Scoring Frugality for Remote Sensing Data Processing Algorithms

Matthieu Verlynde
LISTIC Université Savoie Mont Blanc
Annecy, France
matthieu.verlynde@univ-smb.fr

Ammar Mian
LISTIC Université Savoie Mont Blanc
Annecy, France
ammar.mian@univ-smb.fr

Yajing Yan
LISTIC Université Savoie Mont Blanc
Annecy, France
yajing.yan@univ-smb.fr

***Abstract*—Assessing the frugality of data processing algorithms has become a priority in the machine learning and artificial intelligence community, and the use of extensive data processing algorithms increases exponentially for remote sensing applications. However, identifying a quantifiable measure of this dimension combining performance and energy consumption is a complex task. This paper proposes both a methodology to collect energy consumption data and the use of three different frugality scoring methods. Through a case study of two classical data processing tasks in remote sensing, change detection and clustering, we show that these three scores encompass different aspects of frugality altogether and suggest a combined approach by users.**

***Index Terms*—frugality, multi-criteria index, energy consumption, remote sensing, data processing**

I. Introduction

While machine learning and artificial intelligence tools became prominent processing methods in remote sensing [1], the ecological impact of AI tools and machine learning models is growing exponentially [2], [3]. Therefore, it is imperative to expand research on both the evaluation and the improvement of their frugality [2] for remote sensing practitioners. In this paper, we call frugality the aim for a low energy consumption with a guarantee of a satisfactory performance of the method used.

Measuring frugality can be challenging for the ecological or energetic cost of a method is often estimated by the running time [4], [5]. It has been shown that runtime does not capture all the information about energy consumption [6]. Other metrics such as the algorithm complexity or number of lines of code can be used [7], [8] but also only give an estimation of a theoretical running cost, hardly translatable into carbon footprints. Thus several energy consumption measuring software have been published such as *carbontracker* [9], CodeCarbon [10] or Experiment Impact Tracker [11]. Finally, another approach is to directly measure the energy consumption of the system used through an external connected device. Given that the measurement frequency is high enough to acquire precise information about the energy consumption during runtime, this approach draws an accurate picture of the frugality of the studied method.

Evaluating frugality involves combining the estimated energy consumption of the method with its performance. Several studies focus on combining the performance of a model with its running time [5], [12], but the literature lacks studies scoring methods that use their actual energy consumption. In this regard, many different approaches exist to create a multi-criterion index. If using weighted sums to aggregate metrics has been criticized [13], it is still a common multi-criteria scoring method [14]–[16]. The use of fuzzy-logic-based approaches has also spread for this type of task [17], [18], but it requires expert knowledge to fit the decision boundaries and could be poorly scalable for different data processing methods.

In this paper, we propose three main contributions to assess these issues of estimation of frugality in remote sensing :

- an energy consumption and performance measuring pipeline for frugality evaluation,
- a framework of metric combination to assess frugality through three scoring methods,
- a study case on two classical methods for remote sensing data, change detection and clustering.

II. Measuring Energy Consumption

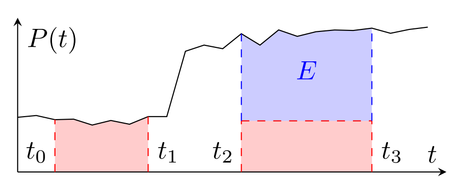


Fig. 1: Example of energy consumption measure E over data processing method runtime

The scoring of frugality requires accessing internal and external hardware and software measures. In this work, we focus on the global energy consumption measured using a smart plug connected to an InfluxDB database through a Z-Wave protocol. Measurements are made through three periods of time: a standard period with no model training, a latent time with a single training execution, and a running period. Let $P(t)$ be the power measure at time t, t_0 the start of the standard period until t_1 the start of the latent time, and t_2 the start of the running period until t_3. These periods are represented in Fig.1. The measured energy consumption E, used to evaluate frugality, is then calculated following eq.1:

$$E = \int_{t_2}^{t_3} P(t)\,dt - \frac{t_3 - t_2}{t_1 - t_0} \int_{t_0}^{t_1} P(t)\,dt. \quad (1)$$

The standard period allows us to not take account of the background processes of the machine used to run the algorithm of interest. The latent time consists of a period during which the algorithm of interest runs once, giving time for the machine to warm up and reach a running state with higher temperature. This workflow ensures to collect only the energy consumption measure due to the execution of the algorithm.

