UDC 004.891.3(045)
DOI:10.18372/1990-5548.84.20192

[1]I. O. Pyshnograiev,
[2]A. E. Shyraliiev

A COMPREHENSIVE BENCHMARK OF COLLABORATIVE FILTERING METHODS ON IMPLICIT FEEDBACK DATASETS

[1,2]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, Ukraine
E-mails: [1]pyshnograiev@gmail.com ORCID 0000-0002-3346-8318,
[2]anarshyraliiev@gmail.com ORCID 0009-0007-4140-5476

***Abstract*—*Collaborative filtering is a foundational technique in modern recommender systems, especially when dealing with implicit feedback signals such as clicks, purchases, or listening behavior. Despite the abundance of collaborative filtering models, including classical, probabilistic, and neural approaches, there is a lack of standardized, large-scale evaluations across diverse datasets. This study presents a comprehensive empirical benchmark of 13 collaborative filtering algorithms encompassing matrix factorization, pairwise ranking, variational and non-variational autoencoders, graph-based neural models, and probabilistic methods. Using four representative implicit feedback datasets from different domains, we evaluate models under a unified experimental protocol using ranking-based metrics (MAP@10, NDCG@10, Precision@10, Recall@10, MRR), while also reporting training efficiency. Our results reveal that neural architectures such as NeuMF, VAECF, and LightGCN offer strong performance in dense and moderately sparse scenarios, but may face scalability constraints on larger datasets. Simpler models like EASE^R and BPR often achieve a favorable balance between performance and efficiency. This benchmark offers actionable insights into the trade-offs of modern collaborative filtering methods and guides future research in implicit recommender systems.*

Keywords—Collaborative filtering; implicit feedback; recommender systems; benchmarking; ranking metrics.

I. INTRODUCTION

Recommender systems have become a cornerstone of digital platforms, driving user engagement by personalizing content across domains such as e-commerce, streaming services, and social media. A foundational technique underlying many of these systems is collaborative filtering (CF), which infers user preferences based on patterns of interaction across a population. Since the early work [1], which introduced the GroupLens system, a user-based CF approach relying on similarities between users for rating-based recommendation, and the subsequent development of scalable item-based methods [2], CF has evolved into a diverse landscape of models and algorithms.

Traditional CF methods often relied on explicit feedback – numerical ratings that directly reflect user preferences. However, in modern applications, such feedback is rare. Instead, systems increasingly depend on implicit feedback, such as clicks, purchases, or watch durations – signals that are abundant but indirect, noisy, and lack clear negative examples. This shift introduces unique modeling challenges, which were first formally addressed [3] through confidence-aware matrix factorization tailored to implicit data.

Despite the proliferation of algorithms designed to operate on implicit feedback, including classical matrix factorization methods, neural network-based models, graph convolutional architectures, as well as both variational and shallow autoencoder-based approaches, a comprehensive and fair benchmarking of these methods under a unified evaluation framework remains scarce. Existing studies often compare only a limited subset of models, focus on a narrow set of datasets, or employ inconsistent evaluation protocols, making it difficult to draw robust and generalizable conclusions about algorithmic performance. Moreover, training efficiency, an important factor in the practical deployment of recommender systems, is frequently overlooked, with few benchmarks providing a systematic comparison of training time across methods.

To address this gap, a comprehensive empirical benchmark of 13 CF algorithms on stratified subsets derived from 4 widely-used benchmark datasets for implicit feedback is presented. The benchmark covers a broad spectrum of models, including classical matrix factorization methods (MF, PMF), pairwise optimization techniques (BPR, IBPR, WBPR), probabilistic models (HPF), neural CF (NeuCF), autoencoder-based approaches (VAECF, EASE^R, BiVAECF, SANSA), and graph-based neural recommenders (NGCF, LightGCN). We include both recent advances and historically

influential baselines to assess performance across diverse model classes and evaluate them under a consistent framework.

Our contributions are as follows:

- we provide a standardized evaluation of 13 CF algorithms on implicit feedback tasks, ensuring consistent preprocessing, training, and evaluation across models;
- we highlight the relative strengths and weaknesses of each method in terms of accuracy, scalability, and robustness across datasets with varying sparsity and scale.

Through this work, we aim to inform practitioners and researchers about the current state of CF under implicit feedback, shedding light on which algorithms offer the best trade-offs in practice and helping guide future developments in recommender system design.

II. Related Work

Despite the proliferation of CF algorithms, comprehensive and standardized benchmarking has been identified as an area needing more attention. Existing studies often compare a limited subset of models or use inconsistent evaluation protocols.

However, several efforts have been made to compare and benchmark CF techniques

- *Domain-Specific Benchmarks:* several studies provide benchmarks within specific domains. For example, one study provided a detailed evaluation of pure and hybrid CF methods in drug repurposing by comparing 11 models across eight datasets, with a focus on reproducible methodology [4]. Another study compared several collaborative filtering methods (such as ALS, LightFM, Prod2Vec, RP3Beta, and SLIM) for job recommendations on classifieds, using both offline metrics and online A/B testing [5].
- *Comparative Analyses:* researchers have conducted comparative analyses of different CF algorithms. One study compared memory-based and model-based collaborative filtering techniques, discussing how they handle challenges such as quality and scalability, and shared comparison results for several methods [6]. Another study analyzed collaborative filtering algorithms like SVD, SVDpp, and SlopeOne for e-commerce, using the Surprise library to evaluate their accuracy in rating prediction and their efficiency [7].
- *Benchmarking Frameworks and Platforms:* efforts towards open benchmarking are emerging. One example is the BARS framework, which was proposed to support rigorous evaluation of recommender systems [8]. Platforms like Papers With Code also list datasets (e.g., MovieLens, Yelp, Amazon-Book) and the best-performing models, offering a form of continuous benchmarking for various recommendation tasks, including CF. Tools like RecBole are also mentioned as providing preprocessed datasets and evaluation frameworks.
- *Focus on Evaluation Rigor:* one study shows that newer and more complex models do not always perform better than simpler, well-tuned ones, and it emphasizes the importance of better scientific practices in benchmarking [9].

