Training Green and Sustainable Recommendation Models: Introducing Carbon Footprint Data into Early Stopping Criteria

Giuseppe Spillo
University of Bari Aldo Moro
Bari, Italy
giuseppe.spillo@uniba.it

Allegra De Filippo
Department of Computer Science and Engineering
DISI Università di Bologna
Bologna, Italy
allegra.defilippo@unibo.it

Emanuele Fontana
University of Bari Aldo Moro
Bari, Italy
e.fontana7@studenti.uniba.it

Michela Milano
DISI Universita' di Bologna
Bologna, Italy
michela.milano@unibo.it

Giovanni Semeraro
University of Bari Aldo Moro
Bari, Italy
giovanni.semeraro@uniba.it

Abstract

With the growing focus on Green AI, there is an urgent need for algorithms that are designed to minimize their environmental impact while maintaining satisfying performance. In this paper, we introduce a novel *early stopping* strategy that considers carbon footprint data while training a recommendation algorithm. In particular, during the training phase, our criterion epoch-by-epoch analyzes the improvement in terms of predictive accuracy and compares it to the increase in carbon emissions. Then, we analyze the trade-off between the scores, and when the accuracy improves at a rate that is not favorable, the training is stopped.

In the experimental evaluation, we showed that our strategy could significantly reduce the carbon footprint of several *state-of-the-art* recommendation models, with a limited decrease in accuracy and fairness. While more work is needed to automatically balance the trade-off between accuracy and emissions, this paper sheds light on the need for more sustainable recommendation models and takes a significant step toward designing *green* training strategies.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Sustainability, Recommender Systems, Green AI, Carbon Footprint

ACM Reference Format:
Giuseppe Spillo, Allegra De Filippo, Emanuele Fontana, Michela Milano, and Giovanni Semeraro. 2025. Training Green and Sustainable Recommendation Models: Introducing Carbon Footprint Data into Early Stopping Criteria. In *33rd ACM Conference on User Modeling, Adaptation and Personalization (UMAP '25), June 16–19, 2025, New York City, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3699682.3728336

UMAP '25, New York City, NY, USA

ACM ISBN 979-8-4007-1313-2/25/06
https://doi.org/10.1145/3699682.3728336

1 Introduction and Related Works

The last decade saw a significant advance of Artificial Intelligence (AI) technologies in several fields. However, while the accuracy and the effectiveness of these systems increase daily, this improvement implies more complexity. This complexity is manifesting in both *time-related issues* (AI models require longer training periods) and *space-related issues* (requiring more advanced hardware); for these reasons, AI is defined as an energy-intensive technology [3, 7, 32].

Both the above-mentioned issues led to a significant increase of CO_2 emissions of AI algorithms [12]. As an example, the training of recent large language models such as BLOOM and GPT-3 emitted approximately 522,000 and 50,000 kilograms of CO_2, respectively [22]. As a result, in the last few years, a new attention towards the development of more sustainable AI models is emerging, and the term *Green AI* [18, 25] was introduced to promote a vision of AI that also considers the *environmental* impact of the algorithms [23].

The study of sustainable and green aspects of AI has also taken hold in the Recommender Systems (RS) community [4]. Indeed, tools and metrics to evaluate carbon footprint [1, 31] were recently proposed, and [2, 10] called for prioritizing sustainable practices in RS design and training. In this research line, [19] benchmarked state-of-the-art (SOTA) recommendation models, concluding that simpler models (*e.g.*, Collaborative Filtering - CF) offered strong performance with low emissions, while more complex models (*e.g.*, Knowledge-Aware RSs - KARSs) emitted excessive CO_2 for marginal accuracy gains. Similar findings were reported in [24], highlighting that simpler models better balance accuracy and emissions.

While these analyses underscore the importance of addressing carbon footprint in the recommendation field, the challenge of reducing RS models' overall emissions while maintaining good predictive accuracy remains unresolved. A preliminary attempt in this direction is discussed in [20], where the authors explore the impact of data reduction strategies on the trade-off between performance and emissions, finding that these techniques often need an accurate case-by-case evaluation.

