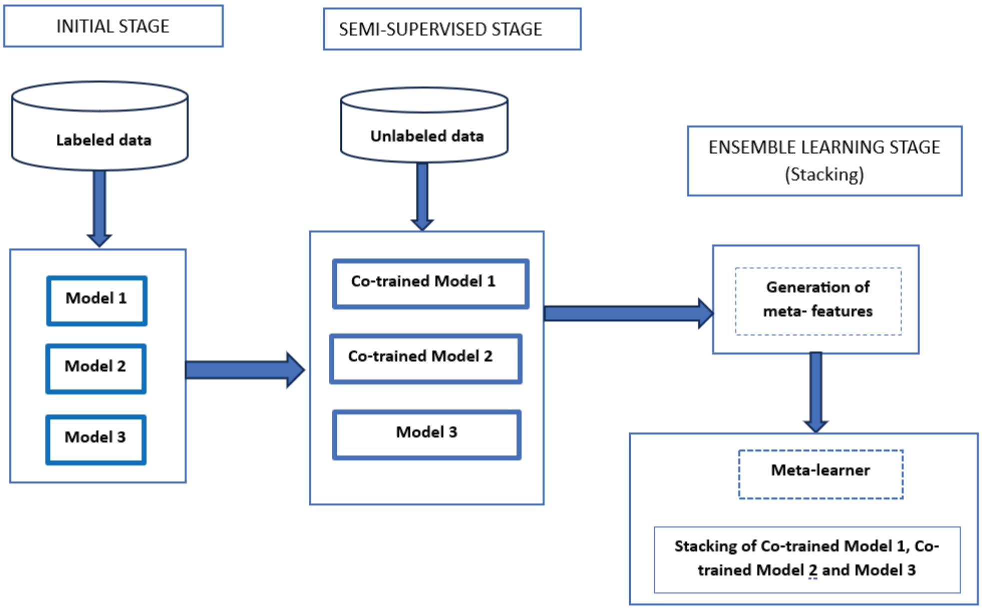

Figure 1. Flow chart of our proposed CTEL approach

3.6. Model training

Three distinct recommendation models—user-based collaborative filtering (CF), item-based CF, and singular value decomposition (SVD)—are initialized and trained throughout the training phase. The format of the training data is designed to work with the Surprise library, a recommendation system specialist library. Three models are initialized: SVD, a matrix factorization approach, and the k-nearest neighbours (KNN) algorithms for user-based and item-based CF.

3.7. Semi-supervised stage

3.7.1. Co-training approach

Through co-training, the item-based and user-based CF models are further improved. Unlabeled data points that were not part of the original training set are found. The user-based and item-based CF models are used iteratively to generate pseudo-labels for the unlabeled data during co-training. To ensure that new training data is highly accurate, pseudo-labels are permitted under a confidence criterion. Using the expanded training set, which now includes the freshly pseudo-labeled data, the models are retrained. The models are improved by this repeated process, which increases their capacity to generalize from the existing data.

3.8. Ensemble stage

3.8.1. Meta-feature preparation

Meta-features are generated from the predictions of all three models to prepare for the stacking stage. In the validation and test sets, predictions are produced for every user-item pair using all three models. In addition to the raw predictions, other features are computed, like the variance and the product of the forecasts. These other features provide richer information to the meta-learner, capturing additional interaction effects and the degree of agreement between the models. The introduction of variance and product-based meta-features in the CTEL framework is guided by theoretical reasoning rather than arbitrary design. The variance feature represents the degree of disagreement among the base recommenders and serves as an indicator of predictive uncertainty. Incorporating this feature enables the meta-learner to identify instances where model opinions diverge, allowing it to assign appropriate weights and improve reliability under sparse rating conditions. The product feature, on the other hand, captures the joint reinforcement between user-based and item-based predictions, emphasizing cases where both models produce consistent estimations. This interaction term helps the meta-learner recognize non-linear complementarities without increasing model complexity. Similar strategies are widely adopted in meta-learning and algorithm-selection research, where such statistical and interaction-based meta-features are used to estimate the competence and cooperation level of candidate algorithms. Therefore, the use of variance and product features in this study is theoretically grounded in ensemble and meta-learning principles, providing interpretable and effective signals that enhance the stacking process within the recommender system.

