What Are We Optimizing For? A Human-centric Evaluation of Deep Learning-based Movie Recommenders

Ruixuan Sun[1,*], Xinyi Wu[2], Avinash Akella[1], Ruoyan Kong[1], Bart Knijnenburg[3] and Joseph A. Konstan[1]

[1]Grouplens Research, University of Minnesota
[2]Institute for Data, Systems, and Society (IDSS), Massachusetts Institute of Technology
[3]School of Computing, Clemson University

Abstract

In the past decade, deep learning (DL) models have gained prominence for their exceptional accuracy on benchmark datasets in recommender systems (RecSys). However, their evaluation has primarily relied on offline metrics, overlooking direct user perception and experience. To address this gap, we conduct a human-centric evaluation case study for four leading DL-RecSys models in the movie domain. We test how different DL-RecSys models perform in personalized recommendation generation by conducting a survey study with 445 real active users. We find some DL-RecSys models to be superior in recommending novel and unexpected items but weaker in diversity, trustworthiness, transparency, accuracy, and overall user satisfaction compared to classic collaborative filtering (CF) methods. Qualitatively, we confirm with real user quotes that accuracy plus at least one other attribute is necessary to ensure good user experience, while their demands for transparency and trust cannot be neglected. Based on our findings, we discuss future human-centric DL-RecSys design and optimization strategies.

Keywords

Recommender Systems, Human-centered AI, Explainable Artificial Intelligence

1. Introduction

For modern recommender systems (RecSys), deep learning (DL) models are commonly recognized as state-of-the-art (SOTA) solutions, usually credited to their high accuracy scores (e.g., RMSE, HR, recall, MRR, NDCG) on benchmark datasets. However, how well such standards transfer to end user-related values, such as recommendation transparency [1, 2], trustworthiness [3, 4] and user satisfaction [5, 6] is still an open question [7].

Referring to prior works that evaluated RecSys algorithms beyond offline scores from users' perspective on collaborative filtering or matrix factorization models [8, 9], we design this study to comprehensively assess the performance of four representative types of DL models from an open-sourced recommender model leaderboard. Our human-centric framework consists of both offline benchmark dataset measurement and personalized online recommendation feedback collected from 445 real users of an online movie recommender. Inspired by previous human-centric approaches [8, 9], we focus on seven human-centric metrics including *Novelty, Diversity, Serendipity, Accuracy, Trustoworthiness, Transparency*, and their overall *Satisfaction* in real user evaluation.

With this study design, we aim to answer the following two research questions:

RQ1: *How do different DL-RecSys models perform compared with each other and classic CF methods, as evaluated by our human-centric framework?*

RQ2: *What are some common human-centric value requests from users that future DL-RecSys designers should consider?*

IntRS'24: Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, October 18, 2024, Bari (Italy)
*Corresponding author.
sun00587@umn.edu (R. Sun); xinyiwu@mit.edu (X. Wu); akell011@umn.edu (A. Akella); kong0135@umn.edu (R. Kong); bartk@clemson.edu (B. Knijnenburg); konstan@umn.edu (J. A. Konstan)
https://sophiasun0515.github.io/ (R. Sun)
0000-0003-4653-0384 (R. Sun)

In the rest of the paper, we first discuss related work, then provide a high-level overview of our research methods and the four deep learning models. After that, we share the model-wise performance comparison based on offline and real-user input results, also the qualitative data analysis of real user evaluation data. Finally, we consolidate and discuss all findings and propose human-centric design implications for future DL-RecSys studies.

2. Related Work

2.1. Deep Learning and its Evaluation in RecSys

Deep Neural Networks were pervasively used in the RecSys domain for its nature to abstract user-item interaction patterns and effectively learn the representation of large amount of input data [10]. Depending on the choice of optimization, there exists a diverse set of architectural paradigms. Multilayer Perceptron (MLP) [11] that served as a basic approximation technique relies on a feed-forward neural network that consists of hidden layers of non-linear transformations of features. Graph Convolutional Neural Network (GNN) is one step further as it can model real-world network structure, such as social networks or user-item relationships with its graph structure and conduct link prediction for recommendation tasks [12]. Recurrent Neural Network (RNN) is appropriate when it comes to sequential recommendation [13] for its ability to remember former computations in memory. Transformers work well in session-based recommendation tasks [14] with its self-attention mechanism. As Dacrema et al. points out, systematic studies are needed for a fair evaluation of DL-RecSys models to truly assess the progress they bring to the field of recommender systems [15]. In prior literature, DL-RecSys model benchmarking was primarily offline, centering around prediction accuracy [16] and training time [17].

