

Algorithm Selection for Recommender Systems via Meta-Learning on Algorithm Characteristics

Jarne Mathi Decker

University of Siegen

Siegen, Germany

jarne.decker@student.uni-siegen.de

Joeran Beel

University of Siegen & Recommender-Systems.com

Siegen, Germany

joeran.beel@uni-siegen.de

ABSTRACT

The Algorithm Selection Problem for recommender systems—choosing the best algorithm for a given user or context—remains a significant challenge. Traditional meta-learning approaches often treat algorithms as categorical choices, ignoring their intrinsic properties. Recent work has shown that explicitly characterizing algorithms with features can improve model performance in other domains. Building on this, we propose a per-user meta-learning approach for recommender system selection that leverages both user meta-features and automatically extracted algorithm features from source code. Our preliminary results, averaged over six diverse datasets, show that augmenting a meta-learner with algorithm features improves its average NDCG@10 performance by 8.83% from 0.135 (user features only) to 0.147. This enhanced model outperforms the Single Best Algorithm baseline (0.131) and successfully closes 10.5% of the performance gap to a theoretical oracle selector. These findings show that even static source code metrics provide a valuable predictive signal, presenting a promising direction for building more robust and intelligent recommender systems.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Algorithm Selection, Meta-Learning, Recommender Systems, Feature Engineering

ACM Reference Format:

Jarne Mathi Decker and Joeran Beel. 2025. Algorithm Selection for Recommender Systems via Meta-Learning on Algorithm Characteristics. In *Proceedings of 19th ACM Conference on Recommender Systems (Submitted to RecSys 2025)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

It is a well-established principle in recommender systems and machine learning that no single algorithm performs optimally across all problem types, a manifestation of the "No

Free Lunch" theorem [44]. In the domain of recommender systems, this is particularly evident: evaluations regularly show that the performance of recommendation algorithms are inconsistent across different applications or datasets [8]. This performance variance extends down to the individual user level, where the optimal algorithm can differ significantly between users within the same context [15, 22].

1.1 Background: Algorithm Selection

The high context dependence of algorithm performance leads to the Algorithm Selection Problem (ASP) [30]. The standard practice of selecting the Single Best Algorithm (SBA) for a diverse user base is inherently suboptimal. It leads to a quantifiable performance loss for a large fraction of users for whom the chosen algorithm is not the best fit. At least in theory, it would be ideal to identify and then use the individually best algorithm for each user, item, or scenario. To do this perfectly well, an "Oracle" would be needed [31]. The performance gap between the Single Best Algorithm (SBA) and such a theoretical Oracle selector - which could choose the best algorithm for each user - is substantial [16]. Consequently, a common approach across various fields is to employ a meta-learning paradigm to approximate this Oracle, thereby automating algorithm selection for contexts like recommender systems [15, 19, 22, 24, 27, 43]. Here, for each instance in a dataset a machine learning model (meta-learner) is trained to predict which algorithm will perform best on that individual instance.

1.2 Research Problem

In the past years, there has been lots of research on algorithm selection for recommender systems [19, 22], also, but not exclusively, from our own research group [1–7, 9–13, 15, 17, 18, 20, 23, 33–43]. However, current works on meta-learning, especially in the domain of recommender systems, treat algorithms as featureless classes. This means, based on e.g. user, item or dataset characteristics, the meta-learner tries to learn which algorithm is best, without knowing anything about the algorithm itself.

Recognizing the limitations of treating algorithms as feature-less black boxes, recent research has begun to explore the use of algorithm features (f_A) to explicitly characterize the algorithms themselves. The primary motivation is to improve the prediction quality and to train a single, unified meta-model that can generalize to new, unseen algorithms by understanding their properties [14, 29]. To the best of our knowledge, there are few researchers who have explored this (Table 1). These feature types range from static, implementation-level properties like source code metrics [29] to more abstract, semantic representations derived

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Submitted to RecSys 2025, September 22–26, 2025, Prague, Czech Republic

© 2025 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

from Large Language Models [45] or Knowledge Graphs (KG) [26]. Other approaches characterize algorithms by their configurable parameters [32] or their empirical performance footprints on a set of probe datasets [21].