3.8.2. Cross-validation

K-Fold cross-validation is used to reliably create meta-features and targets for the meta-learner's training. The training data is divided into K folds, with one fold serving as the validation set and the other K folds serving as the training folds for each iteration. The SVD, user-based CF, and item-based CF models are trained in each fold, and meta-features for the validation set are generated from their predictions. The meta-learner is given a comprehensive training set consisting of meta-features from all folds.

Algorithm 1 The pseudocode of the proposed Co-training Ensemble Learning (CTEL)

Input: MovieLens dataset D = (userId, movieId, ratings)}
Base learners: User-based CF, item-based CF, SVD
Output: Final model; Evaluation metrics: RMSE, MSE, MAE
Initial Stage
Input: MovieLens dataset D
Output: Prepared training, validation, and test sets
Data Loading and Preprocessing
Load the dataset D = (u,m,r_{um})
Encode feature as categorical values
For each user u, split the data into: $D_{train}^{(u)}$ (80%) training set and $D_{temp}^{(u)}$(20%) as temporary set
Further split the temporary set into: $D_{val}^{(u)}$ (50%) Validation set and $D_{test}^{(u)}$ (50%) test set
Semi-supervised stage
Input: Training set D_{train} and validation set D_{val}
Output: Enhanced models through co-training
Model training
Initialize base learners: User-based CF (CF_{user}), item-based CF(CF_{item}), SVD
Train SVD on D_{train}
Identify unlabeled data $D_{unlabeled} = D \backslash D_{train}$
Fit CF_{user} and CF_{item} on D_{train}
For *iter=1* to N:
For each $(u,m) \in D_{unlabeled}$:
Predict $\hat{r}_{um}^{user}$ using CF_{user}
Predict $\hat{r}_{um}^{item}$ using CF_{item}
If $\hat{r}_{um}^{user} \geq \theta$(thresold), add (u,m, $\hat{r}_{um}{}^{user}$) to D_{train}
Re-train CF_{user} and CF_{item} on the updated D_{train}
Meta- Feature Preparation
Generate predictions for each $(u,m) \in D_{val} \cup D_{test}$ using all models:

$$\hat{r}_{um}^{user}, \hat{r}_{um}^{item}, \hat{r}_{um}^{SVD}$$

Create meta features: $X_{um} = [\hat{r}_{um}^{user}, \hat{r}_{um}^{item}, \hat{r}_{um}^{SVD}, var(\hat{r}_{um}), \prod(\hat{r}_{um})]$
Where var($\hat{r}_{um}$) is the variance and $\prod(\hat{r}_{um})$ is the product of the predictions
Ensembling stage
Input: Combined meta-features X, Targets y
Output: Final model and evaluation metrics
Cross-validation
Perform k-fold cross-validation on D_{train}
For each fold k
Split D_{train} into training subset $D_{train}^{(k)}$ and validation subset $D_{val}^{(k)}$
Train CF_{user},CF_{item} and SVD on $D_{train}^{(k)}$
Generate meta-features $X_{um}^{(k)}$ for $D_{val}^{(k)}$
Combine meta-features $X=\cup_k X_{um}^{(k)}$ and targets $y= \cup_k r_{um}^{(k)}$
Meta-learning Training
Train linear regression meta -learner on combined meta-features X and targets y
Evaluation
Generate meta- features X_{tset} for D_{test}
Make final prediction on D_{test} using the meta-learner
$\hat{r}_{um}^{final}$ = meta-learner(X_{test})
Evaluate using RMSE, MSE, MAE to check the accuracy of final model

$$RMSE = \sqrt{\frac{1}{|D_{\text{test}}|}} \sum_{(u,\, m) \in D_{\text{test}}} (\hat{r}_{\text{um}}^{\text{final}} - r_{\text{um}})^2$$

$$MSE = \frac{1}{|D_{\text{test}}|} \sum_{(u,\, m) \in D_{\text{test}}} (\hat{r}_{\text{um}}^{\text{final}} - r_{\text{um}})^2$$

$$MAE = \frac{1}{|D_{\text{test}}|} \sum_{(u,\, m) \in D_{\text{test}}} |\hat{r}_{\text{um}}^{\text{final}} - r_{\text{um}}|^2$$