With this paper, we fit into this research direction and propose a Green Early Stop (GES) criterion that considers the information about the current carbon emissions during the training of the recommendation algorithm itself. To this end, we first introduce a

novel metric, the Accuracy Emission Ratio (AER), calculated as the ratio between the gain in performance (measured by the validation score) and the total emissions of the algorithm. Next, our GES strategy calculates the AER score at each epoch during the training. If the score is not favorable for a number of consecutive epochs, the training is stopped. As shown in the experimental evaluation, the proposed strategy significantly decreases the emissions of the algorithms at the cost of a small reduction in their predictive accuracy, thus confirming the validity of our intuition.

To our knowledge, this work is the first attempt at developing a strategy specifically designed to reduce the carbon emissions of a recommendation algorithm since the training phase. To sum up, this paper provides the following contributions:

(1) We introduce a Green Early Stopping criterion that also considers carbon emissions;
(2) We benchmark our GES strategy on several RSs and state-of-the-art datasets as well as in different scenarios;
(3) We identify strategies to make our criterion fully automatic, paving the way for further research.

2 Methodology

In this section, we first describe how we track the carbon footprint of RS models. Then, we introduce our novel GES criterion.

2.1 Measuring Carbon Footprint

Currently, the carbon dioxide equivalent ($CO_2 - eq$) is a widely adopted metric used to measure and compare greenhouse gas emissions [8], and, consequently, the carbon footprint[1]. Following [16], we estimate it by multiplying the *power consumption (PC)* and the *carbon intensity (CI)*. Formally:

$$emissions = PC \cdot CI \tag{1}$$

PC refers to the electrical power required by the hardware (CPUs, GPUs, memory, etc.) to perform computations and can be measured through low-level hardware interfaces. It is expressed in kilowatt-hours. Next, CI is calculated as the weighted average of the carbon footprint of each energy source in the energy mix, and with a fixed amount of CO_2 emitted per kilowatt-hour for each energy source. It is important to note that different countries use different energy mixes, meaning each mix (and thus each country) has its own carbon intensity[2].

In this paper, we used the Python library CodeCarbon[3] to measure the emissions and to implement our methodology; this choice is in line with other works in the field [19, 20]. First, it considers the energy mix of the country where the computation is performed, allowing for an accurate calculation of CI. Second, it measures the power consumption of each hardware component throughout the computation. As a result, at the end of the computation, it provides a measure of the emitted $CO_2 - eq$. The use of CodeCarbon is in line with other work in the area, such as [19, 20].

[1]https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Carbon_dioxide_equivalent
[2]Data available at: https://ourworldindata.org/electricity-mix
[3]https://mlco2.github.io/codecarbon/

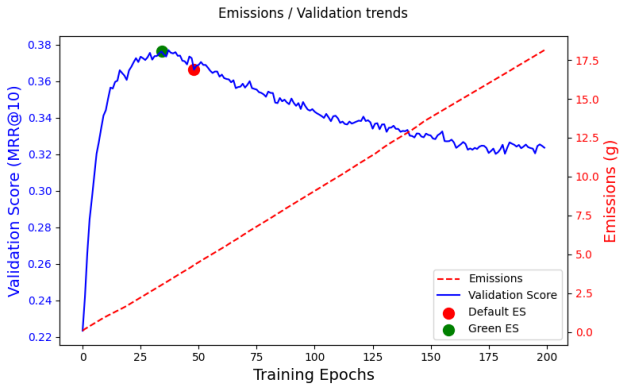


Figure 1: An example of the accuracy/emission curve graph

2.2 Introducing Green Early Stop

The early stopping strategies that are commonly used to train RS rely on the analysis of the validation scores epoch by epoch. Differently, our novel GES criterion also considers the amount of $CO_2 - eq$ emitted to decide whether to stop the training ahead of time.