While this prior work establishes the value of f_A , its application context is critical. Research on algorithm features has largely focused on domains such as SAT solving [29, 45], continuous optimization [21, 25, 26] or dataset-level selection of ML classifiers [32].

1.3 Research Goal

The application of this feature-based meta-learning approach to the per-user algorithm selection task in recommender systems remains an open and promising field of research. Standard meta-learning approaches for recommender systems still largely treat diverse algorithms like k-NN and Matrix Factorization as equivalent "choices," despite their fundamentally different operational principles [22]. To the best of our knowledge, a systematic study combining automatically extracted algorithm features with user features for this specific task has not been conducted. Our work aims to fill this gap by providing a first empirical analysis using static source code metrics as a practical and fully automated source of algorithm features.

Our key hypothesis is that by making the meta-learner aware of the fundamental differences between algorithms—via explicit algorithm features (f_A)—its selection capabilities can be improved beyond what is possible using only user characteristics (f_U). For example, a meta-learner could learn that users with sparse interaction histories (f_U) benefit from algorithms with low code complexity (f_A), a connection that is impossible to learn without an explicit algorithm representation. Our goal, hence, is to explore if and to what extent the use of algorithm features can improve meta-learning for recommender systems algorithm selection.

2 METHODOLOGY

2.1 Ground Truth Generation

Datasets: We conducted all experiments on Six datasets from different domains, and with different data characteristics: MovieLens-1M, LastFM-360K, Book-Crossing, Retail-Rocket, Steam and a dataset containing restaurant-ratings. After applying initial preprocessing, such as filtering users with fewer than 10 interactions, the final statistics of these datasets are summarized in Table 2.

Algorithms portfolio: Our portfolio comprises nine implementations from LensKit and RecBole, selected to cover a range of distinct algorithmic paradigms. We include classic baselines (Popularity), neighborhood-based methods (ItemKNN), matrix factorization (BPR, ImplicitMF), an autoencoder-based model (EASE), and a sequential model (FPMC). To assess potential implementation-specific effects, our portfolio includes versions of Popularity, ItemKNN, and BPR from both libraries.

HPO: For the sake of time and energy efficiency, we did not optimize hyperparameters for the single algorithms. For our scenario, we consider this acceptable as our goal is not to develop one single best algorithm but to predict which algorithm out of a pool of algorithms performs best. This is regardless of hyperparameters.

Table 1: Overview of Algorithm Feature (f_A) Generation Approaches from Literature.

Feature Type	Key Paper(s)	Brief Description
Source Code & AST	Pulatov et al. [29]	Static analysis (LOC, CC) and AST graph properties.
LLM Embeddings	Wu et al. [45]	High-dimensional embeddings from code/text via LLMs.
Explainability	Kostovska et al. [26], Nikolikj et al. [28]	Vector of SHAP scores of instance features.
KG Embeddings	Kostovska et al. [26]	Learned vectors of algorithm nodes from a KG.
Hyperparameters	Tornede et al. [32]	Vector of hyperparameter values defining a configuration.
Performance	Eftimov et al. [21]	Vector of performance scores on benchmark problems.

Performance and Ground Truth: We applied a temporal evaluation protocol. For each user in every dataset, we sorted their interaction history chronologically and split it into an 80%/20% training/test split. Each algorithm from the portfolio was trained on the aggregate training data of all users. We measured performance by nDCG@10.

This process yielded our ground truth: a performance matrix $P_{u,a}$ where each entry represents the NDCG@10 score for user u with algorithm a .

Table 2: Statistics of Preprocessed Datasets.

Dataset	Users	Items	Interactions	Sparsity
MovieLens	6,040	3,706	1,000,209	95.53%
LastFM	1,874	17,612	92,779	99.72%
BookCrossing	2,946	17,384	272,677	99.47%
RetailRocket	9,446	68,433	240,843	99.96%
Steam	2,189	5,076	104,737	99.06%
Restaurants	59	84	681	86.26%

2.2 Meta-Feature Engineering

For meta-learning, we engineered features describing the users (f_U) and algorithms (f_A).