2.2. User Perspective in Recommender Systems

User experience has been a critical aspect in evaluating the success of recommendations ever since the field started. Konstan and Riedl suggested that the only reliable way to measure the RecSys behavior in a natural context is through a long-term field experiment [18]. Munawar et al. identified that subjective recommender system aspects, such as perceived quality and effectiveness, can be significant factors in user satisfaction [19]. Similarly, Knijnenburg and Willemsen and Pu et al. also proposed a user-centric evaluation framework to assess recommender systems with user experiments and statistical analysis [20, 8]. Kunkel et al. evaluated differences of trustworthiness between personal and impersonal recommendations with real human explanations [21]. In industry, RecSys practitioners mainly evaluated user values from their engagement [22], long-term satisfaction [23], and privacy [24] as complements to accuracy or monetization metrics.

3. Research Methods

Our overall study design is split into two parts: 1) In phase one, we reproduced the four DL-RecSys models with the MovieLens-1M (ML-1M) dataset and evaluated their performances of novelty, diversity, and serendipity based on formulas defined in previous literature; 2) In phase two, we generated personalized recommendation with each DL-RecSys model for over 3000 active users of an online movie recommender. We then designed a survey with top recommendation lists from each model and sent to users for subjective evaluation including both Likert-scale questions and free-form text input. Detailed user evaluation flow can be found in Fig. 1.

3.1. Deep Learning and Baseline Models

We select four distinct DL models from the SOTA leaderboard based on their accuracy performances with the major benchmark dataset MovieLens-1M (ML-1M)[1]:

[1]https://paperswithcode.com/sota/collaborative-filtering-on-movielens-1m

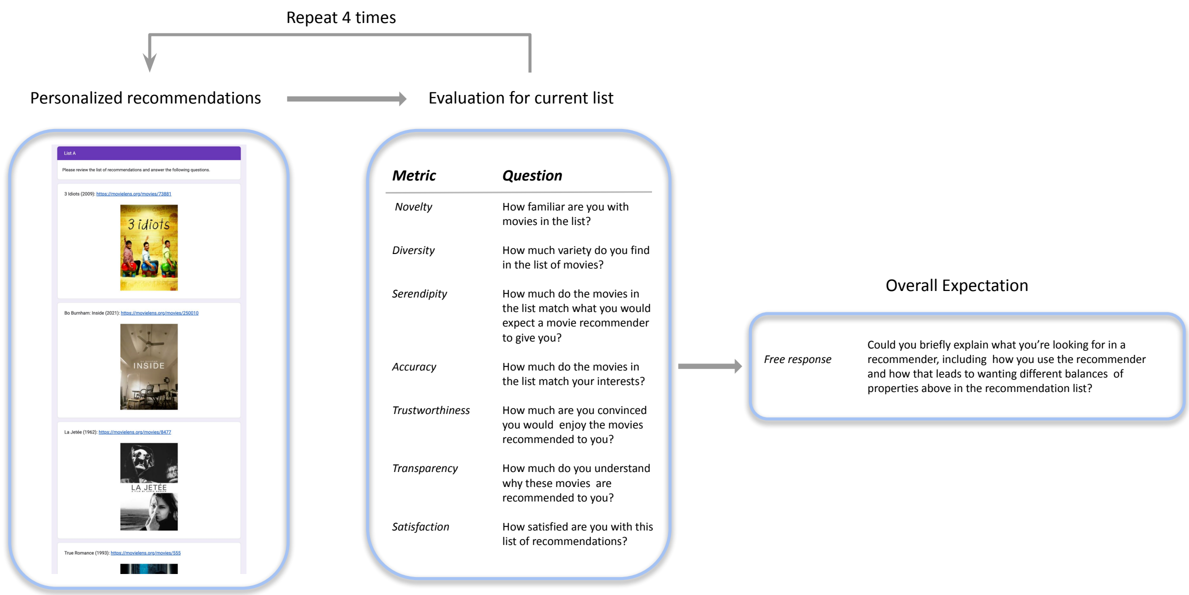

Figure 1: User evaluation flow and survey questions.

- ***Neural Collaborative Filtering (NCF)*** [25] is one of the early seminal works introducing DL methods to recommender systems. It employs deep learning for collaborative filtering (CF) by replacing the inner product with a neural network.
- ***BERT4Rec*** [26] is a deep bidirectional self-attention model to learn the representations for users' historical behavior sequences. One can trace the reasons for recommendation by checking the attention scores and finding the most important historical item in prediction.
- ***SSE-PT*** [27] is a sequential-based personalized transformer. Similar to BERT4Rec, SSE-PT enjoys interpretability by allowing one to check the attention scores and find the most important historical item in prediction.
- ***GLocal-K*** [28] focuses on feature extraction by generalizing and representing a high-dimensional sparse user-item matrix into a low-dimensional space. Efficiency in data sparsity is the key advantage of GLocal-K.

