

A Reproduction of Green Recommender Systems

Murat Ergün
murat.erguen@student.uni-siegen.de
University of Siegen
Siegen, Germany

Frederic Lück-Reuße
frederic2.lueck@student.uni-
siegen.de
University of Siegen
Siegen, Germany

Leon Ulrich Pieper
leon.pieper@student.uni-siegen.de
University of Siegen
Siegen, Germany

ABSTRACT

This paper presents a reproduction of the paper Green Recommender Systems. The goal of the paper was to find out if Recommender Systems could be trained with fewer data to save power while still achieving satisfying results. The authors of the original paper found that it depends on the type of algorithms used and divided those by their ability to be reduced. Our reproduction aims at verifying those results and tries to translate them to some general machine learning algorithms outside the realm of recommender systems. We tested the same idea on a set of additional algorithms and a wider range of datasets to test if those findings are not just restricted to the originally used algorithms and datasets. We found similar results for certain algorithms that tend to respond more positively to downsampling compared to others.

KEYWORDS

Recommender Systems, Environmental Impact, Sustainability

1 INTRODUCTION

1.1 Background

Current day Recommender Systems and similar AI based systems have a high demand for computational performance. This performance comes at the cost of high energy consumption. Especially these days, the impact such a high energy consumption has on the environment is criticized.

1.2 Research Problem

The research problem addressed in the original paper [1] is the high energy consumption and environmental impact of the deployment of large-scale recommender systems. While recommender systems are crucial for personalizing user experiences in domains like e-commerce and streaming platforms, they often rely on computationally expensive algorithms, especially deep learning models, that consume significant electricity and emit substantial CO₂. The authors argue that current approaches focus heavily on accuracy but neglect the ecological footprint. For example, training a single large model can emit as much CO₂ as multiple car lifetimes worth of emissions. This problem is critical in the age of climate change and increasing energy demands, and solving it means designing recommender systems that are both effective and energy-efficient, balancing recommendation quality and sustainability.

1.3 Original Work

The paper "Green Recommender Systems: Optimizing Dataset Size for Energy-Efficient Algorithm Performance" explores how downsampling training data can reduce the environmental impact of

recommender systems without significantly harming their accuracy. With recommender systems becoming increasingly data- and compute-intensive, the authors question whether using the full dataset is always necessary, especially when simpler or more efficient models may perform comparably well with far fewer data.

To evaluate this, the authors performed experiments on four datasets: MovieLens "100K", "1M", "10M", and "Amazon Toys and Games". They train ten different algorithms—ranging from basic models like popularity-based methods to more complex ones like FunkSVD and BiasedMF—on varying fractions of training data, from 10% to 100%. The key metric used for evaluation is nDCG@10, a common measure of ranking quality. The goal is to observe how much performance degrades as training data is reduced.

The results reveal that the impact of downsampling varies considerably across algorithms and dataset types. Algorithms categorized as Group 2 (e.g. FunkSVD, BiasedMF, and Popularity) showed only a modest performance drop when trained on 50% of the data, often within 13% of their full-data performance, particularly on sparse datasets like Amazon Toys and Games. In contrast, Group 1 algorithms (e.g., UserKNN, SVD, NMF) experienced much steeper declines in accuracy with smaller training sets, showing a near-linear dependency on dataset size.

The study also estimates the environmental benefits of this approach. Downsampling to 50% can reduce runtime by approximately 28%, translating to an estimated 27.4 kg of CO₂ savings per algorithm-dataset combination when considering tuning and repeated experimentation. These findings underscore that using less data can yield substantial energy savings an increasingly important consideration in the age of green computing.

Ultimately, the paper demonstrates that data volume is not always the key to better recommendations. Instead, certain algorithms can maintain strong performance with far less data, offering a compelling trade-off between energy efficiency and recommendation quality. The authors advocate for broader adoption of sustainability-focused evaluation practices and propose dataset downsampling as a practical step toward greener recommender systems.