To implement our strategy, we rely on two novel constructs: *(a)* the Accuracy Emission Ratio, calculated as the ratio between the gain in performance and the increase in carbon emissions in two consecutive epochs of training; *(b)* the *tolerance* of the GES criterion when monitoring the AER epoch by epoch.

In particular, AER is inspired by the finite difference method [13], which is designed to approximate the derivatives (the *rates of change*) of a function by using the difference between function values at discrete points. In our context, the function is the model's accuracy given by the validation score. At a given epoch, the slope of this curve represents how favorable the gain in accuracy is with respect to the increase in emissions. Based on the slope, we can decide whether to stop the training or not. Formally, AER can be defined as follows:

$$AER(r) = \frac{v\_score_{i+1} - v\_score_i}{e_{i+1} - e_i} \tag{2}$$

where r is a recommendation algorithm, v_score_i and v_score_{i+1} are the validation metric values observed at training epochs i and $i+1$, respectively, and e_i and e_{i+1} are the CO_2 emissions produced after i and $i+1$ training epochs. Based on these constructs, we can sketch our GES criterion as follows:

(1) For each epoch, we calculate the AER score. Any accuracy metric, i.e., NDCG, can be used as a validation metric.
(2) Based on the size of the datasets and on the characteristics of the recommendation algorithm, *i.e.,* average emission, we set a threshold score for AER. Let n be the threshold.
(3) If the AER score is lower than n, it means that the increase is unfavorable. If this holds for a consistent number of epochs (*i.e.,* the tolerance), the training is stopped.

Generally speaking, the training stops when, for too many epochs, the improvement in predictive accuracy is not justified by the increase in carbon emissions. Formally, let k be the tolerance and

r be the recommendation algorithm; the training is stopped if $AER(r) < n$ for more than k consecutive epochs. Of course, the higher the tolerance, the higher the number of epochs not surpassing the threshold that is needed to stop the training. Conversely, low tolerance values may stop the training even after a few epochs, where the ratio between accuracy and emissions is unfavorable.

Figure 1 sketches an example that illustrates the underlying intuition. At the beginning of the training, the validation score increases rapidly, as shown by the blue line in the picture. However, after some epochs, the improvement of the validation score decreases; in this case, the training process could be stopped after few epochs (red dot), but our GES is able to stop it even before (green dot), by considering the emission procuded during the training process (red dotted line). This results in a general lower carbon footprint.

2.3 Discussion

In this section, we deeply discuss the intuition behind the design of our GES criterion, and why it can be considered an effective solution to better trade-off performance and carbon footprint of RS models.

Generally speaking, traditional early-stop approaches are able to stop the training of a model if no improvements are observed for a number of subsequent epochs. On the other hand, our GES takes into account the AER (defined in Equation 2), in which the carbon footprint is considered at the denominator, and linearly increases epoch by epoch (as depicted in Figure 1).

Although, given a model, the emission trend increases linearly over time, each model exhibits a different slope: more complex models emit more CO_2 per epoch, resulting in a steeper line (i.e., a higher slope coefficient), while simpler models have a lower carbon footprint per epoch, corresponding to a less steep line (i.e., a lower slope coefficient). In this context, the strenght of our GES relies in giving greater reward to models that, *for the same improvement in the validation metric, have a lower carbon footprint*. In other words, models with higher carbon footprints should guarantee, for each epoch, a higher gain in accuracy as well to justify the training process continuation.

3 Experimental Session

We designed experiments aiming at answering the following **Research Questions (RQs)**:

RQ1 - Emissions Impact: *how does our early stopping criterion affect the emission of RS models?*

RQ2 - Impact on Recommendation Metrics: *how does our early stopping criterion affect the performance of RS models?*

RQ3 - Trade-off analysis: *Can our early stopping criterion be considered a possible solution to improve the trade-off between emissions and performance of RS models?*

3.1 Experimental Design

Dataset. To answer our RQs, we performed experiments on a state-of-the-art dataset, MovieLens-1M, that is available on our GitHub repository[4]. This dataset includes 6,036 users, 3,192 items, and 946,772 interactions, with a sparsity of 96.07%. On average, each user and item have 125 and 237 interactions, respectively. The KG provided with the dataset encodes 20,454 entities, 13 relations, and 70,668 triples, with 2,823 items linked to it. Due to space reasons, it was not possible to extend the analysis to further datasets.