Once these data are collected, the performance of the model is measured according to the task. Evaluating frugality then requires the aggregation of both measurements for which a framework is presented in the next section.

III. Scoring Frugality

Creating a frugality score requires combining both the performance of the model α on the targeted task and the energy consumption β during runtime. A common way to combine two metrics is to use a weighted sum. However, this method requires normalizing the metrics of interest, which introduces a bias in the data. We set $\epsilon \in [0,1]$ the weight given to the performance of the method compared to the energy consumption, and $\alpha_n \in [0,1]$ and $\beta_n \in [0,1]$ the normalized values of α and β respectively. This score s_{WS} is then shown in eq.2:

$$s_{WS} = \epsilon \times \alpha_n + (1-\epsilon) \times (1-\beta_n) \quad (2)$$
$$\text{where } \alpha_n = \frac{\alpha - \min(\alpha)}{\max(\alpha) - \min(\alpha)},\ \beta_n = \frac{\beta - \min(\beta)}{\max(\beta) - \min(\beta)}.$$

Another method combining these metrics is the harmonic mean, similarly as the calculation of the F-measure for the precision and recall metrics for a classification task. We choose this method because it is particularly adequate for combining ratios [19], therefore, for our normalized metrics α_n and β_n. We set $\kappa \in \mathbb{R}^+$ the weight given to the energy consumption and the harmonic mean s_{HM} is calculated as shown in eq.3:

$$s_{HM} = (1+\kappa^2)\frac{\alpha_n(1-\beta_n)}{\alpha_n + \kappa^2(1-\beta_n)}. \quad (3)$$

Because these two scores require normalization of both the performance of the method and the energy consumption metrics, we suggest using a frugality score s_F inspired by [5] based on both metrics α and β and a weight $w \in [0,1]$ given to the available resource—here the energy consumption—as opposed to the model performance. This score is shown in eq.4 and the dependence on the weight w is unique for each model and input data used:

$$s_F = \alpha - \frac{w}{1+\frac{1}{\beta}}. \quad (4)$$

In the next section, we will present the two study cases on which these frugality scores will be applied as well as the obtained results.

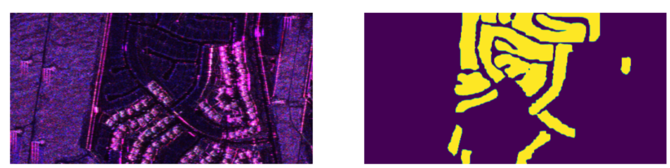
IV. Results

A. Methodology

To evaluate the performance of our frugality measure on remote sensing applications, we focus on a clustering task and a change detection task—two classical tasks in data processing.

The clustering task is an unsupervised approach. Because of its complexity, the literature lacks benchmark clustering datasets based on remote sensing data. Therefore, in this paper, classical clustering approaches were used on a toy dataset presenting 5 blobs of data along 5 features, adapted to 5-group clustering models. The selected clustering methods included : two hierarchical clustering methods, **Agglomerative Clustering** using the average of the cityblock distances of each observation to form clusters, and **Ward** using the variance of the clusters ; four density based methods, **DBSCAN**, **OPTICS**, **HDBSCAN** and **GMM** ; and both the **K-Means** and **K-Means++** models. The parameters of each model were chosen to maximize the adjusted Rand index (ARI) calculated between the clustering result and the initial known clusters defined by the data blobs. Based on the Rand index RI with n the number of instances in the data, a the number of pairs of instances in the same cluster both in the clustering result and the data ground truth, and b the number of pairs of instances in different clusters both in the clustering result and the data ground truth, the ARI presented in eq.5 standardizes the RI by the expected RI guaranteeing that a random clustering returns a zero value:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}(\text{RI})}{\max(\text{RI}) - \mathbb{E}(\text{RI})} \text{ where RI} = \frac{a+b}{\binom{n}{2}}. \quad (5)$$



(a) Composite RGB SAR image at $t = 0$.