Table 5. Dataset characteristics, highlighting the dataset's degree of sparsity for rating prediction tasks

Dataset	User	Item	Ratings	Density
MovieLens 100K	943	1682	100, 000	6.3%
FilmTrust	1508	2071	35,497	1.14%

Table 6. Comparison of our proposed CTEL model with collaborative approaches and blending ensemble technique

Dataset	Metric	U-Col	I-Col	SVD++	Blending	CTEL
MovieLens	RMSE	0.99	0.95	0.90	0.89	**0.83**
	MSE	0.98	0.91	0.81	0.79	**0.70**
	MAE	0.76	0.73	0.69	0.69	**0.64**
FilmTrust	RMSE	0.89	0.81	0.79	0.78	**0.70**
	MSE	0.83	0.66	0.63	0.62	**0.49**
	MAE	0.70	0.62	0.61	0.61	**0.54**

3.8.3. Meta-learner training

The aggregated meta-features and targets from the cross-validation procedure are then used to train the meta-learner, a Linear Regression model. By utilizing the advantages of each individual model, the meta-learner learns how to integrate the predictions of the base models in the best possible way to reduce prediction error.

3.9. Evaluation

Lastly, the test set is used to assess the recommendation system's performance. The predictions from the training user-based CF, item-based CF, and SVD models are used to create meta-features for the test set. The accuracy of the recommendations is measured using Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE) when the meta-learner makes its final predictions on the test set.

The pseudo-labeling process in CTEL depends on a confidence threshold that controls which predicted ratings are accepted as pseudo-labels. These parameters influence how additional training data are generated during each co-training round. When pseudo-labels are selected with sufficiently high confidence, the majority of the new samples contribute correct information to the learning process. This increases the effective density of the user–item matrix, improving model generalisation under sparse conditions. Conversely, if the confidence threshold is too low, a larger fraction of inaccurate pseudo-labels is introduced, leading to accumulated error and model drift. Hence, there exists a critical precision level above which each co-training iteration continues to reduce overall error.

4. Results and Discussion

Explicit ratings from the publicly available MovieLens 100K (ML-100K) dataset and the FilmTrust dataset, gathered from the FilmTrust social network, are used to evaluate the proposed technique experimentally. The MovieLens dataset has a rating density of 6.3%, and the FilmTrust dataset has a rating density of 1.14% (Table 5), highlighting the degree of sparsity in the datasets for rating prediction tasks [48].

SVD, Blending, item-based collaborative filtering, and user-based collaborative filtering were tested

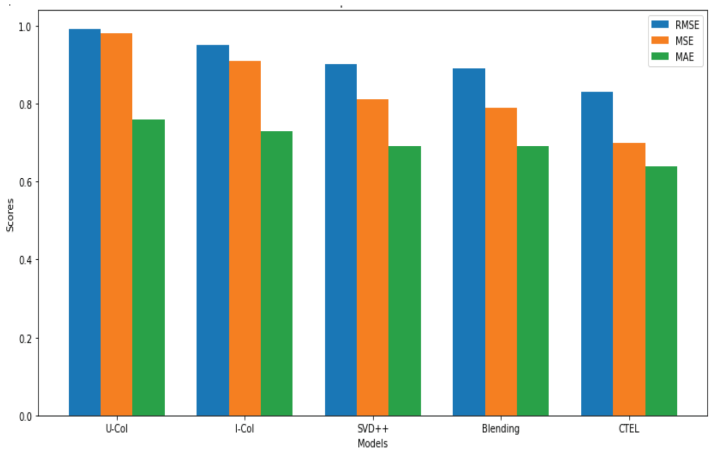

Figure 2. Comparison of metrics across different models (MovieLens dataset)

against the CTEL model. The CTEL model outperformed the other methods, achieving notably lower error rates on both datasets.