The current work aims to contribute to this area by providing a comprehensive and fair benchmark of a wide range of CF algorithms specifically for implicit feedback datasets, using a unified evaluation protocol across multiple standard datasets. This aligns with the identified need for robust and generalizable conclusions regarding algorithmic performance and efficiency in the domain of implicit feedback.

III. Methodology

To ensure a rigorous and reproducible comparison of CF methods on implicit feedback data, we adopt a unified evaluation framework spanning multiple datasets, model families, and metrics. This section details the experimental design of our benchmark, including dataset selection and preprocessing, the set of algorithms evaluated, evaluation metrics tailored to implicit recommendation tasks, and the training and evaluation protocol used to ensure fairness and consistency across models.

A. Datasets

We evaluate CF methods on four widely-used benchmark datasets that represent a diverse range of domains and interaction characteristics. For two large-scale datasets, 10% stratified subsets are used to ensure manageable training time and consistent comparison across models with varying computational demands. In this context, stratification means preserving the relative interaction density of each user — that is, maintaining the ratio of the number of items a user interacted with in the subset compared to the full dataset. This ensures that the sampled data reflects the original user behavior distribution and prevents bias introduced by uniform or random subsampling. All datasets are treated as implicit feedback, where observed interactions indicate user preference, although the form of the signal varies across datasets. An overview of the dataset statistics is provided in Table I.

TABLE I. EXPERIMENTAL DATA STATISTICS

Dataset	Users #	Items #	Interactions #	Density
MovieLens 100k	943	1,682	100,000	6.30%
MovieLens 20M	138,493	26,744	20,000,263	0.54%
Steam (10%)	121,521	14,807	388,535	0.02%
Last.fm 360k (10%)	35,887	93,612	1,742,433	0.05%

In the MovieLens 100k and MovieLens 20M datasets [28], originally collected as explicit rating datasets, the ratings are treated as implicit signals, rather than being binarised, interpreting each rating as a proxy for the number of times a user has interacted with an item. This approach preserves the relative strength of user preferences while maintaining consistency with the implicit feedback setting. MovieLens 100k contains 100,000 ratings from 943 users on 1,682 movies, and serves as a relatively dense benchmark (sparsity 6.30%). MovieLens 20M includes over 20 million ratings from more than 138,000 users and 26,000 items, offering a large-scale scenario for model evaluation.

Steam is a game interaction dataset consisting of user play hours [29]. A stratified 10% subsample of the full dataset is used, yielding 388,535 interactions between 121,521 users and 14,807 games. As users typically play only a small subset of available titles, this dataset is highly sparse (0.02%), making it well-suited for testing model robustness in cold-start and sparse settings.

Last.fm 360k contains music listening histories from Last.fm users [30]. A 10% stratified subset is used, comprising over 1.7 million listening events from 35,887 users and 93,612 artists. With a sparsity of 0.05%, this dataset presents another high-sparsity challenge and reflects long-tail consumption patterns common in music recommendation.

B. Algorithms

We evaluate 13 CF algorithms, grouped into five major families based on their modeling approach: matrix factorization-based, pairwise ranking, autoencoder-based, graph-based neural models, and probabilistic models. This classification helps capture the spectrum of algorithmic strategies used in modern recommender systems and to understand their comparative strengths in implicit feedback settings.

Matrix Factorization-Based Models. Matrix factorization (MF) approaches are among the most established techniques for CF. These models learn low-dimensional latent vectors for users and items and estimate preferences through their inner product.

- *Probabilistic Matrix Factorization (PMF)*: xtends MF into a probabilistic framework by placing Gaussian priors over latent factors and treating user-item interactions as normally distributed variables. PMF is trained using maximum a posteriori estimation and provides a Bayesian interpretation of uncertainty in recommendations [10].
- *Matrix Factorization (MF):* a classical baseline in CF, MF maps users and items to latent factors and predicts interactions via dot product. It assumes that observed interactions reveal underlying user-item affinities. This work adopts an implicit-feedback version based on the general principles [11].
- *Neural Matrix Factorization (NeuMF):* combines the linear modeling capability of MF with the expressiveness of deep neural networks. NeuMF consists of two branches: a Generalized MF (GMF) and a Multi-Layer Perceptron (MLP), whose outputs are fused to model both linear and non-linear user-item interactions. It is optimized using a log loss suitable for implicit data and trained via negative sampling [12].

Pairwise Ranking Models. Pairwise ranking models optimize for the relative ordering of items rather than predicting absolute scores, which aligns well with the nature of implicit feedback.

- *Bayesian Personalized Ranking (BPR):* originally presented in [13], BPR is a seminal framework that learns to rank items by training on triplets (user, positive item, negative item). It optimizes a pairwise logistic loss, assuming users prefer observed items over unobserved ones.
- *Weighted Bayesian Personalized Ranking (WBPR):* a method called WBPR was proposed to address biases caused by non-uniform negative sampling, which often occur in practice, for example, due to popularity bias in candidate generation [14]. WBPR extends the BPR objective by weighting training triplets based on item popularity, improving ranking performance in real-world datasets.
- *Indexable BPR (IBPR):* a variant of BPR designed for scalability and top-k retrieval efficiency. It replaces the dot-product kernel with an angular similarity function, learning normalized embeddings compatible with indexing structures such as LSH. This makes IBPR well-suited for real-time recommendation tasks [15].