1.4 Research Goal

The goal of our work was to see if the results of the original paper were applicable to a further set of datasets and algorithms and test how the ones used by the original paper would respond to changes in their hyper parameters. Further we also tried to extend the scope of the original paper to more general machine learning.

2 METHODOLOGY

For our reproduction of the "Green Recommender Systems" paper, we extended the original study in several directions. Most notably,

instead of reusing the original datasets, we applied the algorithms to five new datasets: Amazon Video Games, Goodreads Poetry Reviews, Beauty Products, Book Crossing, and Grocery and Gourmet Food.

In total, 17 recommender algorithms were tested, including 6 additional ones not used in the original paper. These were categorized into three groups: 1. Algorithms highly impacted by reduced dataset size 2. Algorithms only slightly impacted 3. Newly introduced algorithms

We tested the algorithms using the nDCG@10 metric and later compared the results with Recall@10. Each experiment was conducted across 10 levels of data availability to investigate how much performance degrades with less training data. We also repeated experiments with a different random seed to evaluate stability.

Other variations included testing with different pruning levels (from 10-core to 5-core) and observing the effects on recommendation quality.

The methodology used in our reproduction of the original paper followed the methodology of the original paper closely. We were able to easily modify the original source code to use additional datasets. A major problem in our reproduction was a mismatch of the used Python versions. This caused problems with the libraries used for parallel processing. These problems forced us to have the algorithms run single threaded which vastly increased the needed runtime of certain algorithms. As a consequence of this we later on started to skip the "biasedmf" and "nmf" algorithms cause a single run of one of these algorithms could last several hours which would have far exceeded the time frame set aside for the reproduction.

The methodology is to make every algorithm run with a changing amount of available data. This happened in 10 steps. The results of the algorithms are then saved to a json file to be later visualized. We then simply plot the algorithms to simplify comparing the usability of the algorithms with different amounts of data.

In addition to recommender systems, we evaluated one deep learning model (implemented with TensorFlow) and one statistical model (based on exponential smoothing). For our experiments, we used two datasets: one containing household power consumption measurements, and another with wind power generation data from a specific German energy company.

For both models, we examined the impact of ten different levels of data availability. In the deep learning model, we also incorporated a random seed (or more precisely, a random state) to ensure reproducibility.

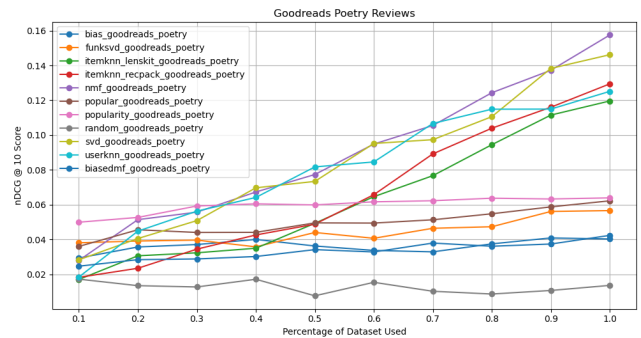
The statistical model does not rely on a random state. Instead, we varied the pattern with which the model is trained on the data. For example, if the dataset contains measurements taken every minute, and we want the model to learn daily patterns, we use 1440 (minutes per day) as a key parameter. Similarly, we adjust the prediction horizon: to forecast one day ahead, we again use 1440 as the prediction interval.

Model performance was evaluated using root mean squared error (RMSE) for the deep learning approach. For the statistical model, we assessed performance using both RMSE and mean absolute percentage error (MAPE).

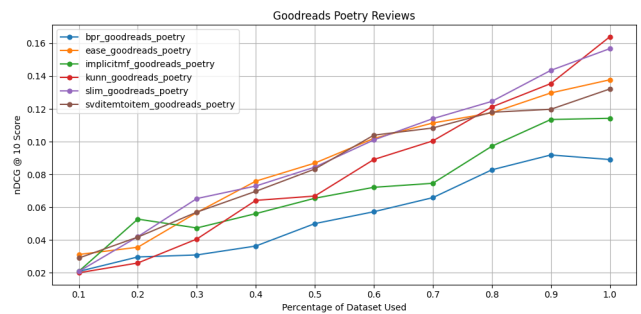
Finally, the results were visualized to facilitate a clear comparison between the models.