(b) Change detection ground truth

Fig. 2: Change detection data from the UAVSAR (Courtesy NASA/JPL-Caltech) database [20], [21]

To evaluate the frugality measurement strategy on the change detection task, three algorithms were used on the same remote sensing data. This data consists in a 2 multi-band PolSAR image time series from the UAVSAR (Courtesy NASA/JPL-Caltech) database [20]. Ground truth data, presented in Fig.2, was also available from [21]. Each image was cropped to a size of 1000×500 pixels. To consider the multivariate aspect of the data (the three polarization bands of each image) and the speckle noise, we use methods based on the covariance matrices between each band at each pixel. Thus with p the number of bands per image, T the number of multi-band images, N the number of pixels per band, we consider $x_k^{(t)}$ for $t \in [\![1, T]\!]$ and $k \in [\![1, N]\!]$ a sample of pixels of all bands in a sliding spatial window of the image. For this task, we consider $x_k^{(t)}$ the realization of a random vector following a probability model $p_x(x_k, \theta_t)$ where θ_t is the set of parameters at time t. The change detection is then defined as the detection of change of these parameters, so comparing two hypotheses H_0 and H_0 following eq.6:

$$\begin{cases} \mathsf{H}_0 : \theta_0 = \theta_1, \\ \mathsf{H}_1 : \theta_0 \neq \theta_1. \end{cases} \tag{6}$$

Our test statistics are base on a calculated empirical covariance matrix $S_k^{(t)}$ equal to $x_k^{(t)} x_k^{(t)\mathsf{H}}$. The first test statistic used is the Generalized Likelihood Ratio Test statistic (GLRT). We consider this method under the Gaussian distribution hypothesis of the pixel values, and thus we refer to this method **G-GLRT** in this paper. The calculation of the test statistic $\hat{\Lambda}_{\mathrm{G}}$ used in this case is shown in eq.7:

$$\hat{\Lambda}_{\mathrm{G}} = T^{pkT} \frac{\prod_{t=1}^{T} |\mathsf{S}_k^{(t)}|^k}{|\prod_{i=1}^{T} \mathsf{S}_k^{(t)}|^{kT}} \text{ where } \mathsf{S}_t \sim W_C(p, k, \Sigma_t). \tag{7}$$

Here we consider the covariance matrices $S_k^{(t)}$ as realizations of independent random variables $\mathsf{S}_k^{(t)}$ that follow complex Wishart distributions. The implementation chosen for this method was proposed by [22] with a pairwise approach between successive images along the time series to detect changing points.

The second method used is a GLRT method extended to non-Gaussian distributions on PolSAR images, proposed by [20], using a texture information within the data. This method is referred as **NG-GLRT** in this paper. Its associated test statistic $\hat{\Lambda}_{\mathrm{NG}}$ is shown in eq.8.

$$\hat{\Lambda}_{\mathrm{NG}} = \frac{|\hat{\Sigma}_0^{\mathrm{NG}}|^{TN}}{\prod_{t=1}^{T} |\hat{\Sigma}_t^{\mathrm{TE}}|^N} \prod_{k=1}^{N} \frac{\left(\sum_{t=1}^{T} q\left(\hat{\Sigma}_0^{\mathrm{NG}}, x_k^{(t)}\right)\right)^{Tp}}{T^{Tp} \prod_{t=1}^{T} \left(q\left(\hat{\Sigma}_0^{\mathrm{TE}}, x_k^{(t)}\right)\right)^p} \tag{8}$$

where $q(\Sigma, x) = x^H \Sigma^{-1} x,\ \hat{\Sigma}_0^{\mathrm{NG}} = \frac{p}{N} \sum_{k=1}^{N} \frac{\sum_{t=1}^{T} S_k^{(t)}}{\sum_{t=1}^{T} q\left(\hat{\Sigma}_0^{\mathrm{NG}}, x_k^{(t)}\right)}$,