User Features (f_U): We represent each user as a vector of 15 meta-features derived from their training data. These features capture multiple dimensions including activity (e.g. number of interactions), rating patterns (e.g. average rating and rating entropy), temporal dynamics (e.g. history duration), and item popularity preferences (e.g. average popularity of all items the user interacted with). For our experiments, these features cover the standard approach for Algorithm Selection, where only instances, in this case users, are explicitly characterized through features.

Algorithm Features (f_A): As an initial, automatically extractable representation, we characterized each algorithm implementation using static source code analysis via the Radon tool¹. This provided a quantitative "fingerprint" of each implementation based on metrics for size (e.g. SLOC), complexity (e.g. Average Cyclomatic Complexity), and Halstead metrics (e.g., Effort).

In addition, we constructed the Abstract Syntax Tree for each algorithm implementation and calculated AST-features like node count, average degree and depth using the networkx library. A full list of the features f_A we utilized is provided in Table 3.

Table 3: Overview of Algorithm Meta-Features (f_A).

Feature Name	Category
<i>Static & Complexity Metrics</i>	
sloc	Size
lloc	Size
average_cc_file	Complexity
num_complexity_blocks	Complexity
<i>Halstead Metrics</i>	
hal_volume	Code Volume
hal_difficulty	Code Difficulty
hal_effort	Implementation Effort
<i>AST Graph Metrics</i>	
ast_node_count	Graph Structure
ast_edge_count	Graph Structure
ast_avg_degree	Graph Structure
ast_max_degree	Graph Structure
ast_transitivity	Graph Structure
ast_avg_clustering	Graph Structure
ast_depth	Graph Structure

2.3 Meta-Learning and Evaluation

We designed two meta-learning models and evaluated them against standard baselines to quantify their effectiveness.

Baselines: We established two performance baselines. The Single Best Algorithm (SBA) represents the performance of the single best-performing algorithm when applied uniformly to all users in a dataset. The Virtual Best Algorithm (VBA), or Oracle, represents the theoretical maximum performance achievable by a perfect per-user algorithm selector that chooses the best algorithm for each user.

Meta-Learners: We compare two models:

- (1) $M(\text{User-Only})$: A baseline meta-learner trained only on user features (f_U) to predict a performance vector for all candidate algorithms. This would represent the state of the art in meta-learning for recommender systems.
- (2) $M(\text{User+Algo})$: Our proposed model trained on a re-structured dataset of (user, algorithm) pairs. It uses a concatenated features vector of both users and algorithm features (f_U, f_A) to predict a single performance score.

¹Radon: A Python tool for computing code metrics. Available at: <https://radon.readthedocs.io/>

For this initial work, we used a hyperparameter-tuned LightGBM regressor as the underlying model for both approaches to ensure a fair comparison.

Evaluation Protocol To obtain robust performance estimates, we employ a 5-fold cross-validation scheme. The data is split on user_id to ensure that all interactions from a single user remain in the same fold, preventing data leakage. For each fold, we train our meta-learners on the training users and evaluate them on the held-out test users. The final reported scores are the average across all 5 folds. We report three metrics: (1) **Avg. NDCG@10**, the average actual quality of the selected algorithms; (2) **Top-1 Accuracy**, the percentage of times the single best algorithm is correctly identified; and (3) **Top-3 Accuracy**, the percentage of times the actual best algorithm is among the top three predicted choices.

3 RESULTS

3.1 Baselines

When choosing the single best algorithms for each dataset, an average nDCG@10 of 0.131 is achieved (Table 4). The Oracle - the theoretical algorithm selector that chooses the best algorithm for each user - could theoretically achieve an NDCG of 0.282 (+116%). The theoretical performance of the Oracle demonstrates the potential of algorithm selection.