3 RESULTS & DISCUSSION

In this section, we present and discuss the results of our reproduction experiments, with a particular focus on analyzing the impact of dataset downsampling on algorithm performance and assessing which models are most suitable for green recommender systems. We first report the outcomes of applying different levels of data reduction across two representative datasets, using nDCG@10 as our primary evaluation metric and 10-core pruning as our default preprocessing strategy. Next, we examine whether further experiments—specifically, applying more aggressive pruning (5-core), testing random seed variability, and additionally evaluating performance using Recall@10—alter our initial assessment of model robustness and sustainability. Finally, we discuss results obtained from deep learning and statistical forecasting models to explore the extent to which the concept of downsampling generalizes beyond traditional recommendation algorithms. Throughout this section, we compare our observations to the findings of the original study and reflect on how our additional analyses confirmed, refined, or challenged our earlier hypotheses regarding the trade-off between data volume, accuracy, and energy efficiency.

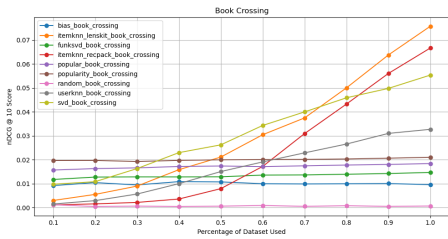


(a) Algorithms From The Original Paper [1]

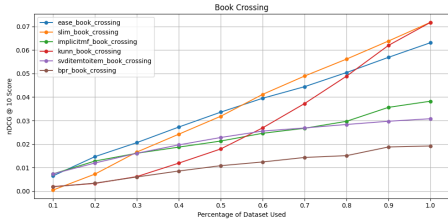


(b) New Algorithms

Figure 1: Goodreads Poetry Reviews



(a) Algorithms From The Original Paper [1]



(b) New Algorithms

Figure 2: Book Crossing

Figures 1–2 display the nDCG@10 results for all evaluated algorithms on the Goodreads Poetry Reviews and Book Crossing datasets, which we selected as representative examples in our study. The Goodreads dataset is characterized by moderately dense user-item interactions, while the Book Crossing dataset is more sparse and exhibits greater variability in ratings. Each figure illustrates how performance evolves as the amount of available data increases from 10% to 100%. In both datasets, we observe that some algorithms achieve higher accuracy as more data becomes available, while others remain relatively constant across all data fractions. A detailed discussion of the specific behaviors and sensitivities of individual algorithms is provided in later sections.

We also conducted the same experiments on three additional datasets—Amazon Video Games, Beauty Products, and Grocery and Gourmet Food—and found highly similar patterns. For clarity, these graphs are not shown here but are available in the supplementary materials.

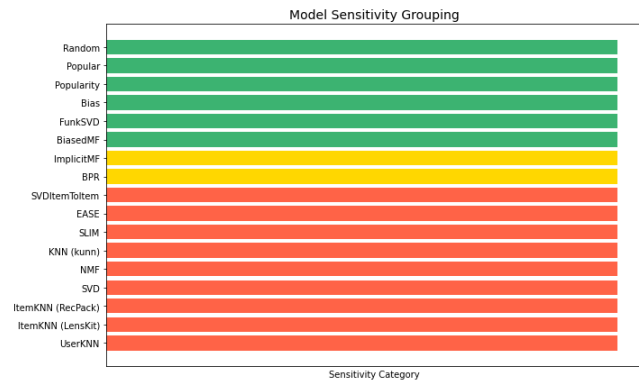


Figure 3: Sensitivity classification of all evaluated recommendation algorithms. Models are color-coded based on their responsiveness to downsampling.