$\forall t \in [\![1, T]\!],\ \hat{\Sigma}_t^{\mathrm{TE}} = \frac{p}{N} \sum_{k=1}^{N} \frac{S_k^{(t)}}{q\left(\hat{\Sigma}_t^{\mathrm{TE}}, x_k^{(t)}\right)}$.

Compared to the previous method, this test statistic involves the estimation of fixed point, which is computationally extensive. Both the G-GLRT and NG-GLRT methods were applied using three window sizes of 5, 7 and 21 pixel width to calculate covariance matrices within the images. Using this values allows one to understand the effect of this key parameter on the energy consumption of the methods.

Finally, a simple direct approach used to estimate changing pixels was using the log difference—the difference of the log value of the inter-band mean value for each pixel between two images. By definition, this method applies only on 2-image time series. The statistic used is then named $\hat{\Lambda}_{\mathrm{LD}}$ and presented in eq.9. This method is cited as **LogDiff** in this paper.

$$\hat{\Lambda}_{\mathrm{LD}} = \frac{1}{p} \left(\sum_{\text{bands}} \ln x^{(t_1)} - \sum_{\text{bands}} \ln x^{(t_0)} \right). \tag{9}$$

The performance of these three change detection models was evaluated using the Area Under the Curve (AUC) based on the test statistics previously explained. The application of these change detection and clustering methods produced a set of performance and energy consumption data were carried out following an experiment setup presented in the next subsection.

B. Experiment setup

The experiments were carried out using implementations relying exclusively on CPU, avoiding GPU usage with a higher energy consumption value. Thus the implementations used a CPU Intel i5-12600 3.30GHz and a 2 × 32 Go RAM, and a Smart Switch 7 Aeotec® with a USB Z-Stick 7 Aeotec® controller. All the code is available on GitHub[1].

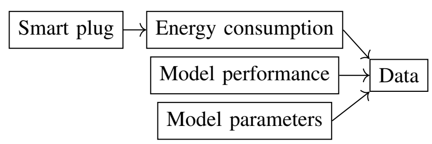


Fig. 3: Experiment data collection pipeline

For statistical significance of the energy consumption measure, each run is repeated 30 times. For each run, as shown in Fig.3, the energy consumption data is stored as well as the performance of the method and its parameters. The results obtained are then analysed in the next subsection.

C. Results obtained

During the running time of each model, their performance and energy consumption were measured, and each frugality score was calculated to compare their relative frugality.

1) Clustering: The measurements for the clustering task show significantly different energy consumption depending on the clustering method, as presented in Tab.I. The K-Means and K-Means++ models show better performance results, and a lower energy consumption for K-Means.

Overall, the three scoring results presented in Fig.4 agree on the ranking of the methods on their frugality, the methods K-Means, K-Means++, GMM and Ward having higher scores than the other methods. Indeed, the DBSCAN, HDBSCAN, OPTICS and and Agglomerative Clustering methods show poor clustering performance according to their measured ARI.

