The Energy-Accuracy Trade-off of Hyperparameter Tuning

Abdulrahman Alshahrani
Department of Computer Science
The University of Texas at Austin
abdulrahmanalshahrani@utexas.edu

Abdallah Al-Sukhni
Department of Computer Science
The University of Texas at Austin
doogls@utexas.edu

***Abstract*—This paper explores the energy-accuracy trade-off in machine learning (ML) model development, particularly focusing on the effects of hyperparameter tuning in grid search with cross-validation testing. As the demand for high-performing ML models increases across various sectors, this study underscores the significant impact of hyperparameter tuning on energy consumption and environmental sustainability. By employing a diverse set of datasets and ML algorithms—including Decision Trees, K-Nearest Neighbors (KNN), Multi-Layer Perceptrons (MLPs), and Random Forests—our research evaluates the correlation between the breadth of hyperparameter tuning and both model accuracy and energy consumption. The findings reveal that extensive hyperparameter search frequently results in minimal accuracy gains while substantially increasing energy consumption. This pattern is consistent across different algorithms and datasets, indicating a potential for substantial energy savings with carefully considered hyperparameter selection strategies. Through this experiment, we aim to build the path towards more energy-efficient ML practices, balancing the pursuit of model accuracy with the crucial goal of saving the environment.**

***Index Terms*—Machine Learning, Energy Efficiency, Hyperparameter, Grid Search**

I. Introduction

The rise of machine learning in diverse applications has been a defining feature of the technological landscape in recent years. As industries and sectors ranging from healthcare and finance to entertainment and transportation integrate ML into their operations, more and more machine learning applications are being developed to accommodate the increased demand. This widespread adoption is driven by several critical factors: the rapid growth in data availability, continuous enhancements in ML algorithms, and significant strides in computational power [1]. For example, in the healthcare industry, the application of ML models has revolutionized diagnostic procedures, and patient monitoring, marking a significant leap toward precision medicine [2]. The [3] paper explains how ML can be effectively developed for recognizing strokes due to its capability to track a multitude of variables and patterns.

The majority of machine learning development focuses on increasing the model's accuracy, prioritizing performance gains despite their increased energy consumption [4]. Modern machine learning models require significant energy to train them, where for example, training a large Transformer model with neural architecture search emits 17x the amount of carbon compared to an average American in one year [5]. Additionally, data centers are estimated to consume anywhere between 8-21% of electricity worldwide by 2025, in which training a large deep-learning model can produce 284000 kgs of CO2 [4].

Hyperparameters are a set of configurations that specify the structure of the machine learning model [11]. They are manually set by the developer, and they are set before training the model [11]. However, changing these configurations affects the model's accuracy dramatically. The process of choosing the optimal hyperparameters for the current dataset is called hyperparameter tuning [11]. Hyperparameter tuning contributes significantly to energy consumption during the Model's training phase. [5] reported that the process of hyperparameter optimization can consume a substantial portion of the total energy expenditure in training machine learning models, where training a big Transformer initiated 120+ small hyperparameter grid searches, which took about 240k hours and an estimated cloud computing cost of $103k-350$k.

Measuring the accuracy of machine learning models on a continuous scale allows us to adjust the thoroughness of the model training phase, potentially discovering areas where we can achieve similar model accuracy with reduced energy consumption [4]. Given these points, this paper aims to explore the trade-off between the increased accuracy of more exhaustive hyperparameter grid searches and their increased energy consumption, across several datasets, several machines, and multiple ML algorithms. We aim to investigate if the increase in accuracy model justifies the rise in energy consumption.

II. Related Works

[4] conducted a study where they explored the trade-off between energy efficiency and model accuracy that primarily focused on the hyperparameter tuning of multilayer perceptrons. The paper began investigating the energy and accuracy of using grid search on multiple hyperparameter options across several datasets. Their findings revealed a clear opportunity to trade off small accuracy in exchange for a much higher reduction in energy consumption. This was observed in the majority of the datasets used (four out of five). Their paper then goes on to investigate which parameters contribute more towards energy consumption, which is outside the scope of our paper. While [4] primarily focused on hyperparameter tuning of MLPs, our study extends this research by examining the

trade-off between energy efficiency and model accuracy of not only MLPs but also other widely used algorithms, including k-nearest-neighbors (KNN), decision trees, and random forest.

[5] argued that the NLP community should adopt more transparent reporting practices regarding the training time and sensitivity to hyperparameters of models. This is proposed as a means to facilitate a more informed decision-making process regarding the trade-offs between computational costs and the performance benefits of different models. While [5] discussed plans for the NLP community, we find the goal applicable to other areas of ML, and we plan to expand on it by reporting the time and energy consumption of ML algorithms outside the scope of that paper.

