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*—In remote sensing, the demand for extensive data processing algorithms continues to grow exponentially. Assessing the frugality of data processing algorithms has become a priority in the machine learning and artificial intelligence community. However, defining a quantifiable measure that combines performance and energy consumption remains a complex challenge. This paper introduces a methodology for collecting energy consumption data and presents three different frugality scoring methods. Through a case study of two classical data processing tasks in remote sensing, change detection and clustering, we demonstrate that these three scores capture different aspects of frugality. We further suggest a combined approach for users to achieve a more comprehensive assessment.**

***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, further research is essential for both the evaluation and the improvement of their frugality [2] for remote sensing practitioners. In this paper, we define frugality as the pursuit of low energy consumption while ensuring satisfactory performance of the chosen method.

Measuring frugality can be challenging, and the ecological or energetic cost of a method is often estimated by the runtime [4], [5]. It has been shown that runtime does not capture all the information about energy consumption [6]. Other metrics such as algorithm complexity or the number of lines of code can also be used [7], [8]. However, they only provide an estimate of a theoretical running cost, which is difficult to translate into a carbon footprint. Thus, several energy consumption measurement tools 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 using an external connected device. If the measurement frequency is sufficiently high to capture precise energy consumption data during runtime, this method provides an accurate assessment of the frugality of the studied approach.

Evaluating frugality involves combining the estimated energy consumption of the method with its performance. Several studies focus on combining the performance of a method with its running time [5], [12], but the literature lacks studies that propose scoring methods based on empirical energy consumption. In this regard, many different approaches exist to create a multi-criteria index. Although the use of weighted sums to aggregate metrics has been criticized [13], it remains a commonly used multi-criteria scoring method [14]–[16]. Fuzzy-logic-based approaches have also gained popularity for this type of task [17], [18]. However, they require expert knowledge to define the decision boundaries and may not scale well across different data processing methods.

In this paper, we address, for the first time, the issue of estimating frugality in remote sensing. We propose:

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

II. Measuring Energy Consumption

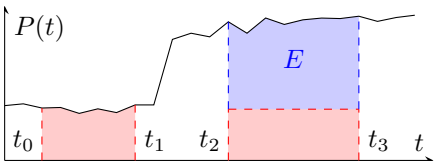


Fig. 1: Example of energy consumption measure E over data processing method runtime.

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 algorithm running, 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:

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

The standard period allows us to exclude 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 method 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 method 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:

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

Another method combining these metrics is the harmonic mean, similarly to 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] and is therefore ideal 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:

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

Since these two scores require the normalization of both the performance of the method and the energy consumption metrics, we propose a frugality score s_F inspired by [5]. This score is based on both metrics α and β, with a weight $w \in [0, 1]$ assigned to the available resource—here the energy consumption—in contrast to the method's performance. This score is shown in eq.4 and the dependence on the weight w is unique for each method and input data used:

$$s_F = \alpha - \frac{w}{1 + \frac{1}{\beta}}. \tag{4}$$

IV. Results

In this section, we will present the two case studies to which these frugality scores will be applied as well as the results obtained.

A. Task descriptions

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

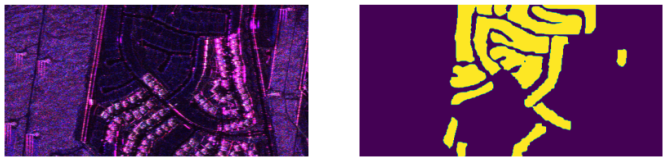
The clustering task is an unsupervised approach. Because of its complexity, the literature lacks benchmark clustering datasets based on remote sensing data. Therefore, in this paper, classical clustering approaches were applied to a toy dataset presenting 5 blobs of data along 5 features, adapted to 5-group clustering algorithms. 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++** methods. The parameters of each method were chosen to maximize the adjusted Rand index (ARI) calculated between the clustering result and the initial known clusters defined by the data blobs. Indeed, the ARI presented in eq.5 standardizes the Rand index (RI) by the expected RI guaranteeing that a random clustering returns a zero value:

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

n the number of instances,

a the accurate number of pairs in the same cluster,

b the accurate number of pairs in different clusters.