Model Grouping & Sensitivity: Figure 3 provides an overall summary of algorithm sensitivity, complementing the dataset-specific performance trends discussed previously. To provide a structured understanding of how different algorithms respond to reduced data availability, we grouped all evaluated models based on their sensitivity to training data volume, using performance improvements in nDCG@10 across downsampling levels.

- High Sensitivity Models – Less suitable for green systems:** Algorithms like UserKNN, ItemKNN, SVD, NMF, SLIM, and EASE fall into the red category, showing strong dependence on large volumes of training data. These models require more computational effort, longer training times, and tend to involve dense matrix operations or fine-grained similarity calculations. Although they often offer higher accuracy, their resource consumption scales with data size, making them less energy-efficient. Therefore, they are less suitable for green deployments unless data volume is fixed and large-scale infrastructure is available.
- Moderate Sensitivity Models – Conditionally suitable:** Models like BPR and ImplicitMF show moderate performance gains as more data becomes available. They offer a balance between quality and cost, and could be considered partially suitable for green settings—especially when lightweight optimization or partial training is applied. Their suitability may depend on domain characteristics (e.g., cold-start likelihood, data sparsity).
- Low Sensitivity Models – Best fit for green recommender systems:** Algorithms such as BiasedMF, FunkSVD, Popularity, and Bias exhibit robust performance even with limited data. They maintain relatively consistent accuracy from the earliest stages of training and require minimal computation. Their efficiency and resilience under constrained data make them highly recommended for sustainable recommendation system design.

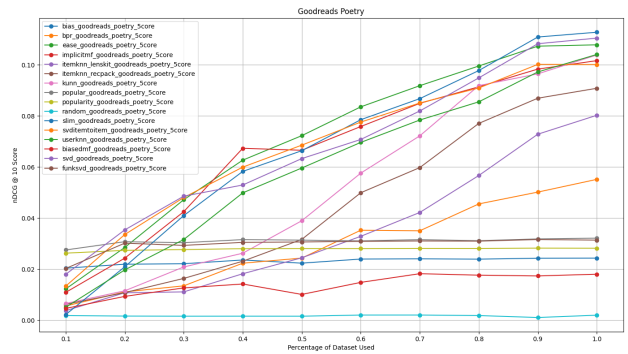


Figure 4: Algorithm performance on Goodreads Poetry Reviews with 5-core pruning

Figure 4 presents the results of our experiments on the Goodreads Poetry Reviews dataset using 5-core pruning. Compared to the 10-core preprocessing strategy, this more aggressive pruning substantially reduced the number of interactions per user and item.

As expected, the overall nDCG@10 scores decreased across nearly all algorithms, reflecting the additional sparsity introduced by the filtering.

Notably, while the general ranking of model performance remained broadly consistent—high-sensitivity algorithms such as SVD, SLIM, and EASE still benefited most from larger datasets—the performance gap between low- and high-sensitivity models widened. For example, Random, Bias, and Popularity maintained almost flat performance across all data fractions, while the gains of data-hungry algorithms were more pronounced at higher data percentages.

Overall, this experiment demonstrates that more extreme pruning conditions amplify the differences in model sensitivity observed in the standard 10-core setup. However, the relative patterns remained largely stable, suggesting that the conclusions about model suitability for green recommender systems are robust even under substantially increased sparsity. Therefore, while 5-core pruning reduces the absolute effectiveness of most algorithms, it does not fundamentally change the assessment of which models are more resilient or sustainable.

In addition to the Goodreads Poetry Reviews dataset, we also conducted 5-core pruning experiments on the Book Crossing and Beauty Products datasets to verify whether the observed trends were consistent across domains with different levels of sparsity. In all cases, the results closely mirrored those obtained from Goodreads: high-sensitivity models exhibited substantial accuracy degradation under increased pruning, while low-sensitivity models remained relatively stable. Because these findings did not reveal fundamentally different patterns compared to our primary dataset, we chose not to include the additional plots in the main body of this paper. For completeness, the corresponding figures are available in the supplementary materials.