III. Methodology

A. Dataset Selection

As seen in Table I, we selected four diverse datasets: Olivetti Faces [6], Census Income [7], RT-IoT2022 [8], and CDC Diabetes Health Indicators [9]. The Olivetti Faces dataset consists of images of people making different expressions. Each record represents an image, and therefore, the features are pixel values. The Census Income dataset consists of information about different individuals. RT-IoT2022 consists of data collected from various sensors in real-time Internet of Things environments. The CDC Diabetes Health Indicators consist of data regarding health indicators pertaining to diabetes. These datasets were chosen to represent a variety of data types and complexities commonly encountered in real-world applications.

TABLE I
DATASETS WE USED IN OUR EXPERIMENT

Dataset	# of Instances	# of Features
Olivetti Faces	400	4096
Census Income	48842	14
RT-IoT2022	123117	83
CDC Diabetes Health Indicators	253680	21

B. Model Selection and Implementation

We employed four distinct machine learning algorithms for our study: decision trees, k-nearest neighbors (KNN), multilayer perceptrons (MLPs), and random forests. These algorithms were chosen based on their popularity, versatility, and applicability to a wide range of classification tasks.

For the implementation of the algorithms, we used the `scikit-learn` library, a widely-used machine learning library for Python [10]. For hyperparameter tuning and model selection, we employed the `GridSearchCV` function, which performs an exhaustive search over a specified parameter grid to find the optimal combination of hyperparameters.

To construct our classification models, we utilized algorithms from `scikit-learn`'s collection of classifiers. This included `DecisionTreeClassifier`, a supervised learning method that constructs a decision tree based on the input features; `KNeighborsClassifier`, an instance-based learning algorithm that classifies new instances based on their similarity to the nearest neighbors in the training data; `MLPClassifier`, an implementation of a multi-layer perceptron; and `RandomForestClassifier`, an ensemble learning method that combines multiple decision trees to improve predictive accuracy and control overfitting.

C. Hyperparameter Configurations

The hyperparameter selection process for our study was informed by a manual exploration of potential value ranges and their effects on the performance of each algorithm. Although systematic grid search techniques were employed to fine-tune these parameters, we acknowledge that alternate configurations might yield different results. Our choice of hyperparameters was guided by the assumption that they hold significant sway over the predictive outcomes. This targeted approach allowed us to discern which hyperparameters are most impactful, even though there exists the possibility that other hyperparameter combinations could further optimize model performance. Table II, III, IV, and V show the hyperparameter and value ranges we chose in our experiment. Please note that in the Grid cells, the notation "x-y (step r)" means that the tested values ranged from x (inclusive) to y (exclusive) in steps of r.

TABLE II
HYPERPARAMETER VALUES TESTED AT EACH GRID SEARCH FOR DECISION TREES

Hyperparameter	Grid Search Number			
	1	2	3	4
max_depth[a]	1-31 (step 12)	1-31 (step 6)	1-31 (step 3)	1-31 (step 3)
min_samples_split[b]	-	2-17 (step 4)	2-17 (step 2)	2-17 (step 2)
min_samples_leaf[c]	-	-	-	1-17 (step 2)

[a]Maximum depth of the tree. [b]Minimum samples required to split a node. [c]Minimum samples required at a leaf node.

TABLE III
HYPERPARAMETER VALUES TESTED AT EACH GRID SEARCH FOR K-NEAREST NEIGHBORS

Hyperparameter	Grid Search Number			
	1	2	3	4
n_neighbors[a]	5-20 (step 6)	5-20 (step 6)	5-20 (step 3)	5-20 (step 3)
weights[b]	-	uniform, distance	uniform, distance	uniform, distance
metric[c]	-	-	minkowski, euclidean	minkowski, euclidean
p[d]	-	-	-	1, 2, 3

[a]Number of neighbors. [b]Weight function used in prediction. [c]Distance metric used. [d]Power parameter for the Minkowski metric.

D. Measuring Energy Consumption

To measure energy consumption, we utilized the `CodeCarbon` Python library—an open-source, lightweight

TABLE IV
HYPERPARAMETER VALUES TESTED AT EACH GRID SEARCH FOR NEURAL NETWORKS

Hyperparameter	Grid Search Number			
	1	**2**	**3**	**4**
hidden_layer_sizes[a]	(100), (50,50)	(100), (50,50), (100,100), (200,200)	(100), (50,50), (100,100), (200,200), (300,300)	(100), (50,50), (100,100), (200,200), (300,300)
alpha[b]	-	-	-	0.0001, 0.001
activation[c]	-	-	-	relu, tanh, logistic

[a]Number and size of the neural network layers. [b]L2 penalty parameter. [c]Activation function for the hidden layer.