(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 scores on the change detection task, three algorithms were applied to remote sensing data. This data consists of 2 PolSAR images of 3 bands from the UAVSAR database (Courtesy NASA/JPL-Caltech) [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 (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 bands per image, T multi-band images, N 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 $x_k^{(t)}$ as 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_1 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 based on the sample 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 test under the assumption of Gaussian distribution of the pixel values, and therefore 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:

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

Σ_t a covariance matrix, parameter in θ_t.

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 test statistic was proposed by [22] with a pairwise approach between successive images along the time series to detect changing points.

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

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

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 change detection method, its test statistic involves the texture using a fixed-point estimator $\hat{\Sigma}_t^{\mathrm{TE}}$ (the Tyler Estimator), which is computationally intensive. Both the G-GLRT and NG-GLRT methods were applied using three window sizes (5, 7 and 21 pixels) to calculate covariance matrices within the images. Using these 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 is the log difference—the difference in the log value of the interband mean for each pixel between two images. By definition, this method applies only to image pairs. 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 methods 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, which were collected following the experiment setup presented in the next subsection.

B. Experiment setup

The experiments were conducted using implementations that relied exclusively on CPU, avoiding GPU usage, which would have resulted in higher energy consumption. Thus, the implementations used an Intel i5-12600 3.30GHz CPU, a 2 $\times$ 32 Go RAM, and a Smart Switch 7 Aeotec® with a USB Z-Stick 7 Aeotec® controller. The code for all experiments is available on GitHub[1].

For statistical significance of the energy consumption measure, each run is repeated 30 times. For each run, the energy consumption data, along with the method's performance and its parameters, are stored. The results obtained are then analysed in the next subsection.

C. Results obtained

During the running time of each method, their performance and energy consumption were measured, and each frugality score was calculated for comparison.

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

Overall, the three frugality scores presented in Fig.3 agree with the ranking of the methods based 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 Agglomerative Clustering methods show poorer clustering performance according to their measured ARI. OPTICS in particular stands out as the least frugal method for these data as its energy consumption is significantly higher than the other methods, as shown in TABLE I. It also appears that s_{WS} and s_{HM} exhibit different behaviours

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

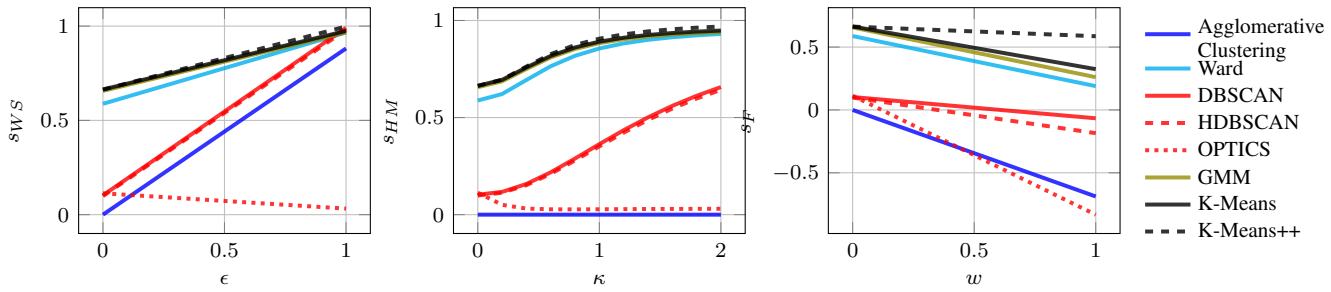


Fig. 3: Mean frugality scores on the clustering task.