For each model, we take the code from the official open-source repository provided by the original authors and train the model on the dataset we collect for real users (detailed in section 3.3). For fair comparison, we adopt the optimal hyperparameters for ML-1M dataset reported in their original papers. All code repository links and hyperparameters can be found in the Appendix. The two baseline models we choose are k-nearest neighbor user-based collaborative filtering (with min k = 2) and funk SVD(with factors=10 and epochs=20) models.[2]

3.2. Benchmark Dataset Evaluation

Based on previous empirical analyses on human-centric evaluation [29], we define the three objective metrics for a list of recommendation R as follows:

- *Novelty* as the normalized average item self-information, where U stands for all users in the dataset [29]:

$$\text{Novelty}(R) = \frac{1}{-\log_2 \frac{1}{|U|} \cdot |R|} \sum_{i \in R} -\log_2 \frac{|\{u \in U, r_{ui} \neq \emptyset\}|}{|U|} \tag{1}$$

[2]The two CF models can be found in the Surprise library at https://surpriselib.com/.

- *Diversity* as the average pairwise distance between items' *Tag Genome* information [30], which is a vector with normalized relevance scores indicating different characteristics users used to describe the movie. Pearson correlation [31] is used as the distance metric:

$$\text{Diversity}(R) = \frac{\sum_{i \in R} \sum_{j \in R \setminus \{i\}} \text{dist}(i,j)}{|R|(|R|-1)} \tag{2}$$

- *Serendipity* as the content-based surprise metric that measures the average minimal distance between each candidate's *Tag Genome* [30] distance to its closest neighbor among user's previously rated movies. Cosine similarity is used as the distance metric, and P stands for the most recent 100 rated movies (if they rated less than 100 movies, P equals to the actual number of rated movies of the user):

$$\text{Serendipity}(R) = \frac{1}{|R|} \sum_{i \in R} \min_{j \in P} \text{dist}(i,j) \tag{3}$$

We then calculate the human-centric performances of each DL model with the formula above. To ensure convergence, we train each model for 300 epochs and plot their performance every 10 epochs. As a comparable baseline, we select the classical collaborative filtering (CF)-based funk singular value decomposition (SVD) with factors as 10 and epochs as 20. We apply 5-fold cross-validation to the baseline training and tested with only unrated movies for each user. The final score is calculated as the mean of all 5 folds. For computational efficiency, all metrics are just evaluated with the top 8 items on the list of recommendations.

3.3. Real User Evaluation and Survey Design

The participants recruited for this study were from MovieLens (https://movielens.org/), an academic-running online movie recommender system with thousands of active users. Due to the strict user privacy policy of the website, we select participants mainly based on their activity level (logged in over 12 times and rated over 20 movies) in the year of 2022. For the training set, we then collect those active users' ratings in the three calendar years before the experiment (from 2020-01-01 until 2022-12-31). To ensure the minimal popularity of movies, we also filter out those movies with less than 20 user ratings. The dataset contains 3,537 users, 7,462 movies, and 983,376 ratings. With the new real user dataset, we generate a personalized list containing the top-recommended movies for each user using the four DL models and two baseline CF models, respectively. To avoid making the questionnaire too long to exceed the general user's attention span, we only randomly assign 3 DL-generated recommendation lists and 1 CF list to each user.

Every user gets a Google form survey of 6 pages. The first page describes the purpose of this study ("test different personalization models"). After that, users proceed to 4 pages of recommendations that each contains a list of the top 12 movies (we ran pilot tests and considered that to be an adequate number for users to consume and make judgments) generated by one of the three DL models or one CF model. Each recommendation item contains the movie name, release year, and detailed MovieLens link that users can visit to check for details. After reviewing each page of recommendations, users are asked to fill in 7 human-centric questions outlined in Fig. 1. Our design and phrasing of questions was inspired by a series of previous evaluation and user study works [32, 33, 34, 20] that include both item attributes (e.g. how novel or diverse a list of recommendations look like) and subjective human perception (e.g. how transparent or trustworthy the results are), along with a summative user experience indicator (i.e. satisfaction). At the end, we also provide an optional text field for users to share their expected recommendation attributes.

All questions are designed on a 5-point Likert scale except for free text responses. Specifically, the first question on the third list of recommendations is designed as an attention check. It is a reverse-scaled 5-point Likert question asking the same *Satisfaction* question in a different phrase: *"How much do you like the list of recommendations?"* In the analysis stage, we compare the reversed response between this check question and the actual satisfaction question on the third list of recommendations from all users with a Mann-Whitney-Wilcoxon test and confirm they are not statistically different ($p = 0.664$), meaning that general user attention during the survey is high.