Autoencoder-based Models. Autoencoders model CF as a reconstruction task, where the system

learns to reconstruct a user's interaction history from a compressed latent representation [16]. These methods have become prominent in learning non-linear embeddings from sparse data.

- *Variational Autoencoder for Collaborative Filtering (VAECF):* based on the Mult-VAE framework [17], VAECF uses a variational autoencoder with a multinomial likelihood to model implicit interactions. It introduces KL divergence annealing to improve ranking metrics and performs well in sparse and cold-start settings [18].
- *Bilateral VAE for Collaborative Filtering (BiVAECF):* enhances traditional VAEs by learning symmetric latent representations for both users and items. This bilateral design captures the generative nature of dyadic interactions more effectively [19].
- *Embarrassingly Shallow Autoencoder ($EASE^R$)*: a linear autoencoder without hidden layers, solving a closed-form least squares reconstruction over the item-item co-occurrence matrix. $EASE^R$ is remarkably efficient and competitive, especially in sparse settings [20].
- *Scalable Approximate Non-Symmetric Autoencoder (SANSA):* a recent linear autoencoder model introducing an asymmetric encoder-decoder design and scalable approximations for efficient training on large datasets [21].

Graph-Based Neural Models. Graph-based recommenders treat the user-item interaction matrix as a bipartite graph and use graph convolutional networks (GCNs) to propagate information across the interaction structure.

- *Neural Graph Collaborative Filtering (NGCF):* a deep GCN model that propagates embeddings through the user-item graph using non-linear transformations and neighbor-aware aggregation. It captures high-order collaborative signals through recursive message passing [22].
- *LightGCN*: a simplified yet highly effective variant of NGCF that removes non-linear transformations, retaining only neighborhood aggregation. LightGCN combines multi-hop embeddings to form the final user/item representations and achieves strong performance with lower complexity [23].

Probabilistic Models. Probabilistic models provide a generative view of interaction data, leveraging count-based distributions and hierarchical priors.

Hierarchical Poisson Factorization (HPF): A Bayesian latent variable model designed for implicit data. HPF models user activity and item popularity using hierarchical Gamma priors and assumes observed interactions follow a Poisson distribution. It is particularly effective in highly sparse environments and avoids explicit negative sampling [24].

Exclusion of Sequential Models. While sequential recommender systems such as SASRec [25], and BERT4Rec [26] have achieved impressive results by modeling temporal dynamics, they are excluded from this benchmark since they cannot be compared to non-sequential CF methods that operate on static user-item interaction matrices. This decision reflects realistic scenarios where interaction timestamps are unavailable or unreliable. Moreover, sequential models typically require specialized temporal evaluation protocols and additional hyperparameter tuning, making them incompatible with our unified and model-agnostic benchmarking framework.

C. Evaluation Metrics

To assess the performance of CF models in the implicit feedback setting, we adopt several widely-used ranking-based metrics that reflect the quality of top-N recommendations. Since implicit feedback lacks explicit negatives, regression-based metrics (e.g., *RMSE*, *MAE*) are not applicable. Instead, the evaluation emphasizes how well the model ranks relevant items above irrelevant ones.

Normalized Discounted Cumulative Gain (NDCG@K). DCG@K evaluates the usefulness of a ranked list of items by assigning higher weights to items appearing near the top. It is defined as:

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \quad (1)$$

where rel_i is the binary relevance of the item at position i (1 if relevant, 0 otherwise).

To normalize this score and make it comparable across users with different numbers of relevant items, *NDCG@K* is computed as:

$$NDCG@K = \frac{DCG@K}{IDCG@K}, \quad (2)$$

where *IDCG@K* is the ideal *DCG@K*, i.e., the *DCG@K* obtained when all relevant items are ranked at the top.

Precision@K. Precision@K computes the proportion of recommended items in the top-K list that are relevant:

$$Precision@K = \frac{1}{K} \sum_{i=1}^{K} rel_i. \quad (3)$$

This metric evaluates how many of the top-K recommendations are correct, but does not account for their order within the list.

Recall@K. *Recall@K* measures the fraction of relevant items that are successfully recommended within the top-K list:

$$Recall@K = \frac{\sum_{i=1}^{K} rel_i}{|R_u|}, \quad (4)$$

where $|R_u|$ is the number of relevant items for user u. *Recall@K* emphasizes coverage – how well the model recovers a user's known preferences.

Mean Average Precision (MAP@K). *MAP@K* averages precision values computed at the ranks where relevant items occur. For a single user:

$$AP@K = \frac{1}{|R_u|} \sum_{i=1}^{K} Precision@i \cdot rel_i. \quad (5)$$

The final *MAP@K* score is the mean of *AP@K* across all users:

$$MAP@K = \frac{1}{|U|} \sum_{u \in U} AP@K_u. \quad (6)$$

MAP@K rewards models that not only retrieve relevant items but also rank them highly.

Mean Reciprocal Rank (MRR). *MRR* measures the inverse of the rank at which the first relevant item appears for each user:

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u}, \quad (7)$$

where $rank_u$ is the position of the first relevant item for user u. *MRR* is especially useful when only the first hit matters (e.g., in search or spotlight recommendation).

Choice of Metrics for Implicit Feedback. In implicit feedback scenarios, users provide only positive signals (e.g., clicks, plays, purchases), and the absence of interaction does not imply dislike. Consequently, evaluation must focus on ranking rather than prediction accuracy. Metrics like *NDCG@K*, *MAP@K*, and *MRR* capture both relevance and ranking order, which are critical in practice. *Precision@K* and *Recall@K* provide interpretable measures of recommendation quality from the perspectives of correctness and coverage, respectively.

By evaluating all metrics at $K = 10$, we emphasize the system's ability to deliver relevant results in the top-10 recommendations, a realistic cutoff for many real-world applications.