3.2 Meta-Learner Performance

Our baseline meta-learner $M(\text{User-Only})$ achieved an NDCG of 0.135 on average over all datasets. This is a slight improvement over using the single best algorithms. Also, the meta learner achieved an accuracy of 20.24% in predicting the best performing algorithm. Accuracy for predicting one of the top 3 algorithms was 59.07%, respectively. This indicates that user features alone contain a predictive signal, though precision in identifying the single best algorithm is limited.

Our novel $M(\text{User+Algo})$ model, augmented with algorithm features, achieved an average nDCG of 0.147. This model outperforms the SBA on four of the six datasets, with an average performance gain of 12.07%. Its average Top-1 Accuracy was 21.63%, and its Top-3 Accuracy was 62.74%. In Figure 1 we provide a visual summary of these results, comparing the average performance of both meta-learners against the SBA and VBA baselines across all datasets.

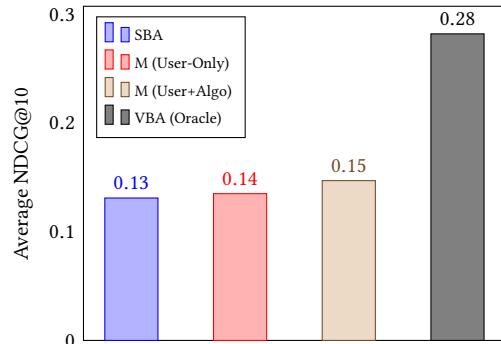
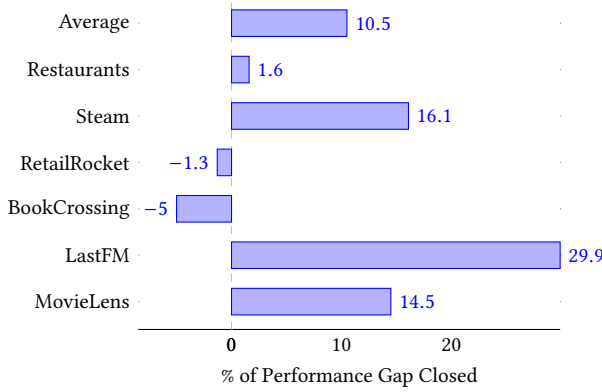


Figure 1: Average performance (NDCG@10) across all six datasets.

Table 4: Performance Summary: Baselines vs. Meta-Learners across all Datasets (Metric: Avg. NDCG@10)

Dataset	SB Algorithm	Baselines		M (User Features Only)			M (User + Algorithm Features)					
		SBA Perf.	VBA Perf.	Perf.	Acc. @1	Acc. @3	Perf.	% Gain / SBA	% Gain / User ML	Acc. @1	Acc. @3	% Gap Closed
MovieLens	LK_BPR	0.284	0.616	0.331	23.39%	48.43%	0.332	+16.99%	+0.27%	19.69%	48.79%	14.54%
LastFM	LK_BPR	0.038	0.086	0.049	28.28%	88.42%	0.052	+38.56%	+5.89%	23.33%	93.92%	29.90%
BookCrossing	RB_ItemKNN	0.041	0.072	0.037	11.98%	77.12%	0.040	-3.89%	+6.18%	11.03%	92.26%	-4.97%
RetailRocket	LK_ImplicitMF	0.107	0.181	0.107	13.84%	43.18%	0.106	-0.84%	-1.30%	12.72%	40.06%	-1.26%
Steam	LK_BPR	0.163	0.358	0.200	34.76%	65.14%	0.195	+19.16%	-2.50%	42.53%	70.81%	16.10%
Restaurants	LK_ImplicitMF	0.150	0.380	0.083	13.33%	32.12%	0.154	+2.46%	+86.56%	20.45%	30.61%	1.64%
Average	N/A	0.131	0.282	0.135	20.24%	59.07%	0.147	+12.07%	+8.83%	21.63%	62.74%	10.49%

Note: SBA = Single Best Algorithm; VBA = Virtual Best Algorithm (Oracle); M = Meta-Learner. 'Perf.' columns show average NDCG@10. 'Acc. @1' / 'Acc. @3' refer to Top-1 and Top-3 selection accuracy. '% Gap Closed' = '(ML Perf. - SBA Perf.) / (VBA Perf. - SBA Perf.)'.