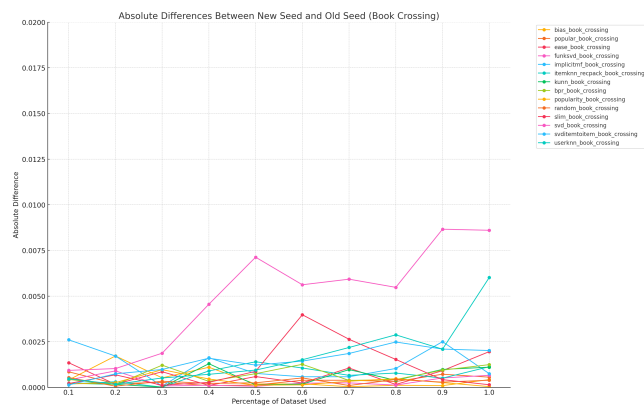


Figure 5: Absolute differences in nDCG@10 scores between experiments conducted with two different random seeds on the Book Crossing dataset. Lower values indicate higher stability across data fractions.

To evaluate the stability and reproducibility of our results, we repeated the entire experimental pipeline with a different random seed and compared the resulting nDCG@10 scores to those from the original runs. As illustrated in the Figure 5, most algorithms

exhibited very limited variation, typically remaining well below 0.002 across all data fractions. This suggests that the observed performance trends are highly consistent and largely independent of the specific random split used.

Among all models, SVD showed the largest absolute differences, reaching up to approximately 0.0086 at higher data percentages. While this fluctuation is somewhat greater than the variability seen in simpler algorithms such as Bias, Popularity, and Random, its absolute magnitude still represents less than 1–2% of the typical nDCG range. Therefore, this variability is unlikely to materially affect conclusions about model ranking or suitability.

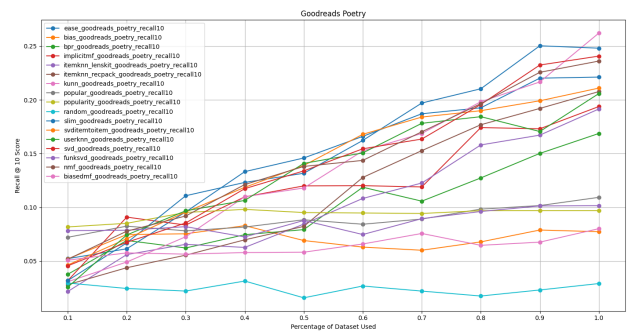


Figure 6: Recall@10 scores for all evaluated algorithms on the Goodreads Poetry Reviews dataset across different training data percentages. Patterns largely align with nDCG@10 results, with high-sensitivity models showing the steepest improvements as dataset size increases.

In general, the Recall@10 trends closely mirror those observed with nDCG@10: high-sensitivity models such as SLIM, SVD, and EASE achieve the highest recall scores, reaching final values between 0.22 and 0.25 at full dataset utilization. These algorithms display steep improvements in recall as more data becomes available, highlighting their strong dependence on large training volumes.

By contrast, simpler models such as Bias, Popularity, and Random maintain relatively stable performance across all data fractions, with Recall scores typically ranging between 0.06 and 0.10. This reinforces their characterization as low-sensitivity algorithms that deliver consistent but modest results even under substantial down-sampling.

A few nuanced differences emerge when comparing Recall to nDCG. Notably, Popularity and Bias appear relatively stronger in recall than in nDCG, suggesting that these simpler methods can recover a higher proportion of relevant items despite offering lower ranking precision. Conversely, models like FunkSVD and BiasedMF show limited recall improvements compared to their ranking performance, indicating that their strength may lie more in precise item ordering than in broad coverage.

