Sustainability of AI-Enabled Systems: Balancing Accuracy and Energy Trade-off

Rafiullah Omar
Engineering and Computer Science and Mathematics
FrAmeLab, University of L'Aquila
L'Aquila, Italy
rafiullah.omar@graduate.univaq.it

Henry Muccini
Engineering and Computer Science and Mathematics
FrAmeLab, University of L'Aquila
L'Aquila, Italy
henry.muccini@univaq.it

***Abstract*—The integration of Machine Learning (ML) models into software systems has raised significant concerns about energy consumption. This study investigates the energy efficiency of ML systems, focusing on data-centric approaches, specifically feature selection and concept drift management. Our analysis reveals substantial differences in energy consumption across feature selection methods and explores their impact on the energy footprint of ML models. Additionally, we examine how various concept drift detectors offer differing accuracy and energy costs, identifying energy-efficient strategies to maintain ML performance. Empirical results highlight different energy and accuracy of different feature reduction methods and concept drift detectors. These findings contribute to the development of sustainable, energy-efficient ML models, offering valuable insights for software engineering to enhance service efficiency, reduce operational costs, and minimize environmental impact.**

***Index Terms*—Software Sustainability, Green AI, Energy consumption, scoring methods, environmental impact, concept drift**

I. Introduction

The integration of Machine Learning (ML) into software systems has sparked concerns regarding energy consumption [1], [2]. Efforts to enhance the energy efficiency of AI systems have explored various methods, with the data-centric approach emerging as a promising strategy [3]. Unlike the model-centric approach, which primarily focuses on optimizing the model itself, the data-centric approach emphasizes improving energy efficiency through more efficient data utilization.

Building upon existing research [3], this study aims to delve deeper into different phases of the ML development life cycle. Our primary objective is to reduce the number of features employed in model training while either maintaining accuracy or incurring minimal accuracy loss. Feature reduction plays a pivotal role in decreasing energy consumption associated with ML models [4]. Various techniques exist for feature reduction, including eliminating features with low variance, model-based selection, mutual information, chi-square, and ANOVA F-test (if-classif). However, determining the most energy-efficient method that also enhances model accuracy remains a critical research question.

Concept drift, where the data distribution evolves over time, poses another significant challenge to model sustainability and performance accuracy. Concept drift can manifest gradually or abruptly, necessitating effective detection mechanisms to adapt ML models accordingly. In this study, we evaluate different concept drift detectors in terms of their energy consumption implications and their impact on model accuracy. This analysis aims to elucidate the trade-offs involved in detecting various types of concept drifts and to identify strategies that optimize both energy use and model performance.

II. Study Design

For the feature reduction part, we opted for various feature reduction methods, implemented in the scikit-learn [1] ML library. The selected feature reduction methods include all available feature selection methods for classification tasks in scikit-learn: removing features with low variance, SelectFromModel, mutual information, chi-square, ANOVA F-test (if-classif), and recursive feature elimination. For our experiments, we used the spam-ham dataset and varied the percentage of features, starting from 10% and increasing in 10% increments up to 90%. Six distinct ML models were chosen for the classification tasks: SVM, KNN, Bagging Classifier, Adaboost, Decision Tree, and Random Forest. Each model was trained on datasets with different percentages of features, for consistency, the experiments were repeated 30 times. In addition to the classification tasks, we explored another set of feature reduction methods designed for regression tasks, namely r_regression, recursive feature elimination, and mutual_info_regression. Correspondingly, we selected regression models for these tasks, including Random Forest Regression, KNN Regression, Gradient Boosting Regression, Gaussian Process Regression, and Decision Tree Regression. Similar to the classification tasks, we repeated the experiments 30 times. These experiments aimed to understand the energy efficiency of different feature reduction methods and their effect on model training and accuracy.

For the concept drift part, we selected seven drift detection methods implemented in the scikit-learn library, six models also implemented in scikit-learn, and five types of synthetic datasets, each featuring both abrupt and gradual drifts.

Selected Drift Detection Methods: DDM, EDDM, HDDM_A, HDDM_W, ADWIN, KSWIN, and PageHinkley.

[1] url:https://scikit-learn.org/

Selected Models: KNN, Adaboost, Bagging Classifier, Random Forest, SVM, and Decision Tree.

Selected Datasets: Sine, Mixed, Sea, RT, and Stagger.

By investigating different drift detection methods in terms of energy consumption and accuracy, we aim to analyze the trade-offs in detecting various types of concept drifts. This comprehensive analysis will contribute to our understanding of how different feature reduction and drift detection methods impact the energy efficiency and accuracy of ML models. To measure energy consumption, we executed a warm-up function initially and set a 5-second sleep interval between each experiment. The software tool utilized for measuring energy consumption was **CodeCarbon**[2]. The code was run on dedicated experimental infrastructure specifically designed for software energy experiments. This infrastructure consists of a server equipped with a 36 TB HDD, 384 GB RAM, and an Intel Xeon CPU featuring 16 cores with hyper-threading, operating at 2.1 GHz (yielding 32 virtual CPUs).