**Figure 2: Percentage of the performance gap between the SBA and the VBA (Oracle) that was closed by the M (User+Algo) meta-learner.**

3.3 The Impact of Algorithm Features

Our primary hypothesis is that explicitly modeling algorithm characteristics improves selection performance. The results in Table 4 allow for a direct comparison between the two meta-learners.

Comparing the two meta-learners reveals that the effect of algorithm features is highly dependent on dataset characteristics. We observed a positive performance gain on four of the six datasets. This effect was most pronounced on the Restaurants dataset (+86.56%), where the baseline *M (User-Only)* model performed particularly poorly, suggesting that the algorithm features provided a crucial signal that was absent in the user features for that specific data structure. On larger datasets, the performance gains were more moderate and even slightly negative on the Steam and RetailRocket datasets.

Relative to the baselines, the *M (User+Algo)* model closed an average of 10.49% of the total performance gap between the SBA and the VBA (% Gap Closed). The effectiveness varies significantly by dataset, with the strongest results on LastFM (29.90% gap closed) and Steam (16.10%). In Figure 2

we visualize this variance, highlighting the datasets where the meta-learner was most and least effective at closing the performance gap relative to the SBS. The negative gap closure on some datasets indicates that for certain data characteristics, the current set of source code and AST-features is not yet sufficient to consistently outperform the strong SBA, highlighting a clear direction for future feature engineering.

4 CONCLUSION AND FUTURE WORK

In this work, we presented preliminary results from a meta-learning framework for per-user recommender system selection. Our empirical evaluation across six diverse datasets demonstrates that a meta-learner, when augmented with a combination of user and algorithm features, can consistently outperform a strong Single Best Algorithm baseline. We showed that adding algorithm features provides a tangible, though moderate, improvement in both average recommendation quality (NDCG@10) and selection accuracy over a meta-learner that relies on user features alone.

These findings provide a strong foundation for future research. Our immediate next steps are to:

- **Expand the Algorithm Feature Set:** To address cases where our current model underperformed, we will incorporate more diverse features, particularly performance-based landmarks [21] and potentially manually engineered conceptual features (e.g., algorithm family, learning paradigm) to provide a richer signal to the meta-learner.
- **Diversify Models and Datasets:** We plan to expand our algorithm portfolio with more diverse paradigms and continue to add datasets to further test the generalizability of our approach.
- **Explore Advanced Meta-Learners:** We will investigate more advanced model architectures, such as Factorization Machines or Two-Tower Neural Networks, which are explicitly designed to model the interactions between user and algorithm feature sets.

This work successfully demonstrates the promise of a feature-based approach for algorithm selection in recommender systems. The next phase of our research will build directly

on this foundation, shifting from static code analysis to behavioral landmarking features and from standard regressors to interaction-aware architectures, in pursuit of a meta-learning model truly aware of algorithm characteristics and behavior.