[1]https://github.com/MattVerlynde/frugal-score-2025.git

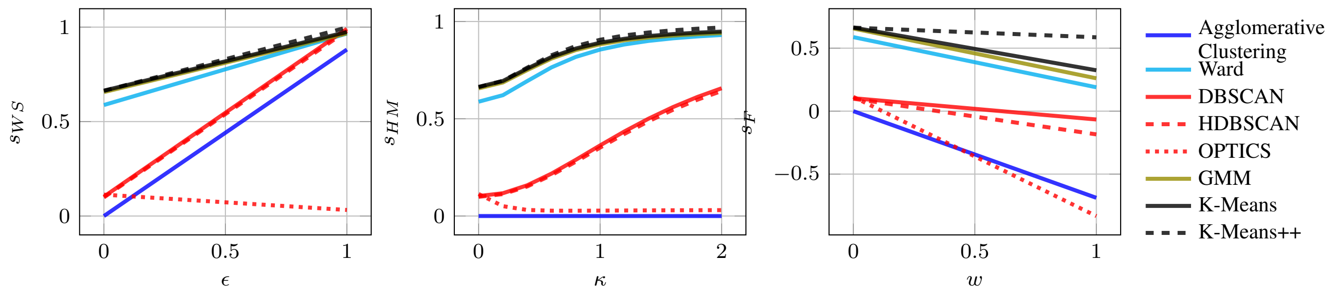


Fig. 4: Mean frugality scores on the clustering task

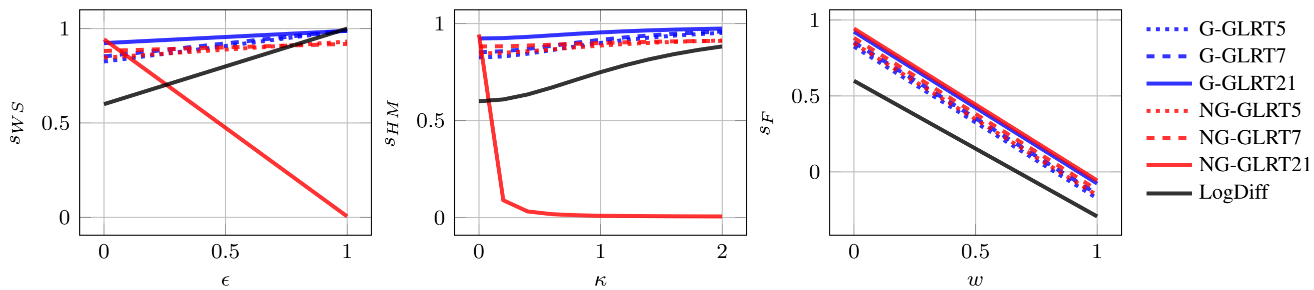


Fig. 5: Mean frugality scores on the change detection task

TABLE I: Energy consumption of clustering methods (average and 95%-confidence interval)

Method	Energy consumption (J)	ARI
Agglomerative Clustering	2.22418 ± 0.45608	0
Ward	0.20775 ± 0.14911	0.10273
DBSCAN	0.65503 ± 0.03421	0.65630
HDBSCAN	0.40005 ± 0.16425	0.09869
OPTICS	$\mathbf{17.8558 \pm 3.07832}$	0.11329
GMM	0.66935 ± 0.23608	0.58771
K-Means	0.51553 ± 0.10470	0.66383
K-Means++	0.08279 ± 0.01539	**0.66279**

TABLE II: Energy consumption of change detection methods (average and 95%-confidence interval)

Method	Energy consumption (J)	AUC
G-GLRT5	1634.99 ± 59.0515	0.82532
G-GLRT7	1650.17 ± 57.1521	0.85262
G-GLRT21	2029.99 ± 57.1596	0.92298
NG-GLRT5	11550.0 ± 3492.78	0.84629
NG-GLRT7	13592.6 ± 2038.46	0.88167
NG-GLRT21	$\mathbf{166668 \pm 3032.21}$	**0.94352**
LogDiff	8.32059 ± 0.04058	0.59958

OPTICS in particular stands out as the least frugal method on this data for its energy consumption is also significantly higher than the other methods, as shown on Tab.I. It also appears that s_{WS} and s_{HM} show a different behaviour of the Agglomerative Clustering method the more the energy consumption is taken into account in the score. For s_{WS}, even though its performance is low, this method still shows a lower energy consumption than OPTICS and is then associated to a higher frugality score. Though, for s_{HM}, the performance is a constant factor, and its information is kept for any κ. This can be interpreted as a lack of relevance to call frugal a method giving the worst performance possible (here 0 for the normalized ARI).