Recommendation Algorithms. In this paper, we have selected 11 SOTA RS algorithms representative of the various approaches. In particular, we considered models falling in the following families: linear and collaborative filtering (CF) [11]; Graph Neural Networks (GNNs) and Graph Convolutional Networks (GCNs) [27, 33]; Knowledge-Aware RSs (KARSs) [6]. To ensure the replicability, we limit to the models implemented in RecBole [37]. In particular, we considered BPR [17], DMF [34], LINE [21], MultiDAE [14] as CF algorithms, NGCF [29], DGCF [30], LightGCN [9] as GNNs, and CKE [35], CFKG [36], KGCN [28], KGNNLS [26] as KARS.

GES Parameters. As previously stated, the threshold score used to evaluate whether AER is favorable strongly depends on both the dataset (*i.e.*, number of users, number of items) and the average emissions of the algorithms. In this work, we evaluate two different threshold values: 30 and 40. From now on, we will refer to *low* and *high* threshold values. As for the tolerance, we evaluate three different values, *i.e.,* 5, 6, and 7. In this case, we will refer to them as *low*, *moderate*, and *high* tolerance. All these values are set through heuristics after several runs of the pipeline and consistently with the data orders of magnitude. Of course, this is the first exploratory step in this direction within RS. For future work, a fully automatic strategy to automatically define the optimal values for threshold and tolerance is needed to improve the soundness of the approach.

Implementation Details. To ensure fair comparisons and reproducible results, we considered RecBole as a recommendation library. Accordingly, we implemented our GES criterion in this framework so that it can be easily reproduced. We conducted our experiments on a Ubuntu 20.04.6 LTS system equipped with an AMD EPYC 7V12 64-Core CPU, an NVIDIA Tesla T4 16 GB GPU, and CUDA driver Version 12.2. The source code for implementing our GES criterion, along with the scripts to run and reproduce our experiments, is available on our repository (see Footnote 4). To track the carbon footprint of the RS, we used CodeCarbon library[5].

Evaluation Metrics. We considered the MRR as an accuracy metric to compute the AER, as it is the default validation metric used by RecBole; further metrics will be considered in future works (*e.g.,* NDCG). We evaluated the RS algorithm performance in terms of Recall, NDCG, Gini index, and Average Popularity. Due to space reasons, we only present results related to Recall and Gini index. Full results are available on our repository.

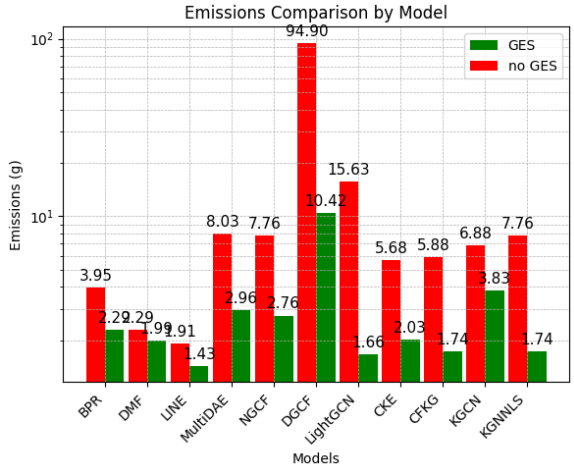
3.2 Results of the Experiments

In Figure 2a, we report how the application of our GES criterion affects the $CO_2 - eq$ emissions of recommendation models. Due to space reasons, we only report the results obtained in the *high* threshold and *low* tolerance setting. Accordingly, it means that the algorithm continues the training only if the performance rapidly increases. Conversely, when the slope of the curve decreases, even for just a few epochs, the training stops. In other terms, this is the most *aggressive* early stopping configuration we evaluated.