D. Experimental Protocol

To ensure fair and reproducible comparison across models, we adopt a unified training and evaluation procedure based on holdout testing and cross-validated hyperparameter tuning. Our protocol is designed to balance evaluation rigor with computational efficiency, given the wide range of models and datasets under consideration.

Train/Test Split. For each dataset, interaction data is divided into a training set (80%) and a test set (20%) at the user level. That is, for every user, 20% of their interactions are randomly withheld for testing, while the remaining 80% are used for training. This strategy reflects a standard evaluation setting for implicit feedback recommenders and simulates the common scenario of recommending additional items to users based on their known historical preferences.

All test sets remain fixed across all models, ensuring consistent and comparable evaluation.

Cross-Validation and Hyperparameter Tuning. Each model is trained and validated using a 5-fold cross-validation ($CV = 5$) strategy applied within the training set. Specifically, the training data is further divided into 5 folds; hyperparameters are tuned by training on 4 folds and validating on the remaining one, with the validation fold rotated in each run. This approach improves the robustness of model selection, especially for smaller or sparser datasets.

To manage the computational cost across the full benchmark suite, the hyperparameter search space is limited to a maximum of 10 configurations per model. These configurations are selected based on prior literature, implementation defaults, and empirical feasibility, and are evaluated using the same cross-validation setup.

Reporting. For each model and dataset, the best-performing hyperparameter setting is identified based on average cross-validation results within the training set. Then, using this selected configuration, the model is retrained on the full training set and evaluated on the fixed 20% test set. We report the final performance scores (e.g., MAP@10, NDCG@10, Precision@10, Recall@10, MRR) obtained on this test set. Additionally, training time for each model under its selected configuration is recorded and reported, providing insights into computational efficiency alongside accuracy. This ensures a consistent and fair comparison across models while reflecting realistic recommendation performance on unseen data.

Hardware. All experiments were conducted on a machine equipped with an Apple M2 Pro chip and 16 GB of unified memory (RAM). Depending on the capabilities and implementation of each model, training was performed using either the CPU or the integrated GPU via Apple's Metal backend.

Classical models and those without GPU acceleration support were run on the CPU, while deep learning models leveraging PyTorch or TensorFlow used the GPU when possible. Training time was recorded under these conditions to reflect realistic usage scenarios on moderately powered hardware, comparable to entry-level cloud compute environments (e.g., M-series Apple chips or budget AWS instances).

Software. All experiments were implemented in Python, utilizing the Cornac library for training and evaluating recommendation models. Cornac offers a consistent API and efficient implementations of a wide range of classical and deep learning-based recommenders, making it suitable for comparative benchmarking. We relied on Cornac's built-in support for model training, evaluation metrics, and cross-validation routines to ensure standardized and reproducible experiments across models and datasets [27].

IV. Experiments

In this section, the performance of 13 CF algorithms across four benchmark datasets of varying size, sparsity, and domain is analyzed. Tables II – V report results on MovieLens 100k, MovieLens 20M, Steam (10%), and Last.fm 360k (10%), respectively. Evaluation metrics include MAP@10, NDCG@10, Precision@10, Recall@10, MRR, and training time (in seconds).

A. Performance on MovieLens 100k

As shown in Table II**,** across all metrics, NGCF achieves the best performance on the MovieLens 100k dataset, followed closely by LightGCN, NeuMF, and EASE^R. These models benefit from either non-linear modeling capacity (NeuMF), high-order collaborative signal propagation (NGCF, LightGCN), or efficient reconstruction of item-item relationships (EASE^R). Notably, EASE^R, despite being a linear model with closed-form training, achieves top-3 performance in multiple metrics (e.g., *MRR* and *Recall@10*) while requiring negligible training time.

Among the VAE-based models, VAECF shows strong performance, outperforming deeper variational designs like BiVAECF. The non-variational SANSA model also performs competitively, though with slightly lower ranking accuracy. HPF ranks well in *MRR* and *Recall@10*, confirming its effectiveness in modeling implicit feedback with simpler priors.

TABLE II. PERFORMANCE COMPARISON ON THE MOVIELENS 100K DATASET. FOR EACH METRIC, BOLD INDICATES THE BEST-PERFORMING METHOD, AND UNDERLINED INDICATES THE SECOND-BEST

	MovieLens 100k					
Method	***MAP@10***	***NDCG@10***	***Precision@10***	***Recall@10***	***MRR***	***Training Time (s)***
PMF	0.0602	0.0924	0.0868	0.0388	0.2124	0.7072
MF	0.0471	0.0639	0.0653	0.0254	0.1439	0.1086
NeuMF	0.2591	0.3824	0.3231	0.2051	0.6180	67.8821
BPR	0.1329	0.2180	0.1912	0.1156	0.4233	0.5559
WBPR	0.1225	0.2189	0.1903	0.0932	0.4343	0.5196
IBPR	0.1456	0.1970	0.1807	0.1022	0.3642	173.0539
VAECF	0.2279	0.3439	0.2914	0.1839	0.5785	4.1308
BiVAECF	0.1987	0.2922	0.2576	0.1617	0.4925	8.7937
EASE^R	0.2441	0.3743	0.3035	<u>0.2135</u>	0.6404	0.1229
SANSA	0.1437	0.2798	0.2144	0.1401	0.5782	7.1746
NGCF	**0.2806**	**0.4070**	**0.3416**	**0.2228**	**0.6481**	2444.5783
LightGCN	<u>0.2630</u>	<u>0.3886</u>	<u>0.3244</u>	0.2062	<u>0.6433</u>	489.3929
HPF	0.2342	0.3527	0.2975	0.1858	0.5892	4.4192

Classical matrix factorization methods like MF and PMF perform relatively poorly across all metrics, indicating their limitations when modeling even moderately complex preference structures, even though the dataset has relatively high density. Meanwhile, BPR and WBPR offer competitive performance improvements over MF, though they are outperformed by their neural and graph-based successors.