As shown in Table 6, our proposed model has consistently outperformed alternative approaches. Based on the evaluation criteria of RMSE, MSE, and MAE, our model achieves better results than other existing techniques on both datasets. While the Semi-Supervised Collaborative Filtering Ensemble (SSEF) existing model introduced a static co-training and blending framework, the proposed CTEL method focuses on enhancing sparsity handling through refined co-training and advanced meta-feature integration. CTEL performs dedicated co-training between user-based and item-based CF models rather than between CF and MF views, allowing view-specific refinement [40]. Furthermore, its stacking phase incorporates statistical meta-features, such as inter-model variance and prediction interactions, enabling the meta-learner to capture agreement patterns among base models better. These design differences make CTEL more resilient to data sparsity compared to static semi-supervised ensembles. The data show that the CTEL model performs better than these conventional techniques across all metrics.

In particular, collaborative filtering approaches based on users and items demonstrated higher RMSE, MSE, and MAE values than the CTEL model, despite their effectiveness. Even with its resilience, the SVD technique was not as accurate as the CTEL model. The CTEL model even surpassed the blending strategy, which combines predictions from different models. Because the CTEL model incorporates both ensemble learning and co-training, it leverages different collaborative filtering techniques, producing more accurate suggestions. The CTEL model's better performance on both datasets demonstrates its efficacy and marks a noteworthy development in the field of recommendation systems.

The graphs for the performance metrics (RMSE, MSE, and MAE) of our proposed CTEL model, compared with SVD, blending, user-based collaborative filtering, and item-based collaborative filtering, for the MovieLens and FilmTrust datasets are shown in Figure 2 and Figure 3, respectively. The comparison results for the MovieLens shown in Figure 2 show that

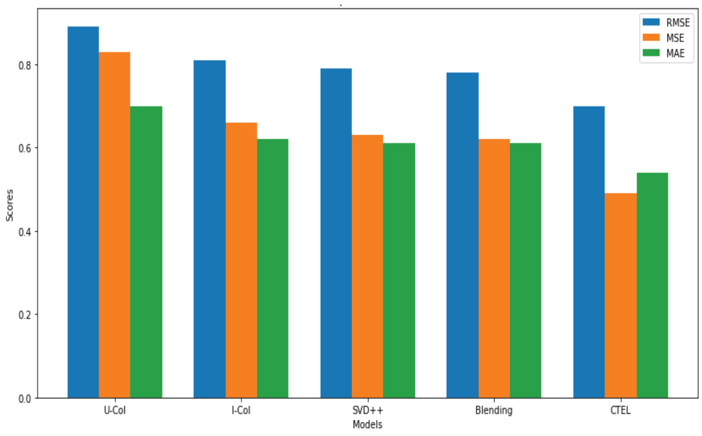

Figure 3. Comparison of metrics across different models (FilmTrust dataset)

Table 7. Performance of CTEL and baseline models at varying data sparsity levels

Data Density	User-CF	Item-CF	Static Blend	CTEL (Proposed)
100	0.89	0.88	0.87	0.83
60	0.92	0.90	0.89	0.84
40	0.96	0.93	0.91	0.86
20	1.03	0.98	0.96	0.89

the CTEL model consistently produces lower error rates. On the FilmTrust dataset, the CTEL model outperforms conventional approaches, as illustrated in Figure 3.

To further validate the robustness of the proposed CTEL framework under sparse data conditions, a sparsity sensitivity analysis was conducted. The MovieLens 100K dataset was randomly subsampled to simulate data densities of 100%, 60%, 40%, and 20%. The CTEL model was compared against a conventional static blending ensemble similar to SSEF [40] and individual base learners.

As shown in Table 7, CTEL maintains significantly lower RMSE across all sparsity levels.When the data density is reduced from 100% to 20%, the RMSE of static blending increases by 0.09, whereas the RMSE of CTEL increases only by 0.06. This indicates that CTEL effectively handles data sparsity by utilizing high-confidence pseudo-labels and complementary model views through co-training and stacking.