REFERENCES

- [1] Rohan Anand and Joeran Beel. 2020. Auto-Surprise: An Automated Recommender-System (AutoRecSys) Library with Tree of Parzen Estimator (TPE) Optimization. In *14th ACM Conference on Recommender Systems (RecSys)*. 1–4. <https://arxiv.org/abs/2008.13532>
- [2] Ardalan Arabzadeh, Tobias Vente, and Joeran Beel. 2024. Green Recommender Systems: Optimizing Dataset Size for Energy-Efficient Algorithm Performance. In *International Workshop on Recommender Systems for Sustainability and Social Good (RecSoGood) at the 18th ACM Conference on Recommender Systems (ACM RecSys)*. <https://isg.beel.org/pubs/2024-Green-RecSys-Dataset-Sampling-Ardalan.pdf>
- [3] Mukesh Arambakam and Joeran Beel. 2020. Federated Meta-Learning: Democratizing Algorithm Selection Across Disciplines and Software Libraries. In *7th ICML Workshop on Automated Machine Learning*. 1–8. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_39.pdf
- [4] Joeran Beel. 2015. Towards Effective Research-Paper Recommender Systems and User Modeling based on Mind Maps. *PhD Thesis. Otto-von-Guericke Universität Magdeburg* (2015).
- [5] Joeran Beel. 2017. A Macro/Micro Recommender System for Recommendation Algorithms [Proposal]. *ResearchGate* (2017). <https://doi.org/10.13140/RG.2.2.14978.79047>
- [6] Joeran Beel. 2019. Federated Meta-Learning: Democratizing Algorithm Selection Across Disciplines and Software Libraries (Proposal). *ResearchGate* (2019). <https://doi.org/10.13140/RG.2.2.25744.35844>
- [7] Joeran Beel, Akiko Aizawa, Corinna Breiteringer, and Bela Gipp. 2017. Mr. DLib: Recommendations-as-a-service (RaaS) for Academia. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries* (Toronto, Ontario, Canada) (*JCDL '17*). IEEE Press, Piscataway, NJ, USA, 313–314. <http://dl.acm.org/citation.cfm?id=3200334.3200389>
- [8] Joeran Beel, Corinna Breiteringer, Stefan Langer, Andreas Lommatzsch, and Bela Gipp. 2016. Towards reproducibility in recommender-systems research. *User modeling and user-adapted interaction* 26 (2016), 69–101.
- [9] Joeran Beel, Andrew Collins, and Akiko Aizawa. 2018. The Architecture of Mr. DLib’s Scientific Recommender-System API. In *Proceedings of the 26th Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, Vol. 2259. CEUR-WS, 78–89.
- [10] Joeran Beel, Alan Griffin, and Conor O’Shey. 2019. Darwin & Goliath: Recommendations-As-a-Service with Automated Algorithm-Selection and White-Labels. In *13th ACM Conference on Recommender Systems (RecSys)*.
- [11] Joeran Beel and Lars Kotthoff. 2019. Preface: The 1st Interdisciplinary Workshop on Algorithm Selection and Meta-Learning in Information Retrieval (AMIR). In *Proceedings of The 1st Interdisciplinary Workshop on Algorithm Selection and Meta-Learning in Information Retrieval (AMIR)*. 1–9.
- [12] Joeran Beel and Lars Kotthoff. 2019. Proposal for the 1st Interdisciplinary Workshop on Algorithm Selection and Meta-Learning in Information Retrieval (AMIR). In *Proceedings of the 41st European Conference on Information Retrieval (ECIR) (Lecture Notes in Computer Science book series (LNCS), Vol. 11438)*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra (Eds.). Springer, 383–388. https://doi.org/10.1007/978-3-030-15719-7_53
- [13] Joeran Beel, Bryan Tyrrell, Edward Bergman, Andrew Collins, and Shahad Nagoor. 2020. Siamese Meta-Learning and Algorithm Selection with ‘Algorithm-Performance Personas’ [Proposal]. *arXiv:2006.12328 [cs.LG]* (2020).
- [14] Gjorgjina Cenikj, Ana Nikolikj, Gašper Petelin, Niki van Stein, Carola Doerr, and Tome Eftimov. 2024. A Survey of Meta-features Used for Automated Selection of Algorithms for Black-box Single-objective Continuous Optimization. *arXiv:2406.06629 [cs.LG]* <https://arxiv.org/abs/2406.06629>
- [15] Andrew Collins and Joeran Beel. 2019. Meta-Learned Per-Instance Algorithm Selection in Scholarly Recommender Systems. *arXiv:1912.08694 [cs.LR]* <https://arxiv.org/abs/1912.08694>
- [16] Andrew Collins, Laura Tierney, and Joeran Beel. 2020. Per-Instance Algorithm Selection for Recommender Systems via Instance Clustering. <https://doi.org/10.48550/arXiv.2012.15151>
- [17] Andrew Collins, Dominika Tkaczyk, and Joeran Beel. 2018. A Novel Approach to Recommendation Algorithm Selection using Meta-Learning. In *Proceedings of the 26th Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, Vol. 2259. CEUR-WS, 210–219.
- [18] Andrew Collins, Dominika Tkaczyk, and Joeran Beel. 2018. One-at-a-time: A Meta-Learning Recommender-System for Recommendation-Algorithm Selection on Micro Level. In *arXiv:1805.12118*.