In terms of efficiency, MF, PMF, and EASE^R are the fastest to train, with sub-second runtimes, making them appealing for real-time or low-resource applications. On the other end of the spectrum, NGCF and LightGCN require significantly longer training times, particularly NGCF, which incurs the highest computational cost.

B. Performance on MovieLens 20M

As detailed in Table III, on the large-scale MovieLens 20M dataset, EASE^R achieves the best overall performance across all metrics, outperforming both neural and probabilistic models. Its linear design

with closed-form optimization results in high accuracy. It tops *MAP@10*, *NDCG@10*, *Precision@10*, *Recall@10*, and *MRR*, while keeping training time relatively low (under 20 minutes), making it a practical choice for large datasets.

NeuMF and VAECF follow closely behind. NeuMF, with its deep neural architecture, shows strong ranking quality, especially in *NDCG@10* and *MRR*, but at a substantial computational cost, requiring over 6 hours to train. VAECF also performs well, especially in *Recall@10* and *MRR*, benefiting from variational inference to capture uncertainty in user preferences. However, deeper generative models like BiVAECF underperform significantly, indicating potential overfitting or optimization challenges at this scale.

In contrast, classical matrix factorization methods (MF, PMF) lag far behind across all metrics, reflecting their limited expressiveness in modeling user-item interactions in large-scale, sparse data.

It is important to highlight the absence of results for NGCF, LightGCN, and IBPR on this dataset. Their training was infeasible due to prohibitively long runtimes, pointing to scalability issues when these models are applied to large-scale datasets with dense user-item interactions.

TABLE III. PERFORMANCE COMPARISON ON THE MOVIELENS 20M DATASET. FOR EACH METRIC, BOLD INDICATES THE BEST-PERFORMING METHOD, AND UNDERLINED INDICATES THE SECOND-BEST. A DASH (–) DENOTES THAT THE MODEL COULD NOT BE TRAINED DUE TO EXCESSIVE TRAINING TIME

	MovieLens 20M					
Method	***MAP@10***	***NDCG@10***	***Precision@10***	***Recall@10***	***MRR***	***Training Time (s)***
PMF	0.0208	0.0491	0.0426	0.0220	0.1356	146.4725
MF	0.0162	0.0463	0.0404	0.0171	0.1258	3.3591
NeuMF	<u>0.2175</u>	<u>0.3392</u>	<u>0.2935</u>	<u>0.1682</u>	<u>0.5448</u>	21790.6354
BPR	0.0935	0.1797	0.1558	0.0707	0.3559	352.7249
WBPR	0.1311	0.2300	0.1948	0.1128	0.4341	320.7549
IBPR	-	-	-	-	-	-
VAECF	0.1874	0.2897	0.2423	0.1578	0.5016	4617.9240
BiVAECF	0.0859	0.1585	0.1475	0.0809	0.2937	14130.8773
$EASE^R$	**0.2554**	**0.4132**	**0.3446**	**0.2126**	**0.6556**	1165.1947
SANSA	0.1418	0.2520	0.2182	0.1199	0.4483	3116.5670
NGCF	-	-	-	-	-	-
LightGCN	-	-	-	-	-	-
HPF	0.1428	0.2421	0.2091	0.1080	0.4325	3810.1220

From an efficiency perspective, MF remains the fastest model to train (~3 seconds), while NeuMF, BiVAECF, and HPF incur high computational costs. This disparity reinforces the trade-off between modeling complexity and practical deployment in large-scale systems.

C. Performance on Steam (10%)

According to the results presented in Table IV, on the Steam dataset, which is characterized by extreme sparsity, the performance gap between traditional and modern recommender models becomes more pronounced. Among the evaluated methods, $EASE^R$ emerges as the most effective, demonstrating strong ranking quality across all evaluated metrics, including *MAP@10*, *NDCG@10*, *Precision@10*, *Recall@10*, and *MRR*. Despite its simple linear formulation and closed-form solution, $EASE^R$ effectively captures item-item relationships, making it particularly well-suited to handling sparse user feedback.

The pairwise ranking model BPR ranks as the second-best performer on this dataset, demonstrating robust effectiveness in sparse interaction settings. In contrast, its weighted variant WBPR consistently underperforms across evaluation metrics, suggesting that the sample reweighting mechanism fails to offer meaningful benefits under extreme data sparsity.

NeuMF and VAECF also deliver solid results across the full metric spectrum, showing their capacity to generalize from limited interactions through deeper architectures or latent variable modeling. However, the deeper variant BiVAECF struggles to match the performance of its simpler counterpart, suggesting that added complexity may not always translate to gains under severe data sparsity.

Bayesian models like HPF maintain reasonable performance, particularly in *Recall@10* and *MRR*, benefiting from probabilistic assumptions that accommodate sparse implicit feedback. Similarly, SANSA shows respectable results across most metrics despite its non-variational design, aligning closely with other middle-tier models.

Contrastively, traditional matrix factorization approaches like MF and PMF perform poorly across all key metrics, failing to extract meaningful patterns from such limited user histories.

It is also notable that certain computationally intensive methods like NGCF, LightGCN, and IBPR could not be trained within reasonable time constraints, reflecting a scalability issue when applied to datasets with large item spaces and minimal signal. In contrast, simpler models like EASE^R and BPR offer a favorable trade-off between accuracy and efficiency, making them attractive for real-world deployment scenarios where both speed and performance.