We sent out surveys via email to 3,537 qualified active users in 7 batches between April to June 2023, with one week between consecutive batches. In the email, we emphasized that filling in this survey was voluntary with no incentives, and users had the right to exit at any time during their participation. We then collected user responses after two weeks of the survey distribution. Overall, 3,172 out of 3,537 surveys were successfully delivered to users' email inboxes, and 445 of them replied, making the final response rate as **14.03%**.

4. Results

4.1. Offline Performance

In Fig. 2, we see that in terms of novelty, only the Bert4Rec model constantly beat the baseline SVD performance, with NCF results being the lowest. As for diversity, most of the DL models beat the baseline model, with Bert4Rec leading the board. Serendipity-wise, all DL models were not hitting the baseline SVD performance. Among all models, GLocal-K performs the most volatile and has no clear pattern, while the rest are generally stable among all epochs, which is expected due to the fact that they were not designed to optimize for the human-centric metrics. With this preliminary observation, we extend the human evaluation to invite real users to share their judgments on the recommendation quality of these DL models.

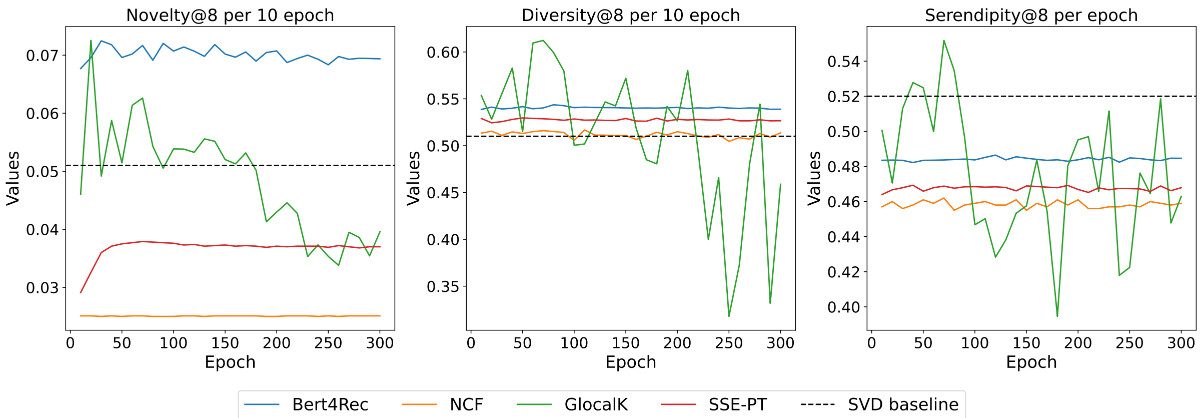

Figure 2: Offline performance on ML-1M dataset of top-8 movies generated for each user with deep-learning and baseline models.

4.2. Individual Model Performance

We report the mean and standard deviation (STD) along with pairwise statistical significant difference of all model-wise user evaluation questions (see Fig. 1) in Table 1. We observe that GLocal-K is the top performer in *Novelty*, while SSE-PT wins in *Serendipity*. Without CF baselines, NCF perfroms the best in terms of delivering diverse, trustworthy, transparent, accurate recommendations, while SSE-PT and BERT4Rec perform worst on *Trustworthiness*, *Transparency*, *Accuracy*, and *Satisfaction*.

> **RQ1**: *How do different DL-RecSys models perform compared with each other and classic CF methods, as evaluated by our human-centric framework?*

With both offline and real user evaluation comparison, we see that some DL models such as GLocal-K outperform in recommending novel items, and SSE-PT produced a good set of unexpected items for real users. While the offline evaluation indicates a good diversity performance for most of DL models,

	NCF(n=337)	SSE-PT(n=317)	BERT4Rec(n=340)	GLocal-K(n=334)	SVD(n=208)	UU-CF(n=237)
Novelty	3.104±1.104(***)	2.535±0.955(***)	2.974±1.119(***)	**3.611±1.010**	2.755±1.143(***)	2.827±1.146(***)
Diversity	3.656±0.985(***)	3.972±0.836(*)	3.657±0.983(***)	3.428±0.992(***)	3.957±0.934	**4.004±0.916**
Serendipity	2.504±0.897(***)	**3.252±1.063**	2.674±1.010(***)	2.361±0.853(***)	2.611±0.899(***)	2.574±1.000(***)
Trustworthiness	3.473±1.042	2.877±1.035(***)	3.095±1.012(**)	3.144±0.996(**)	**3.495±0.863**	3.228±1.024(*)
Transprency	3.516±1.066	2.397±1.212(***)	2.926±1.132(***)	3.197±0.980(**)	**3.587±0.928**	3.245±1.065(*)
Accuracy	3.214±1.201	2.817±0.986(***)	2.808±1.047(***)	2.875±1.125(***)	**3.284±1.228**	3.013±1.152(*)
Satisfaction	3.458±1.030	3.132±1.084(**)	3.118±0.974(*)	3.144±1.003(***)	**3.500±0.896**	3.160±1.017(**)