From a sustainability perspective, these observations further confirm that algorithms with flatter recall curves are highly suitable for green recommender systems. Their stable performance

under varying data availability implies fewer reruns, lower energy consumption, and reduced computational cost. In contrast, while high-sensitivity models offer superior recall when fully trained, the significant additional resource requirements to unlock this potential may limit their practicality in energy-constrained environments.

Overall, our findings closely align with those reported in the original study by [1] Ardalan Arabzadeh. Specifically, we observed that high-sensitivity models such as SVD, NMF, SLIM, and EASE consistently achieved the highest recommendation accuracy but exhibited substantial performance degradation under downsampling, confirming their reliance on large training volumes. Conversely, simpler models like Bias, Popularity, and Random maintained stable performance across different data fractions and pruning levels, demonstrating their suitability for energy-efficient deployments. While our work extended the analysis to additional datasets, random seed tests, and alternative metrics such as Recall@10, the general performance patterns and model rankings remained remarkably consistent. These results reinforce the conclusion that downsampling can effectively reduce computational costs and environmental impact without fundamentally altering the relative performance of recommender algorithms. Our study therefore provides further evidence supporting the adoption of green recommender system practices in diverse application contexts.

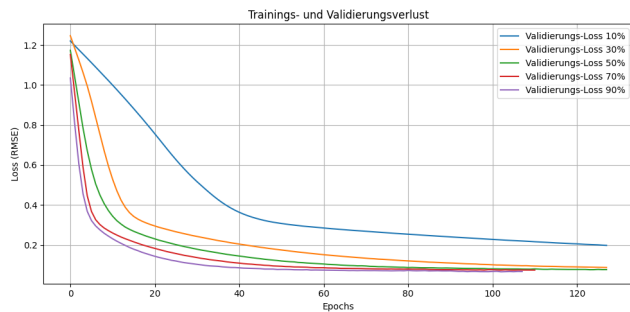


Figure 7: DeepLearning Results of Data set wind power energy generating part 1

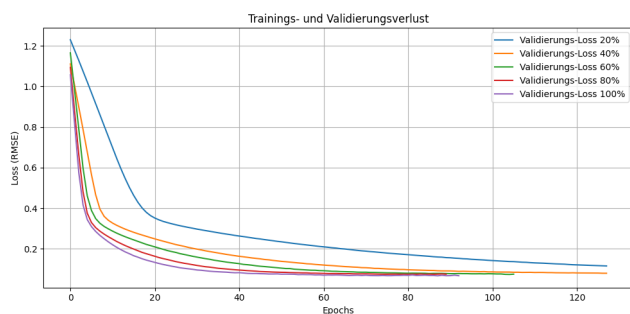


Figure 8: DeepLearning Results of Data set wind power energy generating part 2

Figures 7 and 8 illustrate the results of the deep learning algorithm. To enhance readability and allow for a more precise analysis, the results are divided into two separate graphs rather than combined into a single, more complex one.

The graphs show that downsampling has a significant impact on the quality of model training. In particular, the validation loss at 30% of the data is nearly identical to that at 100%, indicating that a smaller training set can still produce comparable performance.

Furthermore, training with 80% of the data results in a shorter training time than with the full dataset, while achieving nearly the same validation loss.

When considering both validation loss and training duration, the most balanced performance is observed in 60% downsampling. At this level, the validation loss remains close to that of the full dataset, yet the training time is noticeably reduced, making it an efficient trade-off.

However, it is important to note that these results may not represent the optimal configuration. The deep learning model used is self-developed, and the testing phase was relatively short. As such, we cannot conclusively determine whether the model was trained correctly or whether the validation loss was the most appropriate metric for evaluation.