D. Performance on Last.fm 360k (10%)

Performance metrics summarized in Table V highlight that on the large-scale and highly sparse Last.fm dataset, which contains a very large number of items, SANSA achieves the best overall performance across ranking metrics, confirming its effectiveness in capturing user-item affinities even in low-density scenarios. Its non-variational design and relatively efficient training procedure make it a strong candidate for large-scale recommendation tasks.

TABLE IV. PERFORMANCE COMPARISON ON A 10% STRATIFIED SUBSAMPLE OF THE STREAM DATASET. FOR EACH METRIC, BOLD INDICATES THE BEST-PERFORMING METHOD, AND UNDERLINED INDICATES THE SECOND-BEST. A DASH (–) DENOTES THAT THE MODEL COULD NOT BE TRAINED DUE TO EXCESSIVE TRAINING TIME

Steam (10%)						
Method	***MAP@10***	***NDCG@10***	***Precision@10***	***Recall@10***	***MRR***	***Training Time (s)***
PMF	0.0019	0.0019	0.0005	0.0039	0.0024	5.5292
MF	0.0000	0.0001	0.0000	0.0003	0.0005	4.6947
NeuMF	0.0270	0.0282	0.0089	0.0498	0.0392	303.9657
BPR	0.0309	0.0316	0.0094	0.0575	0.0425	8.2983
WBPR	0.0051	0.0030	0.0009	0.0048	0.0080	1.9612
IBPR	-	-	-	-	-	-
VAECF	0.0298	0.0316	0.0100	0.0614	0.0410	297.3855
BiVAECF	0.0167	0.0151	0.0044	0.0279	0.0223	379.0551
EASE^R	**0.0408**	**0.0461**	**0.0126**	**0.0771**	**0.0577**	48.3324
SANSA	0.0191	0.0226	0.0066	0.0376	0.0300	6.6424
NGCF	-	-	-	-	-	-
LightGCN	-	-	-	-	-	-
HPF	0.0198	0.0214	0.0065	0.0389	0.0286	51.9678

TABLE IV. PERFORMANCE COMPARISON ON A 10% STRATIFIED SUBSAMPLE OF THE STREAM DATASET. FOR EACH METRIC, BOLD INDICATES THE BEST-PERFORMING METHOD, AND UNDERLINED INDICATES THE SECOND-BEST. A DASH (–) DENOTES THAT THE MODEL COULD NOT BE TRAINED DUE TO EXCESSIVE TRAINING TIME

Steam (10%)						
Method	***MAP@10***	***NDCG@10***	***Precision@10***	***Recall@10***	***MRR***	***Training Time (s)***
PMF	0.0019	0.0019	0.0005	0.0039	0.0024	5.5292
MF	0.0000	0.0001	0.0000	0.0003	0.0005	4.6947
NeuMF	0.0270	0.0282	0.0089	0.0498	0.0392	303.9657
BPR	0.0309	0.0316	0.0094	0.0575	0.0425	8.2983
WBPR	0.0051	0.0030	0.0009	0.0048	0.0080	1.9612
IBPR	-	-	-	-	-	-
VAECF	0.0298	0.0316	0.0100	0.0614	0.0410	297.3855
BiVAECF	0.0167	0.0151	0.0044	0.0279	0.0223	379.0551
EASE^R	**0.0408**	**0.0461**	**0.0126**	**0.0771**	**0.0577**	48.3324
SANSA	0.0191	0.0226	0.0066	0.0376	0.0300	6.6424
NGCF	-	-	-	-	-	-
LightGCN	-	-	-	-	-	-
HPF	0.0198	0.0214	0.0065	0.0389	0.0286	51.9678

Among variational approaches, both VAECF and BiVAECF demonstrate strong performance, with VAECF slightly outperforming its deeper variant. This suggests that in high-sparsity settings, model

depth does not necessarily translate into improved ranking accuracy, and simpler variational structures may generalize better.

Neural and graph-based methods, including NeuMF and LightGCN, also perform competitively, benefiting from their capacity to model complex preference structures. However, their training time is substantially longer, particularly in the case of LightGCN, which reflects a notable trade-off between predictive power and scalability.

HPF remains a reliable probabilistic model, maintaining a balance between accuracy and computational efficiency. On the other hand, BPR delivers moderate performance as a pairwise ranking method, while its weighted extension, WBPR, falls behind, reaffirming earlier observations about the limited benefit of its sample weighting strategy under sparse data conditions.

Notably, $EASE^R$ could not be applied to this dataset due to memory constraints. Its reliance on an explicit item-item similarity matrix makes it unsuitable for environments with extremely large item spaces. Similarly, NGCF and IBPR encountered computational limitations, further emphasizing the need for scalable methods when dealing with real-world, large-scale recommender systems.

Meanwhile, traditional matrix factorization models like MF and PMF fail to yield meaningful results, reflecting their inability to cope with severe data sparsity and complex interaction dynamics.

V. Conclusions and Future Work

The study provides a standardized evaluation highlighting the relative strengths and weaknesses of various CF methods in terms of accuracy, scalability, and robustness across datasets with differing characteristics.