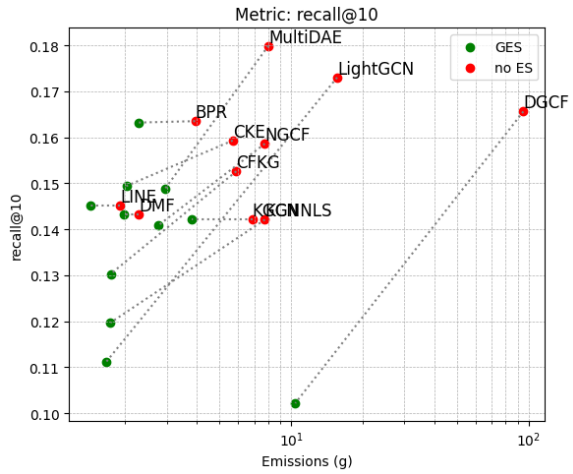
As shown in Figure 2a, GES strongly affects the carbon footprint of RSs: this holds for complex models like LightGCN and DGCF,

[4]https://github.com/swapUniba/GES-green-early-stop

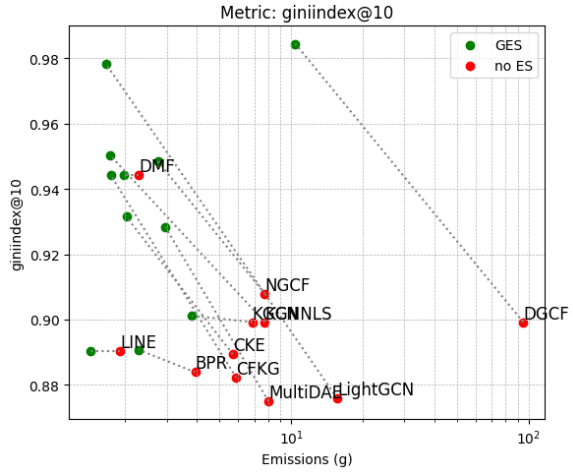
[5]https://mlco2.github.io/codecarbon/



(a) Impact of our GES criterion on emissions



(b) Impact of our GES criterion on Recall



(c) Impact of our GES criterion on Gini index

Figure 2: Results obtained by applying our GES criterion, with *high* treshold and *low* tolerance. *a)* compares the emissions produced by each model whether the GES criterion is applied (green) or not (red); *b)* and *c)* focus on the trade-off between $C0_2$ emissions (*X-axis*) and recommendation performance *(Y-axis).*

emitting a $CO_2 - eq$ that is approximately 9-times lower, as well as for simpler methods like BPR, whose emissions are halved when GES is applied. Even if this is an expected outcome, it is important to quantify the number of emissions that are saved by introducing carbon footprint data directly into the training of recommendation models. A more comprehensive overview of this finding is provided in Table 1, which shows the percentage decrease when GES is applied on varying of different thresholds and tolerance values. As expected, the higher the tolerance, the lower the amount of emissions that are saved. Conversely, higher thresholds and lower tolerance lead to a reduction that may almost reach 90% of the emissions, as shown by LightGCN and DGCF. **Based on these results, we can answer RQ1 by stating that our GES criterion succeeds at decreasing the $CO_2 - eq$ emitted during the training of RS models. For complex models, the amount of emissions that are saved can be considerable.**

Next, to answer RQ2, we can observe Figures 2b and 2c, in which we note that a decrease in performance is generally obtained when GES is applied. Indeed, most of the lines have a decreasing trend towards left (for recall, where *higher is better*) or towards right (for Gini index, where *lower is better*). This particularly holds for models emitting more $CO_2 - eq$, such as DGCF, LightGCN, and MultiDAE. However, it is worth noting that other models, including BPR, DMF, and KGCN, do not decrease their performance, and some techniques, *i.e.*, CKE, are minimally affected by GES. This result is also confirmed in Table 1. As we already discussed for the previous RQ, higher tolerance generally leads to the performance being closer to the original ones. However, the reduction in emissions is lower as well. **To sum up, we can answer RQ2 by stating that GES generally decreases the performance of some RS, but this is not valid for all the algorithms. In some settings, just a tiny decrease in performance is noted.**