Table 1
Means and STDs for DL model recommendation evaluation. For each metric, the highest value is highlighted in bold, and a path model [35] from MPlus [36] is applied between each of the rest of the model's data to the one with the highest value for statistical significance assessment. For the last 4 metrics where NCF shows no significant difference from SVD, we tested with a separate path model and confirmed NCF has a significant difference from the other three DL models. Asterisk (*) indicates p-val between models: * for $p < .05$, ** for $p < .01$ and *** for $p < .001$.

they are worse at producing a diverse set of items in real user evaluation. Moreover, real users report those models struggle with gaining user trust or transparency, and most importantly, matching user personalized interest and achieving high satisfaction.

4.3. Qualitative Analysis

In the last section of real user evaluation, we look into users' commonly requested recommender properties and the balance between them. In total, we have 294 users who shared their preferences in the free response. Then, three researchers followed the grounded theory method (GMT) to conduct open coding, inductive thematic analyses, and affinity map building of different clusters on codes with similar meanings [37]. The process iterated until all inputs were clustered, and no ambiguity or disagreement emerged. Selected user quotes and their mapping recommender properties are displayed in Table 2. While many users only mentioned about their preference for accurate recommendations (N=52), we also see a great portion of others requesting the following three types of RecSys properties: 1) Urge for a good balance between accuracy with other metric(s); 2) Use of specific properties to build trust with the recommender; and 3) Demand for recommendation transparency. We answer RQ2 with our findings:

> **RQ2**: *What are some common human-centric value requests from users that future DL-RecSys designers should consider?*

Accuracy + X. The largest theme under this cluster is about the balance between accuracy and novelty (N=54), like P28 shared: *"I use it primarily to find out about movies I hadn't considered that closely match the kinds of movies I like."* The next is surrounding accuracy with diversity (N=12). For example, P39 said: " *"I want the recommender to find movies I'm not familiar with that it thinks I will like. I want broad recommendations across lots of genres, and time (both old and new movies)."* The third one blends accuracy with serendipity (N=10) – P234: *"I want a recommender that would challenge my tastes without offending me."*

Trust builder. Another big theme users discuss is to gain well-grounded recommendations (N=21). As P290 shares, *"I want to get reliable recommendations of movies that I wouldn't have come across otherwise."* One step further, many users also mention other properties they relate to building trust with the recommender, such as accuracy (N=10), *"I want the recommender to be adapted to my tastes so I can have a big level of confidence that I will enjoy the movies listed (P46)"*. P275 claim that their trust is built upon serendipity, *"I would even go as far to say I 'trust' or enjoy MovieLens specifically because I can't tell where the recommendation came from. I think when it's traceable that's what feels fake or mechanical. "*

Demand for transparency. The third cluster we identify for preferred recommendation value is transparency (N=27). Some users appreciate more explanation about the recommendation, like P286 says, *"...I would also love to see just a tiny bit more info on the films themselves, most importantly*

Participant	*Property*	*Input*
P6	*Trustworthiness, Diversity*	"I want a recommender that I can rely on to pick movies to watch. I want to see it recommend a variety of different genres and styles that introduce me to new movies I wouldn't have come across on my own but that I enjoy or can learn from or otherwise appreciate."
P18	*Diversity*	"Usually I've different moods at that situation and not that many movies will be the fit, so I guess the best would be getting a good variety of movies that are really not similar between them"
P34	*Transparency*	"If I am shown something that appears to be very different for me, I'm interested to understand why the movie was selected for me - that can help me decide if I agree and want to watch it."
P43	*Serendipity*	"A black box that spits out movies I'll enjoy. When I don't have a good movie in mind, I sort the MovieLens list by predicted rating, and pick one of the top ones that looks interesting"
P68	*Accuracy*	"To recommend movies for me to watch. I don't care how it decides this. I want to watch great stories with good actors that are well-directed."
P96	*Trustworthiness*	"I use the recommenders as double-check devices: I find a movie; if I find it interesting I check the rating given by the recommender. If it's good, I watch it. But I wouldn't trust the recommender on 'blind' recommendations."
P146	*Accuracy, Trustworthiness, Diversity*	"I like to see movies recommended to me which I have already seen. This mostly just shows me that the algorithm is in track and I can trust the movies it's shown me that I don't know. It would also be good to see a different set of recommendations each time I visited the website."
P186	*Transparency, Accuracy, Diversity*	"I'm looking for something to be able to make sense of the reasons why I like the movies that I like, and to not be afraid of recommending me strange niche movies. But at the same time, if possible, to not recommend me things that I clearly do not watch, while keeping variety."
P86	*Gap between AI and human*	"I'm expecting a recommender to find movies that have good ratings from other users who's ratings are similar to mine. I don't want recommendations based on information about the movie. Maybe someday AI will be capable enough to watch movies and guess what I like about them, but till then only other users have that information."