5. Conclusion

This study introduces the CTEL (Collaborative filtering with Tuning Ensemble Learning) model, demonstrating the effectiveness of combining semi-supervised learning and ensemble techniques to enhance collaborative filtering in recommendation systems. Our approach addresses key challenges such as data sparsity, scalability, and generalizability—issues that often hinder the performance of traditional recommendation models. Through extensive testing on the MovieLens and FilmTrust datasets, we observe significant improvements in standard evaluation metrics such as RMSE, MSE, and MAE, suggesting that the CTEL model is not only effective in handling sparse data but also maintains robustness and accuracy across diverse datasets. When compared to other state-of-the-art models in the literature, such as those employing deep learning for collaborative filtering (e.g., Neural Collaborative Filtering (NCF)) or those using matrix factorization-based methods (e.g., ALS or SVD++), our CTEL model stands out for its novel integration of semi-supervised learning to generate more reliable pseudo-labels for unlabeled data and the application of ensemble learning to combine multiple weak learners into a more robust and scalable system. While previous studies have explored the use of semi-supervised techniques to improve recommendation quality, few have simultaneously addressed the challenges of ensemble methods in this domain, with a focus on scalability and real-time adaptability to new data. Although future work may explore neural meta-learners to evaluate deeper feature interactions, the current linear approach aligns with the goals of interpretability, reproducibility, and stability in sparse recommendation scenarios.

The model's ability to handle sparse user-item interactions and its scalability are particularly noteworthy. Compared with traditional methods, which may struggle with data sparsity and overfitting, our CTEL model demonstrates superior performance in predicting user preferences even in the presence of missing data, making it more suitable for large-scale real-world applications where data is often incomplete or noisy. Moreover, by incorporating ensemble learning, our approach is more flexible and less prone to overfitting, a common challenge with complex models that rely on limited labeled data. While the results are promising, several avenues remain for future work. To further enhance the performance of the CTEL model, future research could explore more sophisticated feature engineering techniques, such as integrating contextual features (e.g., temporal, geographic, or demographic data) or multimodal data (e.g., combining text and image data from movie descriptions or user reviews). Additionally, the model's effectiveness could be evaluated on a broader set of datasets across different domains, including e-commerce, healthcare, and social media platforms, to assess its versatility and adaptability. Lastly, further exploration of active learning strategies could help minimize reliance on labeled data, making the model even more efficient in environments with limited labeled data. Thus, our CTEL model represents a promising step forward in the development of scalable, accurate, and robust recommendation systems. By combining semi-supervised learning and ensemble methods, it offers a novel solution to the persistent challenges of data sparsity, overfitting, and scalability in collaborative filtering. Its comparative advantage lies in its ability to effectively combine weak learners while utilizing unlabeled data, offering valuable insights for both academia and industry in the quest for more precise and trustworthy recommendation systems.

AUTHORS

Nisha Sharma – Assam Down Town University, Panikhaiti, Guwahati, Assam, India, e-mail: nisha.gu24@gmail.com.

Mala Dutta[*] – Assam Down Town University, Panikhaiti, Guwahati, Assam, India, e-mail: mala.dutta@adtu.in.

[*]Corresponding author

References

[1] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa, "Leveraging Bayesian deep learning and ensemble models for improved uncertainty quantification in recommendations," *Heliyon*, vol. 10, no. 2, 2024, e24188.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, 2005, pp. 734–749.

[3] C. C. Aggarwal, *Recommender Systems: The Textbook* (2nd ed.), Springer, 2023.

[4] J. Bobadilla, F. Ortega, and A. Hernando, "Collaborative filtering techniques: New perspectives and future directions," *Information Fusion*, vol. 92, 2023, pp. 245–261.

[5] R. Cheraghi, A. M. Mahfoozi, S. Zolfaghari, M. Shabani, M. Ramezani, and H. R. Rabiee, "Epistemic uncertainty-aware recommendation systems via Bayesian deep ensemble learning," *arXiv* preprint, 2025, arXiv:2504.10753.

[6] V. Coscrato and D. Bridge, (202. "Recommendation uncertainty in implicit feedback recommender systems," *Communications in Computer and Information Science (AICS 2022)*, Springer, 2023.

[7] X. He, L. Liao, H. Zhang, and T. S. Chua, "Neural collaborative filtering revisited: Deep interaction modeling in recommendation systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, 2024, pp. 410–423.

[8] N. Knyazev and H. Oosterhuis, "A lightweight method for modeling confidence in recommendations with learned Beta distributions," *arXiv* preprint, 2023, arXiv:2308.03186.

[9] Y. Koren, R. Bell, and C. Volinsky, C, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, 2009, pp. 30–37.

[10] D. D. Lee, and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2001, pp. 556–562.

[11] D. Liang, R. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," *arXiv* preprint, 2018, arXiv:1802.05814.