Finally, we can answer RQ3 and assess whether GES can be considered a solution to trade-off performance and carbon footprint or RS models. As previously stated, some models, such as DGCF and LightGCN, strongly reduce their emissions at the cost of an equivalent reduction in performance. Other models, such as MultiDAE or CFKG, are characterized by a lower decrease in performance. Generally speaking, one of the best trade-offs emerges for CKE. Indeed, this model reduces its emissions by approximately 60%, with a tiny performance decrease. Overall, it is the second best-performing approach in terms of recall after the application of GES. Finally, it is worth mentioning that the performance of simpler models such as LINE, DMF, and BPR is not affected by GES. In particular, BPR obtained the best accuracy when early stopping was applied. **To address RQ3, we can conclude that the different algorithms show a heterogeneous behavior. Generally speaking, the impact of GES seems to be scenario-dependent. However, simpler algorithms generally provide a good trade-off between the metrics.**

4 Discussion and Conclusions

In this paper, we have introduced a novel strategy to reduce the emissions of a recommendation algorithm by design, *i.e.*, by early stopping the training when the ratio between the increase in performance and the amount of emissions is not favorable. This is done by

	high treshold									*low treshold*								
Model	*low tolerance*			*moderate tolerance*			*high tolerance*			*low tolerance*			*moderate tolerance*			*high tolerance*		
	Emiss.	Rec	Gini	Emiss.	Rec	Gini	Emiss.	Rec	Gini	Emiss.	Rec	Gini	Emiss.	Rec	Gini	Emiss.	Rec	Gini
BPR	-42.01	-0.18	-0.77	-37.12	0	0	-17.99	0	0	-36.13	0	0	-20.93	0	0	-19.97	0	-
DMF	-13.27	0	0	-12.87	0	0	-13.08	0	0	-9.03	0	0	-12.93	0	0	-13.53	0	-
LINE	-25.03	0	0	-27.02	0	0	-27.42	0	0	-25.17	0	0	-27.14	0	0	-24.98	0	-
MultiDAE	-63.68	-17.23	6.09	-62.04	-14.90	5.19	-60.33	-13.90	4.97	-50.49	-10.01	4.01	-53.42	-9.73	3.47	-50.25	-8.00	3.04
NGCF	-64.46	-11.22	4.46	-63.64	-9.33	4.43	-59.43	-6.75	3.50	-65.30	-11.22	4.46	-62.81	-9.33	4.43	-59.72	-6.75	3.50
DGCF	-89.02	-38.29	9.49	-87.48	-34.78	9.21	-85.89	-32.91	8.89	-89.03	-38.35	9.49	-87.29	-34.78	9.21	-85.90	-32.85	8.89
LightGCN	-89.35	-35.78	11.69	-88.85	-34.91	11.51	-88.14	-32.89	11.29	-87.84	-32.89	11.29	-86.46	-30.40	10.85	-86.66	-30.40	10.85
CKE	-64.34	-6.21	4.73	-63.39	-4.77	4.18	-59.18	-3.14	3.44	-63.99	-6.21	4.73	-63.44	-4.77	4.18	-59.23	-3.14	3.44
CFKG	-70.39	-14.61	7.04	-69.33	-14.74	6.56	-67.39	-11.93	6.18	-67.91	-14.74	6.56	-67.35	-11.93	6.18	-17.81	0	-
KGCN	-44.29	0	0.22	-42.25	0	-0.09	-41.27	0	0	-44.79	0	0.22	-42.85	0	-0.09	-41.21	0	-
KGNNLS	-77.61	-15.76	5.67	-75.90	-13.30	5.09	-73.95	-11.68	4.76	-43.76	0.14	0.23	-38.19	0	-0.10	-36.07	0	-