- [19] Tiago Cunha, Carlos Soares, and André CPL de Carvalho. 2018. Meta-learning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering. *Information Sciences* 423 (2018), 128–144.
- [20] Gordian Edenhofer, Andrew Collins, Akiko Aizawa, and Joeran Beel. 2019. Augmenting the DonorsChoose.org Corpus for Meta-Learning. In *Proceedings of The 1st Interdisciplinary Workshop on Algorithm Selection and Meta-Learning in Information Retrieval (AMIR)*. CEUR-WS, 32–38.
- [21] Tome Eftimov, Gorjan Popovski, Dragi Kocev, and Peter Korošec. 2020. Performance2vec: a step further in explainable stochastic optimization algorithm performance. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* (Cancún, Mexico) (*GECCO ’20*). Association for Computing Machinery, New York, NY, USA, 193–194. <https://doi.org/10.1145/3377929.3390020>
- [22] Michael Ekstrand and John Riedl. 2012. When recommenders fail: predicting recommender failure for algorithm selection and combination. In *Proceedings of the Sixth ACM Conference on Recommender Systems* (Dublin, Ireland) (*RecSys ’12*). Association for Computing Machinery, New York, NY, USA, 233–236. <https://doi.org/10.1145/2365952.2366002>
- [23] Srijan Gupta and Joeran Beel. 2020. Auto-CaseRec: Automatically Selecting and Optimizing Recommendation-Systems Algorithms. *OSF Preprints DOI:10.31219/osf.io/4znmd*, (2020). <https://doi.org/10.31219/osf.io/4znmd>
- [24] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2018. Automated Algorithm Selection: Survey and Perspectives. *arXiv:1811.11597 [cs.LG]* <https://arxiv.org/abs/1811.11597>
- [25] Ana Kostovska, Diederick Vermetten, Sašo Džeroski, Carola Doerr, Peter Korošec, and Tome Eftimov. 2022. The importance of landscape features for performance prediction of modular CMA-ES variants. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) (*GECCO ’22*). Association for Computing Machinery, New York, NY, USA, 648–656. <https://doi.org/10.1145/3512290.3528832>
- [26] Ana Kostovska, Diederick Vermetten, Sašo Džeroski, Panče Panov, Tome Eftimov, and Carola Doerr. 2023. Using knowledge graphs for performance prediction of modular optimization algorithms. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 253–268.
- [27] Lars Kotthoff. 2016. Algorithm selection for combinatorial search problems: A survey. In *Data mining and constraint programming: Foundations of a cross-disciplinary approach*. Springer, 149–190.
- [28] Ana Nikolikj, Ryan Lang, Peter Korošec, and Tome Eftimov. 2022. Explaining Differential Evolution Performance Through Problem Landscape Characteristics. In *Bioinspired Optimization Methods and Their Applications*, Marjan Mernik, Tome Eftimov, and Matej Črepinšek (Eds.). Springer International Publishing, Cham, 99–113.
- [29] Damir Pulatov, Marie Anastacio, Lars Kotthoff, and Holger Hoos. 2022. Opening the Black Box: Automated Software Analysis for Algorithm Selection. In *Proceedings of the First International Conference on Automated Machine Learning (Proceedings of Machine Learning Research, Vol. 188)*, Isabelle Guyon, Marius Lindauer, Mihaela van der Schaar, Frank Hutter, and Roman Garnett (Eds.). PMLR, 6/1–18. <https://proceedings.mlr.press/v188/pulatov22a.html>
- [30] John R. Rice. 1976. The Algorithm Selection Problem**This work was partially supported by the National Science Foundation through Grant GP-32940X. This chapter was presented as the George E. Forsythe Memorial Lecture at the Computer Science Conference, February 19, 1975, Washington, D. C. *Advances in Computers*, Vol. 15. Elsevier, 65–118. [https://doi.org/10.1016/S0065-2458\(08\)60520-3](https://doi.org/10.1016/S0065-2458(08)60520-3)
- [31] Alexander Tornede, Lukas Gehring, Tanja Tornede, Marcel Wever, and Eyke Hüllermeier. 2023. Algorithm selection on a meta level. *Machine Learning* 112, 4 (2023), 1253–1286.
- [32] Alexander Tornede, Marcel Wever, and Eyke Hüllermeier. 2020. *Extreme Algorithm Selection with Dyadic Feature Representation*. Springer International Publishing, 309–324. https://doi.org/10.1007/978-3-030-61527-7_21
- [33] Bryan Tyrrell, Edward Bergman, Gareth Jones, and Joeran Beel. 2020. Algorithm-Performance Personas’ for Siamese Meta-Learning and Automated Algorithm Selection. In *7th ICML Workshop on Automated Machine Learning*, Vol. 1. 16.
- [34] Tobias Vente. 2023. Advancing Automation of Design Decisions in Recommender System Pipelines. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1355–1360. <https://doi.org/doi/10.1145/3604915.3608886>
- [35] Tobias Vente and Joeran Beel. 2024. The Potential of AutoML for Recommender Systems. *arXiv* (2024), 18. <https://doi.org/10.48550/arXiv.2402.04453>