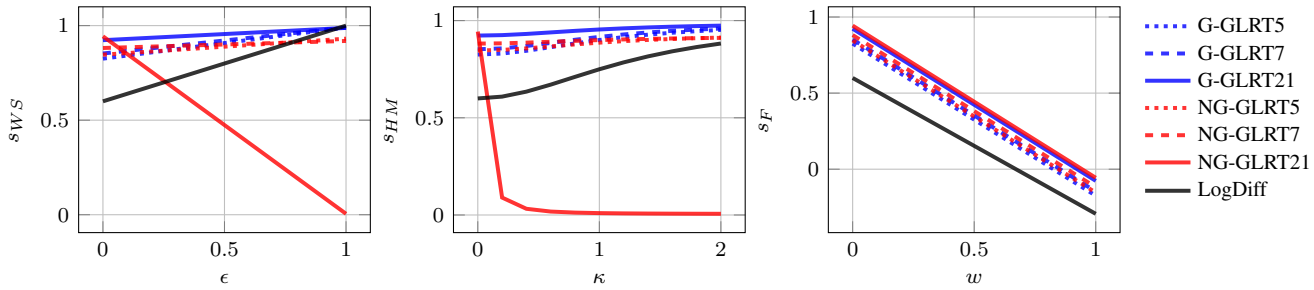


Fig. 4: Mean frugality scores on the change detection task.

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**

for the Agglomerative Clustering method depending on how much energy consumption is factored into the score. For s_{WS}, even though its performance is low, this method still shows lower energy consumption than OPTICS, which results in a higher frugality score. However, for s_{HM}, performance remains a constant factor, and its information is kept for any κ. This can be interpreted as a lack of relevance in labelling a method as frugal when it delivers the worst performance possible (in this case, 0 for the normalized ARI).

TABLE II: Energy consumption of change detection methods (average and 95%-confidence interval). For G-GLRTx and NG-GLRTx correspond to a window size x.

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

2) Change detection: Again, on the change detection task, all three frugality scores are consistent with the ranking of the change detection methods, identifying the G-GLRT method as the most frugal method, as shown in Fig.4. The NG-GLRT method also shows performance equivalent to G-GLRT, while on average, the LogDiff yields poor change detection results on the data, with AUC values around 0.6. It also appears that the window size used for the G-GLRT and the NG-GLRT methods significantly affects their energy consumption. Smaller window sizes induce higher performance for the G-GLRT method, while the opposite effect is observed for NG-GLRT. As energy is taken into account in the calculation of both s_{WS} and s_{HM}, these scores decrease only for the NG-GLRT21 method. This effect is due to the normalization step of α_n and β_n in the calculation of both scores. However, 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 of 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 indicates that its energy consumption is significantly higher than that of the other methods. The evolution of its s_{HM} shows a steep slope for $\kappa \in [0, 0.5]$ creating a clearer separation between methods. Thus, for a given κ, s_{HM} tends to discriminate methods more effectively than s_{WS}. However, both scores rely on normalized measurements, which is highly dependent on the specific experiments conducted. s_F allows for a straightforward comparison as it directly uses the energy consumption measure. As shown in Fig.4, this score does not discriminate the frugality of the change detection methods as effectively as s_{WS} s_{HM}, nor as it does when applied to

the clustering task. This effect is due to the high absolute energy consumption of these methods, as shown in TABLE II. Therefore, this score is not well-suited for cases with particularly high energy consumption.

For both tasks, the three scoring methods highlighted different aspects of the frugality measure. s_{WS} appeared less informative than the other two, but its simplicity makes it an easy-to-read scoring method, while s_{HM} and s_F, respectively, identify with greater precision the least and the most frugal methods. However, s_F shows poor relevance for methods with especially high energy consumption. Using these scores together, as it is often done to assess 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 frugality aspect of a data processing method in remote sensing. A method was proposed to collect energy consumption data and to combine it with the performance of machine learning methods 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. s_{WS} is a straightforward and easy-to-interpret approach, but s_{HM} discriminates more efficiently methods based on their energy consumption while ensuring that the method does not have the worst possible performance. s_F is sensitive to high-energy-consumption measurements, but can identify the most frugal methods more efficiently than the other two scores, and it is scalable to further experiments without pretreatment of the energy consumption measurements. Using multiple scoring methods to assess the frugality of data processing methods thus 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 to highly energy-consuming machine learning methods such as deep learning models will be addressed.

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.