Table 2
Examples of user input about important properties of recommendations.

the writer(s). That would be a good tool to watch films from certain people...It would be cool if users had a better idea of what was going on under the hood as far as the recommendations go." Some more proactive users request control over what can be generated, like P9 shares: *"I want to have control over recommendations. Sometimes I am into fresh thrillers and therefore only want to see thrillers released after a certain year. Sometimes I would like to explore different genres of movies and therefore want to see very diverse recommendations."*

5. Discussion and Future Work

In this section, we consolidate our findings and discuss the state of DL-RecSys models from a human-centered perspective as well as opportunities for future design.

5.1. Current State of DL-RecSys models

While DL-RecSys models perform well in offline accuracy metrics, in our study, we find that they are not as stable when evaluated with pre-defined *Novelty*, *Diversity*, or *Serendipity* measures. As for real users, only NCF performs relatively equal in terms of generating transparent, accurate, and trustworthy recommendations compared to baseline CF models. It also has the highest performance in terms of general user satisfaction among all DL models. Since the other three DL models we assessed did not get close to CF in most metrics other than novelty and serendipity, there are still a lot of work to do with DL-RecSys user-centric improvement.

In particular, though some DL models have superior performance in terms of serendipity, prior work by Kotkov et al. suggested that higher serendipity in movies did not necessarily lead to higher user satisfaction [38]. In our case, the high serendipity performance on SSE-PT also did not produce corresponding high satisfaction in that particular model. For future DL-RecSys work focusing on serendipitous recommendation, it will be helpful to test carefully with real users to explore the right balance of serendipity, instead of indiscriminately optimizing for a higher value.

Overall, it is challenging to give specific optimization plans since different DL models have subtle and complex design choices, but by looking into some specific user cases, we summarize some generalized strategies researcher can consider for future development. One major demand from users asking for more transparency in algorithms can benefit from generating item descriptions with more personalized context, like P132 shares: *"I expect the recommender to recommend me movies that I haven't seen but would enjoy. Pointing out why it was licked would help me select a movie to be watched"*. Thus, incorporating personalized recommendation explanations with new technique such as large language models [39, 40, 41] can be promising future direction to explore. In addition, trustworthiness can be built with more than accuracy in recommendations, as P6 and P186 shared in Table 2. We suggest future practitioners also look into improving diversity and transparency, and supporting more user control [42] to ensure a higher user trust.

5.2. Limitations and Future Directions

We want to clarify again that since the focus of our study is a human-centric evaluation of existing DL-RecSys models, we used the optimal values reported by the authors in the corresponding paper or repository and did not tune the hyperparameters of chosen DL models to optimize for offline metrics. Given that this study uncovers critical human-centric metrics previously overlooked in conventional measurement approaches, we believe our findings carry significant meanings in terms of future optimization directions in DL-RecSys.

We recognize some limitations of this study: 1) We only chose DL-based models based on their performance of movie recommendation and collected real user feedback in the movie domain, without further expansion to other application fields; 2) We only tested personalized recommendations with active users, and did not generalize how those DL-RecSys models perform with cold-start users; 3) More comprehensive analysis can be done to further explore the latent relationships between different user perception values in DL-RecSys. We believe all the above points can be interesting future works. For example, researchers can conduct cross-domain studies, select both new and old users as test subjects, and run more complicated path analysis or structural equation modeling (SEM) to quantify how users' perception of each recommendation attributes impact each other.

6. Conclusion

In this study, we investigate how four DL-RecSys models perform under multi-dimensional human-centric evaluation with a movie recommendation case study. By conducting both offline objective metric evaluation on a benchmark dataset and producing personalized top recommendation lists and evaluating them with real user feedback, we find that sequential and kernel-based DL-RecSys models are superior in recommending novel and serendipitous items while underperforming classic CF models in user-perceived diversity, accuracy, trust, transparency, and general satisfaction. We also analyze users' qualitative input, revealing their requests for beyond-accuracy recommendation attributes and different elements they use to build trust with the system. We hope this case study can serve as a new perspective on evaluating and optimizing future DL-RecSys models under a human-centric framework.

References

[1] X. Chen, Y. Zhang, J.-R. Wen, Measuring" why" in recommender systems: A comprehensive survey on the evaluation of explainable recommendation, arXiv preprint arXiv:2202.06466 (2022).

[2] N. Liu, Y. Ge, L. Li, X. Hu, R. Chen, S.-H. Choi, Explainable recommender systems via resolving learning representations, in: Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 895–904.

[3] M. Hong, J. J. Jung, D. Camacho, Grsat: a novel method on group recommendation by social affinity and trustworthiness, Cybernetics and Systems 48 (2017) 140–161.