- [36] Tobias Vente, Michael Ekstrand, and Joeran Beel. 2023. Introducing LensKit-Auto, an Experimental Automated Recommender System (AutoRecSys) Toolkit. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1212–1216. <https://dl.acm.org/doi/10.1145/3604915.3610656>
- [37] Tobias Vente, Zainil Mehta, Lukas Wegmeth, and Joeran Beel. 2024. Greedy Ensemble Selection for Top-N Recommendations. In *RobustRecSys Workshop at the 18th ACM Conference on Recommender Systems (ACM RecSys)*.
- [38] Tobias Vente, Lennart Purucker, and Joeran Beel. 2022. The Feasibility of Greedy Ensemble Selection for Automated Recommender Systems. *COSEAL Workshop 2022* (2022). <http://dx.doi.org/10.13140/RG.2.2.16277.29921>
- [39] Lukas Wegmeth. 2023. Improving Recommender Systems Through the Automation of Design Decisions. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1332–1338. <https://dl.acm.org/doi/pdf/10.1145/3604915.3608877>
- [40] Lukas Wegmeth and Joeran Beel. 2022. CaMeLS: Cooperative Meta-Learning Service for Recommender Systems. In *Proceedings of the 2nd Perspectives on the Evaluation of Recommender Systems Workshop*. <https://ceur-ws.org/Vol-3228/paper2.pdf>
- [41] Lukas Wegmeth and Joeran Beel. 2022. Cooperative Meta-Learning Service for Recommender Systems. *COSEAL Workshop 2022* (2022). <http://dx.doi.org/10.13140/RG.2.2.10667.41768>
- [42] Lukas Wegmeth, Tobias Vente, and Joeran Beel. 2023. The Challenges of Algorithm Selection and Hyperparameter Optimization for Recommender Systems. *COSEAL Workshop 2023* (2023). <http://dx.doi.org/10.13140/RG.2.2.24089.19049>
- [43] Lukas Wegmeth, Tobias Vente, and Joeran Beel. 2024. Recommender Systems Algorithm Selection for Ranking Prediction on Implicit Feedback Datasets. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 1163–1167.
- [44] D.H. Wolpert and W.G. Macready. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 1 (1997), 67–82. <https://doi.org/10.1109/4235.585893>
- [45] Xingyu Wu, Yan Zhong, Jibin Wu, Bingbing Jiang, and Kay Chen Tan. 2023. Large language model-enhanced algorithm selection: towards comprehensive algorithm representation. *arXiv preprint arXiv:2311.13184* (2023).