2) Change detection: Again, on the change detection task, all three frugality scores agree on the ranking of the change detection methods, identifing the G-GLRT method as the most frugal method, as shown in Fig.5. The NG-GLRT method also shows performances equivalent to G-GLRT when, on average, the LogDiff show poor change detection results on the data, with values of AUC around 0.6. It also appears that the window size used for the G-GLRT and the NG-GLRT method significantly alters their energy consumption. For a window size x, the results of the methods G-GLRT and NG-GLRT with this parameter are shown under the names G-GLRTx and NG-GLRTx respectively in Fig.5. Smaller window sizes induce higher performances for the G-GLRT method, whereas the opposite effect is observed for NG-GLRT. As the energy is taken into account in the calculation of both s_{WS} and s_{HM}, these scores both decrease only for the NG-GLRT21 method. This effect is due to the normalization step of α_n and β_n for the calculation of both scores. Though, we observe a difference in their evolution trends along ϵ and κ. The score s_{WS} shows a linear evolution along ϵ, while s_{HM} is a rational function on κ. Its variations show a more complex impact of the energy consumption on the score. For instance, the evolution of s_{WS} for NG-GLRT21 only shows its energy consumption is far superior to the one of the other methods. The evolution of its s_{HM} shows a steepness for $\kappa \in [0, 0.5]$ creating a clearest separation between methods. Thus, for a given κ, s_{HM} tends to more discriminate methods than s_{WS}. On the other hand, both these scores rely on normalized measurements, which highly depend on the carried out experiments. s_F allows a straightforward comparison as it uses the direct energy consumption measure. As shown on Fig.5, this score does not discriminate the frugality of the change detection methods as evidently as s_{WS} s_{HM}, or as when it is applied to the clustering task. This effect is due to the high absolute energy consumption of these methods, which can be verified on Tab.II. Therefore this score is not adapted to particularly high consumptions.

For both processing tasks, the three scoring methods highlighted different aspects of the frugality measure. The s_{WS} appeared as less informative than the other two, but its calculation makes it an easy-to-read scoring method. The s_{WS} is a straightforward approach easy to interpret, and the s_{HM} and the s_F respectively identify with greater precision the least and the most frugal methods, but the s_F show poor relevance for especially high-energy-consuming methods. Their use as a bundle of indices, as it is often practiced to appreciate the performance in machine learning, is relevant to study the frugality of remote sensing data processing methods.

V. Conclusions

This study highlighted the difficulty of identifying the fugal aspect of a data processing method in remote sensing. A method to collect energy consumption measures and combine them with the performance of machine learning methods was proposed through the use of three frugality scoring approaches. These three representations provided different types of information in the study of the frugality of a method when used for remote sensing applications. The s_{WS} is a straightforward approach easy to interpret, but the s_{HM} discriminates the methods more according to their energy consumption while guaranteeing the method does not have the worst performance possible. The s_F is sensitive to high energy consumption values, but can identify better methods more than the other two scores, and can be scalable to further experimentations with no pretreatment of the energy consumption measurements. Using multiples scoring methods to appreciate the frugality of a data processing methods then appears to be the most relevant approach to encompass the different aspects of frugality in remote sensing. In future work, other scoring methods based on fuzzy logic approaches will be studied, and applications on highly consuming machine learning methods such as deep learning models will be covered.