[4] G. Lenzini, Y. van Houten, W. Huijsen, M. Melenhorst, Shall i trust a recommendation? towards an evaluation of the trustworthiness of recommender sites, in: Advances in Databases and Information Systems: Associated Workshops and Doctoral Consortium of the 13th East European Conference,

ADBIS 2009, Riga, Latvia, September 7-10, 2009. Revised Selected Papers 13, Springer, 2010, pp. 121–128.
[5] T. T. Nguyen, F. Maxwell Harper, L. Terveen, J. A. Konstan, User personality and user satisfaction with recommender systems, Information Systems Frontiers 20 (2018) 1173–1189.
[6] P. Li, M. Que, Z. Jiang, Y. Hu, A. Tuzhilin, Purs: personalized unexpected recommender system for improving user satisfaction, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 279–288.
[7] Z. Fu, X. Niu, M. L. Maher, Deep learning models for serendipity recommendations: A survey and new perspectives, ACM Computing Surveys (2023).
[8] P. Pu, L. Chen, R. Hu, Evaluating recommender systems from the user's perspective: survey of the state of the art, User Modeling and User-Adapted Interaction 22 (2012) 317–355.
[9] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems, User modeling and user-adapted interaction 22 (2012) 441–504.
[10] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM computing surveys (CSUR) 52 (2019) 1–38.
[11] M. W. Gardner, S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric environment 32 (1998) 2627–2636.
[12] R. v. d. Berg, T. N. Kipf, M. Welling, Graph convolutional matrix completion, arXiv preprint arXiv:1706.02263 (2017).
[13] T. Donkers, B. Loepp, J. Ziegler, Sequential user-based recurrent neural network recommendations, in: Proceedings of the eleventh ACM conference on recommender systems, 2017, pp. 152–160.
[14] M. Ludewig, N. Mauro, S. Latifi, D. Jannach, Empirical analysis of session-based recommendation algorithms: A comparison of neural and non-neural approaches, User Modeling and User-Adapted Interaction 31 (2021) 149–181.
[15] M. F. Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019.
[16] J. Zhu, K. Mao, Q. Dai, L. Su, R. Ma, J. Liu, G. Cai, Z. Dou, X. Xiao, R. Zhang, Bars: Towards open benchmarking for recommender systems, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.
[17] Z. Sun, H. Fang, J. Yang, X. Qu, H. Liu, D. Yu, Y. Ong, J. Zhang, Daisyrec 2.0: Benchmarking recommendation for rigorous evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2022) 8206–8226.
[18] J. A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience, User modeling and user-adapted interaction 22 (2012) 101–123.
[19] Z. Munawar, N. Suryana, Z. B. Sa'aya, Y. Herdiana, Framework with an approach to the user as an evaluation for the recommender systems, in: 2020 Fifth International Conference on Informatics and Computing (ICIC), IEEE, 2020, pp. 1–5.
[20] B. P. Knijnenburg, M. C. Willemsen, Evaluating recommender systems with user experiments, in: Recommender systems handbook, Springer, 2015, pp. 309–352.
[21] J. Kunkel, T. Donkers, L. Michael, C.-M. Barbu, J. Ziegler, Let me explain: Impact of personal and impersonal explanations on trust in recommender systems, in: Proceedings of the 2019 CHI conference on human factors in computing systems, 2019, pp. 1–12.
[22] I. Kangas, M. Schwoerer, L. J. Bernardi, Recommender systems for personalized user experience: lessons learned at booking. com, in: Proceedings of the 15th ACM Conference on Recommender Systems, 2021, pp. 583–586.
[23] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, J. Basilico, Deep learning for recommender systems: A netflix case study, AI Magazine 42 (2021) 7–18.
[24] M. Bartlett, F. Morreale, G. Prabhakar, Analysing privacy policies and terms of use to understand algorithmic recommendations: the case studies of tinder and spotify, Journal of the Royal Society of New Zealand 53 (2023) 119–132.
[25] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings

of the 26th international conference on world wide web, 2017, pp. 173–182.
[26] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.
[27] L. Wu, S. Li, C.-J. Hsieh, J. Sharpnack, Sse-pt: Sequential recommendation via personalized transformer, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 328–337.
[28] S. C. Han, T. Lim, S. Long, B. Burgstaller, J. Poon, Glocal-k: Global and local kernels for recommender systems, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3063–3067.
[29] M. Kaminskas, D. Bridge, Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems, ACM Transactions on Interactive Intelligent Systems (TiiS) 7 (2016) 1–42.
[30] J. Vig, S. Sen, J. Riedl, The tag genome: Encoding community knowledge to support novel interaction, ACM Transactions on Interactive Intelligent Systems (TiiS) 2 (2012) 1–44.
[31] D. Freedman, R. Pisani, R. Purves, Statistics (international student edition), Pisani, R. Purves, 4th edn. WW Norton & Company, New York (2007).
[32] P. Pu, L. Chen, R. Hu, A user-centric evaluation framework for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 157–164.
[33] A. Said, B. Fields, B. J. Jain, S. Albayrak, User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm, in: Proceedings of the 2013 conference on Computer supported cooperative work, 2013, pp. 1399–1408.
[34] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, J. A. Konstan, User perception of differences in recommender algorithms, in: Proceedings of the 8th ACM Conference on Recommender systems, 2014, pp. 161–168.
[35] C. Lleras, Path analysis, Encyclopedia of social measurement 3 (2005) 25–30.
[36] B. Muthén, L. Muthén, Mplus, in: Handbook of item response theory, Chapman and Hall/CRC, 2017, pp. 507–518.
[37] V. Braun, V. Clarke, Thematic analysis., American Psychological Association, 2012.
[38] D. Kotkov, J. A. Konstan, Q. Zhao, J. Veijalainen, Investigating serendipity in recommender systems based on real user feedback, in: Proceedings of the 33rd annual acm symposium on applied computing, 2018, pp. 1341–1350.
[39] Í. Silva, L. Marinho, A. Said, M. C. Willemsen, Leveraging chatgpt for automated human-centered explanations in recommender systems, in: Proceedings of the 29th International Conference on Intelligent User Interfaces, 2024, pp. 597–608.
[40] A. Acharya, B. Singh, N. Onoe, Llm based generation of item-description for recommendation system, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1204–1207.
[41] R. Sun, X. Li, A. Akella, J. A. Konstan, Large language models as conversational movie recommenders: A user study, arXiv preprint arXiv:2404.19093 (2024).
[42] R. Sun, A. Akella, R. Kong, M. Zhou, J. A. Konstan, Interactive content diversity and user exploration in online movie recommenders: A field experiment, International Journal of Human–Computer Interaction (2023) 1–15.

A. Appendices

Deep Learning Model Repositories:

- ***NCF***: https://github.com/yihong-chen/neural-collaborative-filtering
- ***SSE-PT***: https://github.com/lizli502/SSE-PT
- ***BERT4Rec***: https://github.com/jaywonchung/BERT4Rec-VAE-Pytorch

- ***GLocal-K***: https://github.com/fleanend/TorchGlocalK

For all model reproduction, we mostly use the default optimized parameter claimed by the repository on the ml-1m dataset, detailed below. Specifically, we ran 301 epochs for each model and select the top 12 recommendations from the best performed eopch (based on its NDCG value) for each user.

	gmf_config	**mlp_config**	**neumf_config**
num_epoch	301	301	301
batch_size	1024	1024	1024
optimizer	adam	adam	adam
adam_lr	1e-3	1e-3	1e-3
latent_dim	8	8	8
num_negative	4	4	4
l2_regularization	0.01	0.0000001	0.01
layers	-	[16,64,32,16,8]	[16,64,32,16,8]
pretrain	-	True	True

Table A1
NCF training parameters

GLocal-K Param	**GLocal-K Value**	**SSE-PT Param**	**SSE-PT Value**
n_hid	500	*num_epoch*	301
n_dim	5	*batch_size*	128
n_layers	2	*max_len*	50
gk_size	3	*user_hidden_units*	50
max_epoch_p	500	*item_hidden_units*	50
max_epoch_f	500	*lr*	0.001
patience_p	5	*num_blocks*	2
patience_f	10	*num_heads*	1
tol_p	1e-4	*dropout_rate*	0.5
tol_f	1e-5	*threshold_user*	1.0
lambda_2	20	*threshold_item*	1.0
lambda_s	0.006	l2_emb	0.0
dot_scale	1	k	12

Table A2
GLocal-K and SSE-PT training parameters

	Dataset	Dataloader	NegativeSampler	Trainer	Model
min_rating	4				
min_uc	5				
min_sc	0				
split	leave_one_out				
eval_set_size	500				
train_batch_size		64			
test_batch_size		64			
train_negative_sampler_code			random		
train_negative_sampler_size			100		
test_negative_sampler_code			100		
test_negative_sampler_size			100		
trainer_code				bert	
optimizer				adam	
lr				0.001	
weight_decay				0	
decay_step				15	
gamma				0.1	
num_epochs				301	
metric_ks				[10,20,50]	
best_metric				NDCG@10	
All BERT-relevant args					None
num_item for DAE or VAE					None
num_hidden for DAE or VAE					0
hidden_dim for DAE or VAE					600
latent_dim for DAE or VAE					200
dropout for DAE or VAE					0.5

Table A3
BERT4Rec training parameters