[12] T. Liu, C. Gao, Z. Wang, D. Li, J. Hao, D. Jin, and Y. Li, "Uncertainty-aware consistency learning for cold-start item recommendation," *arXiv* preprint, 2023, arXiv:2308.03470.

[13] X. Ning, X. Wang, S. Xu, W. Cai, L. Zhang, L. Yu, and W. Li, "A review of research on co-training," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 18, 2021, e6276.

[14] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," *arXiv* preprint, 2020, arXiv:2005.09683.

[15] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook* (3rd ed.), Springer, 2022.

[16] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 880–887.

[17] H. Wang, Z. Sun, Y. Du, L. Zhang, T. He, and Y.-S. Ong, "Uncertain multi-objective recommendation via orthogonal meta-learning enhanced Bayesian optimization," *arXiv* preprint, 2025, arXiv:2502.13180.

[18] S. Wang and Y. Zhang, "Hybrid collaborative filtering via content–context fusion: A comprehensive review," *Expert Systems with Applications*, vol. 228, 2023, 120979.

[19] W. Xu, J. Zheng, J. Lin, M. Han, and J. Du, "Unified representation learning for multi-intent diversity and behavioral uncertainty in recommender systems," *arXiv* preprint, 2025, arXiv:2509.04694.

[20] Z. Xu, Z. Yang, Z. Guo, S. Liu, L. Lin, X. Liu, Y. Liu, and H. Li, "UMRE: A unified monotonic transformation for ranking ensemble in recommender systems," *arXiv* preprint, 2025, arXiv:2508.07613.

[21] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, 2017, pp. 1–38.

[22] S. Zhang, K. Liu, Z. Yu, B. Feng, and Z. Ou, "Hybrid recommendation system combining collaborative filtering and content-based recommendation with keyword extraction," *Applied and Computational Engineering*, vol. 2, 2023, pp. 927–939.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, 2014, pp. 1929–1958.

[24] J. Dean, G. Corrado, R. Monga, et al., "Large-scale distributed deep networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[25] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Neural Networks Deep Learning*, 2015, arXiv:1503.02531.

[26] G. Brown, "Ensemble learning," *Encyclopedia of Machine Learning*, Springer, 2011.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[28] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.

[29] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 253–260.

[30] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, 1996, pp. 69–101.

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[32] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] L. H. Gilpin, D. Bau, B. Z. Yuan, et al., "Explaining explanations: An overview of interpretability of machine learning," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[34] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, 2006, pp. 265–284.

[35] S. Goldwasser, S. Micali, and R. Rivest, "Secure multi-party computation without agreement," *Journal of Cryptology*, vol. 18, 2005, pp. 247–287.

[36] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, Springer, 2017.

[37] Roy D, Dutta M. A systematic review and research perspective on recommender systems. Journal of Big Data. 2022;9:59. doi: 10.1186/s40537-022-00592-5.

[38] P. Datta, "MovieLens 100k dataset," *Kaggle*, 2016. Available: https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset

[39] G. Guo, "Film Trust dataset," *LibRec*, 2011. Available: https://guoguibing.github.io/librec/datasets.html

[40] J. Wu, X. Sang, and W. Cui, "Semisupervised collaborative filtering ensemble," *World Wide Web*, 37 vol. 24, 2021, pp. 19411960.

[41] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. 2008:263-272. doi:10.1109/ICDM.2008.22.

[42] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, 2008, pp. 1257–1264.

[43] S. Rendle, "Factorization machines with libFM," *ACM Transactions Intelligent Systems and Technology*, vol. 3, 2012, pp. 1–22.

[44] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1235–1244.

[45] Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Computing Surveys. 2014;46(4):Article 44. doi:10.1145/2523813.

[46] Ben-Nun T, Hoefler T. Demystifying parallel and distributed deep learning: an in-depth concurrency analysis. ACM Computing Surveys. 2019;52(4):65:1-65:43. doi:10.1145/3320060.

[47] Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence. IEEE Access. 2018;6:52138-52160. doi:10.1109/ACCESS.2018.2870052.

[48] Zhang Y, Chen X. Explainable recommendation: a survey and new perspectives. Foundations and Trends in Information Retrieval. 2020;14(1): 1-101. doi:10.1561/1500000066.