References

[1] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geoscience Frontiers*, vol. 7, no. 1, pp. 3–10, 2016, special Issue: Progress of Machine Learning in Geosciences. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1674987115000821

[2] V. Bolón-Canedo, L. Morán-Fernández, B. Cancela, and A. Alonso-Betanzos, "A review of green artificial intelligence: Towards a more sustainable future," *Neurocomputing*, vol. 599, p. 128096, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231224008671

[3] W. Vanderbauwhede, "Frugal computing – on the need for low-carbon and sustainable computing and the path towards zero-carbon computing," 2023. [Online]. Available: https://arxiv.org/abs/2303.06642

[4] T. Yuki and S. Rajopadhye, "Folklore confirmed: Compiling for speed compiling for energy," in *International Workshop on Languages and Compilers for Parallel Computing*. Springer, 2013, pp. 169–184.

[5] M. Evchenko, J. Vanschoren, H. H. Hoos, M. Schoenauer, and M. Sebag, "Frugal machine learning," 2021. [Online]. Available: https://arxiv.org/abs/2111.03731

[6] S. Abdulsalam, Z. Zong, Q. Gu, and M. Qiu, "Using the greenup, powerup, and speedup metrics to evaluate software energy efficiency," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*. IEEE, 2015, pp. 1–8.

[7] F. A. Mala and R. Ali, "The big-o of mathematics and computer science," *Appl. Math. Comput*, vol. 6, no. 1, pp. 1–3, 2022.

[8] J. K. Nurminen, "Using software complexity measures to analyze algorithms—an experiment with the shortest-paths algorithms," *Computers & Operations Research*, vol. 30, no. 8, pp. 1121–1134, 2003.

[9] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," 2020. [Online]. Available: https://arxiv.org/abs/2007.03051

[10] B. Courty, V. Schmidt, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, SabAmine, inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, A. Saboni, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, alencon, M. Stęchły, C. Bauer, Lucas-Otavio, JPW, and MinervaBooks, "mlco2/codecarbon: v2.4.1," May 2024. [Online]. Available: https://doi.org/10.5281/zenodo.11171501

[11] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," 2022. [Online]. Available: https://arxiv.org/abs/2002.05651

[12] S. Abdulrahman and P. Brazdil, "Measures for combining accuracy and time for meta-learning," *CEUR Workshop Proceedings*, vol. 1201, pp. 49–50, 01 2014.

[13] S. Morasca, "On the use of weighted sums in the definition of measures," in *Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics*, ser. WETSoM '10. New York, NY, USA: Association for Computing Machinery, May 2010, pp. 8–15. [Online]. Available: https://doi.org/10.1145/1809223.1809225

[14] M. Jovanović, N. Afgan, P. Radovanović, and V. Stevanović, "Sustainable development of the belgrade energy system," *Energy*, vol. 34, no. 5, pp. 532–539, 2009, 4th Dubrovnik Conference. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544208000431

[15] N. H. Afgan, P. A. Pilavachi, and M. G. Carvalho, "Multi-criteria evaluation of natural gas resources," *Energy Policy*, vol. 35, no. 1, pp. 704–713, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421506000371

[16] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2018.

[17] J. R. Eastman, "Multi-criteria evaluation and gis," *Geographical information systems*, vol. 1, no. 1, pp. 493–502, 1999.

[18] B. Wirsam, A. Hahn, E. O. Uthus, and C. Leitzmann, "Fuzzy sets and fuzzy decision making in nutrition," *European Journal of Clinical Nutrition*, vol. 51, no. 5, pp. 286–296, May 1997, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/1600378

[19] C. J. Van Rijsbergen, "Foundation of evaluation," *Journal of documentation*, vol. 30, no. 4, pp. 365–373, 1974.

[20] A. Mian, G. Ginolhac, J.-P. Ovarlez, and A. M. Atto, "New Robust Statistics for Change Detection in Time Series of Multivariate SAR Images," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 520–534, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8552453/

[21] A. D. C. Nascimento, A. C. Frery, and R. J. Cintra, "Detecting changes in fully polarimetric sar imagery with statistical information theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1380–1392, 2019.

[22] K. Conradsen, A. A. Nielsen, and H. Skriver, "Determining the points of change in time series of polarimetric sar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 3007–3024, 2016.