Table 1: Sensitivity analysis to the threshold value n and the number of consecutive epochs k. In this table, we encode the percentage change of emissions, recall and Gini index according to n and k parameter values. All these values are percentages.

introducing a new metric inspired by the finite difference method that compares the slope of the curves modeling performance and emissions epoch by epoch. Overall, our analysis showed that the method effectively reduces the emissions of recommendation algorithms, even at a significant rate for carbon-intensive models such as LightGCN and DGCF. However, especially for more complex techniques, the decrease in emissions often comes at the cost of a decrease in performance. This outcome is not surprising. Indeed, as shown in [5], it has been acknowledged since the early 90s that early stopping methods follow the *no-free-lunch* theory. Accordingly, the reduction in terms of emissions needs to be balanced by a decrease in performance.

Generally speaking, the identification of the ideal trade-off between these dichotomic aspects is not trivial. In some settings, it may be fundamental to reduce emissions, regardless of the loss in performance. Conversely, in particular domains, the accuracy of the algorithms may be indispensable, so the *tolerance* needs to be increased. This is a domain- and scenario-specific design choice.

While the automatic tuning of such a trade-off is left as future work, we have already identified some mechanisms to extend the criterion and make it fully automatic. Indeed, the main issue we encountered was the different magnitudes of the values of performance and emissions, which made it hard to define a fixed threshold to compare the curves. Possible solutions may include: *(a)* the adoption of a logarithmic formula that flattens the curves and gets their magnitude closer; *(b)* the use of a *min-max* normalization strategy, where the normalization is obtained by considering as *maximum* scores the performance and the emissions benchmarked in previous related works, such as [19, 24].

These directions will be investigated as future work, together with a more extensive analysis considering more datasets, more algorithms, including explainable recommender systems [15], and more values for threshold and tolerance. However, these limitations do not hinder the validity of the findings. Indeed, this paper sheds light on the need for more sustainable recommendation models and takes a significant step toward designing *green* training strategies, showing that novel early stopping methods can be a suitable solution for the problem.

Acknowledgments

This research is partially funded by the PNRR project FAIR—Future AI Research (PE00000013), Spoke 6—Symbiotic AI and Spoke 8-Pervasive AI, under the NRRP MUR program supported by NextGenerationEU (CUP H97G22000210007), and the PHaSE project — Promoting Healthy and Sustainable Eating through Interactive and Explainable AI Methods, funded by MUR under the PRIN 2022 program - Finanziato dall'Unione europea - NextGeneration EU, Missione 4 Componente 1 (CUP H53D23003530006).

References

[1] Ashmi Banerjee, Tunar Mahmudov, Emil Adler, Fitri Nur Aisyah, and Wolfgang Wörndl. 2025. Modeling sustainable city trips: integrating CO 2 e emissions, popularity, and seasonality into tourism recommender systems. *Information Technology & Tourism* (2025), 1–38.

[2] Joeran Beel, Alan Said, Tobias Vente, and Lukas Wegmeth. [n. d.]. Green Recommender Systems–A Call for Attention. ([n. d.]).

[3] Stefano Bianchi, Allegra De Filippo, Sandro Magnani, Gabriele Mosaico, and Federico Silvestro. 2021. Virtus project: a scalable aggregation platform for the intelligent virtual management of distributed energy resources. *Energies* 14, 12 (2021), 3663.

[4] Ludovico Boratto, Allegra De Filippo, Elisabeth Lex, and Francesco Ricci. 2024. First International Workshop on Recommender Systems for Sustainability and Social Good (RecSoGood 2024). In *Proceedings of the 18th ACM Conference on Recommender Systems*. 1239–1241.

[5] Zehra Cataltepe, Yaser S Abu-Mostafa, and Malik Magdon-Ismail. 1999. No free lunch for early stopping. *Neural computation* 11, 4 (1999), 995–1009.