Key Findings:

- *No Single Best Algorithm:* the performance of algorithms varied significantly across datasets.
- *For instance, NGCF performed* best on MovieLens 100k, while $EASE^R$ excelled on the larger MovieLens 20M and the sparse Steam dataset. SANSA showed top performance on the large and sparse Last.fm dataset. This underscores that model selection should be context-dependent, considering dataset size, sparsity, and specific evaluation metrics.
- *Effectiveness of Simpler Models:* surprisingly, simpler models like $EASE^R$, a linear autoencoder, demonstrated strong and sometimes superior performance, particularly on large-scale (MovieLens 20M) and extremely sparse (Steam) datasets. $EASE^R$ often provided a good balance of accuracy and efficiency.
- *Scalability Challenges:* several advanced models, including graph-based neural networks like NGCF and LightGCN, as well as IBPR, faced scalability issues, with training becoming infeasible on larger datasets like MovieLens 20M and Steam due to excessive runtimes. $EASE^R$ also faced memory constraints on datasets with extremely large item spaces, such as Last.fm.
- *Limitations of Traditional Methods:* classical matrix factorization methods (MF, PMF) generally performed poorly across most datasets and metrics, indicating their limitations in modeling complex preference structures from implicit feedback, especially in sparse environments.
- *Variational Autoencoders:* VAECF showed strong and consistent performance, particularly on MovieLens 100k and MovieLens 20M, and was competitive on Last.fm. However, deeper variational models like BiVAECF did not always offer performance gains and sometimes underperformed their shallower counterparts, especially on larger or sparser datasets.
- *Pairwise Ranking Models:* BPR showed robust performance, especially in sparse settings like the Steam dataset where it was the second-best performer. However, its weighted variant, WBPR, did not consistently offer improvements and sometimes underperformed, particularly in highly sparse scenarios.
- *Impact of Dataset Characteristics:* the sparsity and scale of the datasets significantly influenced model performance. Models that excelled on denser datasets did not always maintain their superiority on sparser or larger-scale data, and vice-versa. For example, NGCF and LightGCN performed well on MovieLens 100k but struggled with larger datasets.
- *This benchmark aims to inform* practitioners about the current state of CF for implicit feedback, helping to guide algorithm selection based on practical trade-offs between accuracy and efficiency.
- *Future Work.* The results and limits of this study suggest several possible directions for future research:
- *Expanding Model Coverage:* future benchmarks could include a wider array of recent and emerging CF techniques, including more advanced graph neural networks, transformer-based models, and hybrid approaches that combine CF with content-based or knowledge graph information.
- *Incorporating Temporal Dynamics:* this study explicitly excluded sequential recommender systems. A significant area for future work is to benchmark and compare these models against non-sequential CF methods, perhaps by developing

evaluation frameworks that can fairly assess both types of models or by focusing on datasets where temporal information is crucial.

- *Broader Range of Datasets and Implicit Signals:* evaluating models on an even more diverse set of datasets, representing different domains and types of implicit feedback (e.g., varying levels of noise, different interaction types beyond purchase/click/play counts), would provide a more holistic understanding of model generalizability.
- *Hyperparameter Optimization Strategies:* while a cross-validation approach was used, the hyperparameter search space was limited. Future studies could explore more extensive and adaptive hyperparameter optimization techniques to ensure each model achieves its peak potential.
- *Investigating Algorithmic Components:* a deeper dive into why certain architectural choices (e.g., linearity in $EASE^R$, simplification in LightGCN over NGCF, depth in VAEs) lead to better or worse performance in specific contexts could yield valuable insights for designing new, more effective recommender algorithms.

REFERENCES

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of Netnews," in *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW)*, 1994, pp. 175–186. https://doi.org/10.1145/192844.192905

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295. https://doi.org/10.1145/371920.372071

[3] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Pisa, Italy, 2008, pp. 263–272, https://doi.org/10.1109/ICDM.2008.22.

[4] C. Réda, J.-J. Vie, and O. Wolkenhauer, "Comprehensive evaluation of pure and hybrid collaborative filtering in drug repurposing," *Sci. Rep.*, vol. 15, no. 1, p. 2711, Jan. 2025, https://doi.org/10.1038/s41598-025-85927-x.

[5] R. Kwieciński, T. Górecki, A. Filipowska, and V. Dubrov, "Job recommendations: Benchmarking of collaborative filtering methods for classifieds," *Electronics*, vol. 13, no. 15, Art. no. 3049, Aug. 2024, https://doi.org/10.3390/electronics13153049.

[6] S. Aramanda, M. H. M. Krishna Prasad, and P. V. L. Suvarchala, "A comparison analysis of collaborative filtering techniques for recommender systems," in *Innovations in Computer Science and Engineering*, S. C. Satapathy, A. Joshi, N. Modi, and N. Pathak, Eds. Singapore: Springer, 2021, pp. 247–254.

[7] A. F. M. S. Islam et al., "Comparative analysis of collaborative filtering recommender system algorithms for e-commerce," *J. Auton. Intell.*, vol. 7, no. 2, pp. 1–17, Nov. 2023. https://doi.org/10.32629/jai.v7i2.1182

[8] J. Zhu et al., "BARS: Towards open benchmarking for recommender systems," *arXiv preprint arXiv:2205.09626*, 2022.

[9] W. X. Zhao et al., "RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2021, pp. 4653–4664. https://doi.org/10.1145/3459637.3482016

[10] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 21st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Red Hook, NY, USA: Curran Associates Inc., 2007, pp. 1257–1264.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, 2009. https://doi.org/10.1109/MC.2009.263

[12] X. He et al., "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182. https://doi.org/10.1145/3038912.3052569

[13] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty in Artificial Intelligence (UAI)*, 2009, pp. 452–461.

[14] Z. Gantner, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Personalized ranking for non-uniformly sampled items," in *Proc. KDD Cup 2011*, vol. 18, JMLR W&CP, 2011, pp. 231–247.

[15] D. D. Le and H. W. Lauw, "Indexable Bayesian personalized ranking for efficient top-k recommendation," in *Proc. 26th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2017, pp. 1389–1398, https://doi.org/10.1145/3132847.3132913.

[16] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web Companion (WWW Companion)*, 2015, pp. 111–112. https://doi.org/10.1145/2740908.2742726

[17] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. 27th Int. Conf. World Wide Web (WWW)*, 2018, pp. 689–698. https://doi.org/10.1145/3178876.3186150

[18] W. Lee, K. Song, and I.-C. Moon, "Augmented variational autoencoders for collaborative filtering with auxiliary information," in *Proc. 26th ACM Int.*

Conf. Inf. Knowl. Manag. (CIKM), 2017, pp. 1139–1148, https://doi.org/10.1145/3132847.3132972.