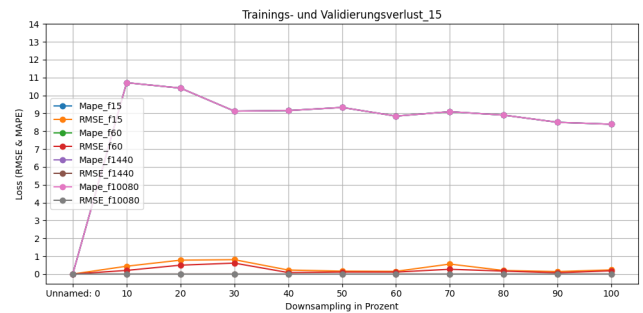


Figure 9: Exponential Smoothing Results of Data set energy consumption, training with 15min pattern

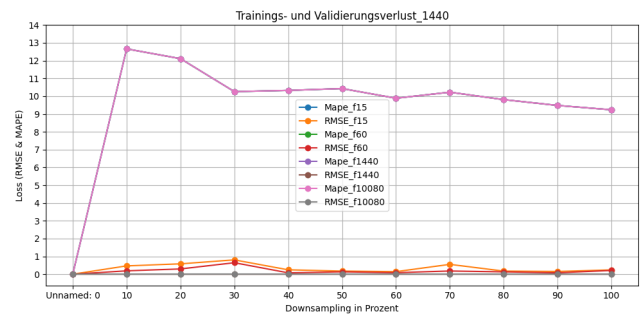


Figure 10: Exponential Smoothing Results of Data set energy consumption, training with daily pattern

Figures 9 and 10 show the results of the statistical algorithm. The graphs illustrate that downsampling offers significant advantages. Looking at the RMSE (Root Mean Squared Error), we can

see that using only 40% to 60% of the data produces results that are nearly equivalent to those obtained with 80% to 100% of the data. For MAPE (Mean Absolute Percentage Error), a data set reduced to 60% yields almost identical results as using the full data set.

In the legend of each graph, the suffix "f" followed by a number indicates the prediction horizon, that is, how many minutes into the future the model is forecasting. Interestingly, for MAPE, the forecast length has no significant impact on the model's performance. In contrast, for RMSE, the results vary depending on how far into the future is predicted.

Each figure is titled "Trainings- und Validierungsverlust", with x denoting the time pattern used for training the model. A comparison of Figures 9 and 10 reveals noticeable differences in performance in the two time patterns.

At first glance, the results might suggest that a shorter training pattern combined with a longer prediction horizon leads to better results. However, such a conclusion should be approached with caution.

The statistical model used is self-developed and lacks the extensive testing applied to the recommender systems used during the internship. Due to time constraints, comprehensive validation was not possible.

Therefore, the chosen loss function may not be optimal and the relationship between the training pattern and the prediction horizon is not sufficiently understood. More analysis and testing are required to confirm these observations and better understand their implications.

4 LIMITATIONS

The biggest limitation was our limited computing power. It meant that we could only run so many algorithms in our set time frame. This problem was further complicated by problems in the library responsible for parallel processing. Because of changes in the library we weren't able to make algorithms run on more than one processor core. This problem persisted on both Windows and Linux machines indicating that it is not a OS specific problem.

But despite these problems the algorithms still ran accurately as we were able to see in our replication of the original results. So we didn't need to vastly expand or modify the existing algorithms.

5 SUPPLEMENTARY MATERIAL

- (1) The complete set of plots, including all figures for every dataset and algorithm evaluated in this study, is available at:
<https://github.com/MStr41/ML/tree/main/Graphs>
- (2) The full source code, documentation, and additional resources related to this reproduction can be accessed at:
<https://github.com/MStr41/ML>

6 ACKNOWLEDGMENTS

This work was conducted as part of the Machine Learning Praktikum at the University of Siegen [2].

REFERENCES

- [1] Ardalan Arabzadeh, Tobias Vente, and Joeran Beel. 2024. Green Recommender Systems: Optimizing Dataset Size for Energy-Efficient Algorithm Performance. arXiv:2410.09359 [cs.LG] <https://arxiv.org/abs/2410.09359>

- [2] Joeran Beel and Lukas Wegmeth. 2025. Machine Learning Praktikum. *Universität Siegen* (2025).