[6] Janneth Chicaiza and Priscila Valdiviezo-Diaz. 2021. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information* 12, 6 (2021), 232.

[7] Allegra De Filippo, Michele Lombardi, and Michela Milano. 2016. Non-linear optimization of business models in the electricity market. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*. Springer, 81–97.

[8] Andrew John East. 2008. What is a carbon footprint? An overview of definitions and methodologies. In *Vegetable industry carbon footprint scoping study—Discussion papers and workshop, 26 September 2008*. Horticulture Australia Limited Sydney.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 639–648. doi:10.1145/3397271.3401063

[10] Dietmar Jannach, Alan Said, Marko Tkalčič, and Markus Zanker. 2024. Recommender Systems for Good (RS4Good): Survey of Use Cases and a Call to Action for Research that Matters. *arXiv preprint arXiv:2411.16645* (2024).

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[12] Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green algorithms: quantifying the carbon footprint of computation. *Advanced science* 8, 12 (2021), 2100707.

[13] Changpin Li and Fanhai Zeng. 2012. Finite difference methods for fractional differential equations. *International Journal of Bifurcation and Chaos* 22, 04 (2012), 1230014.

[14] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[15] Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2021. Generating post hoc review-based natural language justifications for recommender systems. *User Modeling and User-Adapted Interaction* 31, 3 (2021), 629–673.

[16] Divya Pandey, Madhoolika Agrawal, and Jai Shanker Pandey. 2011. Carbon footprint: current methods of estimation. *Environmental monitoring and assessment* 178 (2011), 135–160.

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[18] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM* 63, 12 (2020), 54–63.

[19] Giuseppe Spillo, Allegra De Filippo, Cataldo Musto, Michela Milano, and Giovanni Semeraro. 2023. Towards sustainability-aware recommender systems: analyzing the trade-off between algorithms performance and carbon footprint. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 856–862.

[20] Giuseppe Spillo, Allegra De Filippo, Cataldo Musto, Michela Milano, and Giovanni Semeraro. 2024. Towards green recommender systems: Investigating the impact of data reduction on carbon footprint and algorithm performances. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 866–871.

[21] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[22] Bill Tomlinson, Rebecca W Black, Donald J Patterson, and Andrew W Torrance. 2024. The carbon emissions of writing and illustrating are lower for AI than for humans. *Scientific Reports* 14, 1 (2024), 3732.

[23] Aimee Van Wynsberghe. 2021. Sustainable AI: AI for sustainability and the sustainability of AI. *AI and Ethics* 1, 3 (2021), 213–218.

[24] Tobias Vente, Lukas Wegmeth, Alan Said, and Joeran Beel. 2024. From Clicks to Carbon: The Environmental Toll of Recommender Systems. In *Proceedings of the 18th ACM Conference on Recommender Systems* (Bari, Italy) *(RecSys '24)*. Association for Computing Machinery, New York, NY, USA, 580–590. doi:10.1145/3640457.3688074

[25] Roberto Verdecchia, June Sallou, and Luís Cruz. 2023. A systematic review of Green AI. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2023), e1507.

[26] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 968–977. doi:10.1145/3292500.3330836

[27] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*. 3307–3313.

[28] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3307–3313. doi:10.1145/3308558.3313417

[29] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 165–174. doi:10.1145/3331184.3331267

[30] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1001–1010. doi:10.1145/3397271.3401137

[31] Lukas Wegmeth, Tobias Vente, Alan Said, and Joeran Beel. 2024. Emers: Energy meter for recommender systems. *arXiv preprint arXiv:2409.15060* (2024).

[32] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.

[33] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[34] Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 3203–3209.

[35] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 353–362. doi:10.1145/2939672.2939673

[36] Yongfeng Zhang, Qingyao Ai, Xu Chen, and Pengfei Wang. 2018. Learning over knowledge-base embeddings for recommendation. *arXiv preprint arXiv:1803.06540* (2018).

[37] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM*. ACM, 4653–4664.