[19] Q.-T. Truong, A. Salah, and H. W. Lauw, "Bilateral variational autoencoder for collaborative filtering," in *Proc. 14th ACM Int. Conf. Web Search Data Min. (WSDM)*, 2021, pp. 239–247. https://doi.org/10.1145/3437963.3441759

[20] H. Steck, "Embarrassingly shallow autoencoders for sparse data," in *Proc. 28th Int. World Wide Web Conf. (WWW)*, 2019, pp. 3251–3257. https://doi.org/10.1145/3308558.3313710

[21] G. Jeunen, S. Basu, and C. A. Sutton, "Scalable approximate non-symmetric autoencoder for collaborative filtering," in *Proc. 46th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. (SIGIR)*, 2023, pp. 1797–1807.

[22] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. (SIGIR)*, 2020, pp. 165–174. https://doi.org/10.1145/3331184.3331267

[23] X. He et al., "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. (SIGIR)*, 2020, pp. 639–648, https://doi.org/10.1145/3397271.3401063.

[24] P. K. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical Poisson factorization," in *Proc. 31st Conf. Uncertainty in Artificial Intelligence (UAI)*, 2015, pp. 326–335.

[25] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. 13th ACM Int. Conf. Web Search Data Min. (WSDM)*, 2018, pp. 197–205. https://doi.org/10.1109/ICDM.2018.00035

[26] F. Sun et al., "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2019, pp. 1441–1450. https://doi.org/10.1145/3357384.3357895

[27] A. Salah, Q.-T. Truong, and H. W. Lauw, "Cornac: A comparative framework for multimodal recommender systems," *J. Mach. Learn. Res.*, vol. 21, no. 1, Art. no. 95, pp. 3803–3807, Jan. 2020.

[28] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Art. no. 19, Dec. 2015, https://doi.org/10.1145/2827872.

[29] A. Kozyriev, "Game recommendations on Steam," *Kaggle*, 2020. [Online]. Available: https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam. [Accessed: May 20, 2025].

[30] O. Celma, "Last.fm 360k dataset," Music Technology Group, Universitat Pompeu Fabra. [Online]. Available: https://www.upf.edu/web/mtg/lastfm360k. [Accessed: May 20, 2025]

Received February 26, 2025

Pyshnograiev Ivan. ORCID 0000-0002-3346-8318. Candidate of Physical and Mathematical Sciences. Associate Professor.
Department of Artificial Intelligence, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Education: NTUU "KPI", Kyiv, Ukraine (2012).
Research interests: modeling the behavior of complex systems.
Publications: more than 40.
E-mail: pyshnograiev@gmail.com

Shyralliev Anar. ORCID 0009-0007-4140-5476. Postgraduate Student.
Department of Artificial Intelligence, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Education: NTUU "KPI", Kyiv, Ukraine (2019).
Research interests: machine learning and artificial intelligence.
Publications: 2.
E-mail: anarshyraliiev@gmail.com

І. О. Пишнограєв, А. Е. Ширалієв. Комплексний бенчмарк методів колаборативної фільтрації на наборах даних з неявним зворотним зв'язком

Колаборативна фільтрація є фундаментальною технікою в сучасних рекомендаційних системах, особливо при роботі з даними з неявним зворотного зв'язком, такими як кліки, покупки або поведінка під час прослуховування. Незважаючи на велику кількість моделей КФ, включаючи класичні, імовірнісні та нейронні підходи, бракує стандартизованої, масштабної оцінки ефективності на різних наборах даних. У цьому дослідженні представлено комплексний емпіричний бенчмарк 13 алгоритмів колаборативної фільтрації, що охоплює матричну факторизацію, попарне ранжування, варіаційні та неваріаційні автокодери, нейронні моделі на основі графів та ймовірнісні методи. Використовуючи чотири репрезентативні набори даних з неявним зворотним зв'язком з різних доменів, ми оцінюємо моделі за єдиним експериментальним протоколом з

використанням метрик на основі ранжування (*MAP@10, NDCG@10*, *Precision@10*, *Recall@10*, *MRR*), а також аналізуємо ефективність навчання. Результати показують, що нейронні архітектури, такі як NeuMF, VAECF і LightGCN, демонструють високу продуктивність у щільних і помірно розріджених сценаріях, але можуть стикатися з обмеженнями масштабованості на великих наборах даних. Простіші моделі, такі як $EASE^R$ та BPR, часто забезпечують вдалий баланс між продуктивністю та обчислювальними витратами. Цей бенчмарк пропонує практичні інсайти щодо компромісів у сучасних методах CF і слугує орієнтиром для подальших досліджень у сфері неявних систем рекомендацій.
Ключові слова: колаборативна фільтрація; неявний зворотний зв'язок; рекомендаційні системи; бенчмаркінг; ранжування.

Пишнограєв Іван Олександрович. ORCID 0000-0002-3346-8318. Кандидат фізико-математичних наук. Доцент.
Кафедра штучного інтелекту, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: НТУУ «КПІ», Київ, Україна (2012).
Напрямок наукової діяльності: моделювання поведінки складних систем.
Кількість публікацій: більше 40.
E-mail: pyshnograiev@gmail.com

Ширалієв Анар Ельдар огли. ORCID 0009-0007-4140-5476. Аспірант.
Кафедра штучного інтелекту, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: НТУУ «КПІ», Київ, Україна (2019).
Напрямок наукової діяльності: машинне навчання та штучний інтелект.
Кількість публікацій: 2.
E-mail: anarshyraliiev@gmail.com