III. Results

The results of our experiments on feature reduction methods indicate significant differences in energy consumption among the methods. The f-classif method is one of the most energy-efficient, followed by chi-square and Variance Threshold. SelectFromModel and Recursive Feature Elimination (RFE) perform poorly in terms of energy consumption, with RFE being the least efficient. Quantitatively, f-classif consumes 99.99% less energy compared to RFE. Regarding accuracy, the maximum difference between the accuracies of the different methods is 0.014.

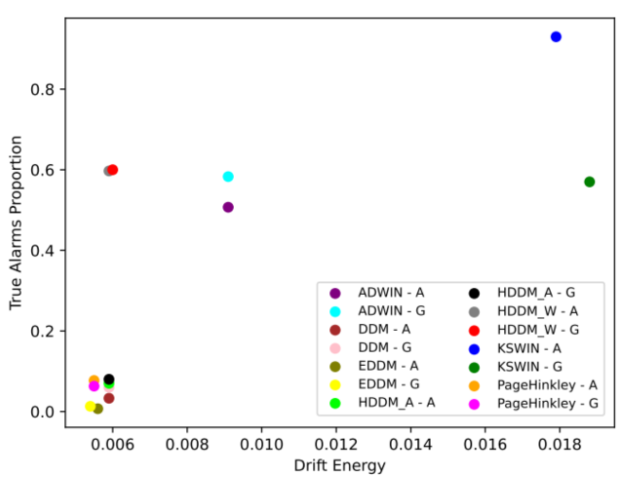

Fig. 1. True Alarm Rate vs Energy Consumption (G: Gradual, A: Abrupt)

The experimental results for the comparison of different drift detectors are summarized in Fig. 2 and Fig. 1. We categorize the drift detectors into three types: a) detectors that prioritize detection accuracy at the expense of energy efficiency (e.g., KSWIN), b) balanced detectors that consume low to medium energy while maintaining good accuracy (e.g., HDDM_W, ADWIN), and c) detectors that consume very little energy but are impractical due to poor accuracy (e.g., HDDM_A, PageHinkley, DDM, EDDM). From a Green AI perspective, the most compelling option for energy-efficient drift detection is HDDM_W, which balances high accuracy with low energy consumption. It performs exceptionally well in gradual drift scenarios, combining precise drift position detection with good true alarm rates.

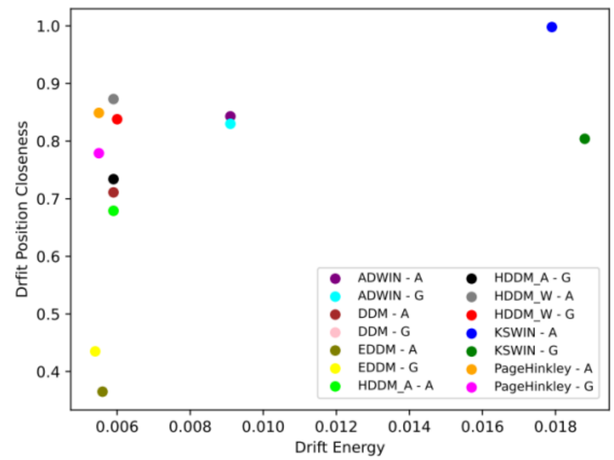

Fig. 2. Drift Position closeness vs Energy Consumption (G: Gradual, A: Abrupt)

IV. Conclusion

Our research investigated the trade-offs between energy consumption and accuracy of feature selection methods using a data-centric approach. We found that the f-classif is the most energy-efficient, consuming 99.99% less energy than RFE with minimal accuracy loss. Among drift detectors, HDDM_W stood out for balancing high accuracy and low energy consumption, especially in gradual drift scenarios. These findings contribute to Green AI by providing guidelines for reducing the environmental impact of AI systems while maintaining performance. Future work can expand on these insights to further enhance the sustainability of machine learning models.

References

[1] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[2] R. Omar, "Ai and energy efficiency," in *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2023, pp. 141–144.

[3] S. Salehi and A. Schmeink, "Data-centric green artificial intelligence: A survey," *IEEE Transactions on Artificial Intelligence*, 2023.

[4] R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier, "Data-centric green ai an exploratory empirical study," in *2022 international conference on ICT for sustainability (ICT4S)*. IEEE, 2022, pp. 35–45.

[2]https://codecarbon.io/