TABLE V
HYPERPARAMETER VALUES TESTED AT EACH GRID SEARCH FOR RANDOM FORESTS

Hyperparameter	Grid Search Number			
	1	**2**	**3**	**4**
n_estimators[a]	10, 50, 100	10, 50, 100, 200	10, 50, 100, 200	10, 50, 100, 200, 300
max_depth[b]	-	None, 10, 20	None, 10, 20	None, 10, 20
min_samples_split[c]	-	-	2, 5	2, 5
min_samples_leaf[d]	-	-	-	1, 2

[a]Number of trees in the forest. [b]Maximum depth of the tree. [c]Minimum samples required to split a node. [d]Minimum samples required at a leaf node.

tool—to measure and estimate the carbon footprint of computational devices [12]. This tool utilized NVIDIA's Management Library (NVML) and Intel Power Gadget to access data on power usage by GPUs and CPUs, respectively. To obtain timely and regular data, we configured `CodeCarbon` to poll the system at five-second intervals.

One of the challenges with this approach is that `CodeCarbon` bases its energy consumption estimates on the overall CPU and GPU usage, which means it captures the energy usage of the entire computer system, not just the specific processes related to model training. This broad measurement could potentially include energy used by background tasks that are not related to our computations.

To counter this, we proactively closed all non-essential background applications to ensure that our energy measurements were as focused on the algorithms' consumption as possible. Moreover, we're comparing the relative energy consumption between the different algorithms under the same conditions—same computer, same system state—which serves to normalize the data and reduce the influence of unrelated processes. By conducting all algorithm tests in this standardized environment, we ensured that our findings on energy efficiency were more accurate and relevant.

E. Hardware Setup

The experiments were executed on two distinct computing setups. The first machine comprised an Intel(R) Core(TM) i7-9750H processor with 6 cores (12 threads) clocked at 2.60GHz, paired with an NVIDIA GeForce GTX 1660 Ti graphics processing unit and 32 GB of RAM. The second machine was equipped with an Intel(R) Core(TM) i5-9400 processor, featuring 6 cores at 2.90GHz, alongside the same model of GPU, the NVIDIA GeForce GTX 1660 Ti, but with a reduced memory capacity of 16 GB.

IV. RESULTS

Our study involves Decision Trees, K-Nearest Neighbors (KNN), Multi-Layer Perceptrons (MLPs), and Random Forests, assessed on a variety of datasets. The results presented here represent the average values between the two previously described machines. The following sections detail the energy consumption in kilowatt-hours (kWh) and accuracy for each algorithm.

A. Decision Trees

In Figure 1 and 2, we can see that the decision tree algorithm demonstrated an initial improvement in predictive accuracy from the first grid search iteration to the second. However, subsequent grid searches yielded marginal gains in accuracy, with some iterations even exhibiting a slight decrease in performance. Notably, the improvements in accuracy fell within the range of 0.01 to 0.02%, which can be considered relatively small in practical terms. Conversely, energy consumption increased steadily with each consecutive grid search iteration.

In Figure 2, we can also notice that the accuracy of the model declines at the last, most comprehensive grid search. This observation, while initially counterintuitive, can be attributed to overfitting during the internal cross-validation steps. Specifically, a more exhaustive grid search provides the model with an extensive range of hyperparameters, which, although it allows for a finely tuned fit to the internal validation and test data, can lead to overfitting. This overfitting captures not only the underlying patterns but also the noise present in the training set, leading to high internal accuracy. However, this overfitting reduces the model's ability to generalize, resulting in lower accuracy when the model was applied to an external testing dataset. This explains the drop in performance observed in the figure as the grid search parameters are expanded.

B. K-Nearest Neighbors (KNN)

In Figure 3 and 4, initially, the KNN algorithm showed negligible improvement in accuracy across the early grid search iterations. However, the final and most exhaustive grid search resulted in a substantial increase in accuracy compared to the previous iterations. It is crucial to note that this gain in accuracy came at a significant cost in terms of energy consumption and computational time.

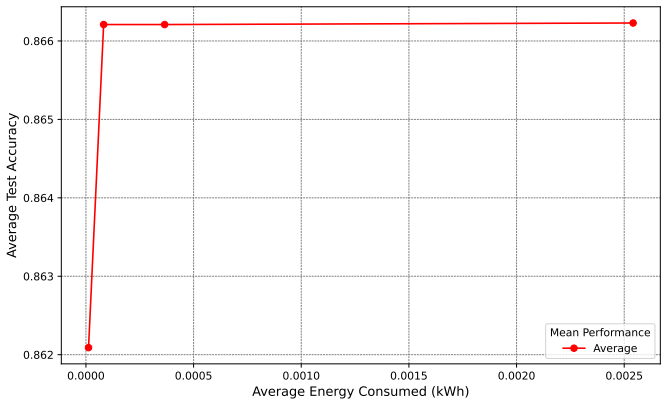

Fig. 1. Decision Tree's Average Energy Consumption vs. Accuracy for each Grid Search in the CDC Health Indicator dataset

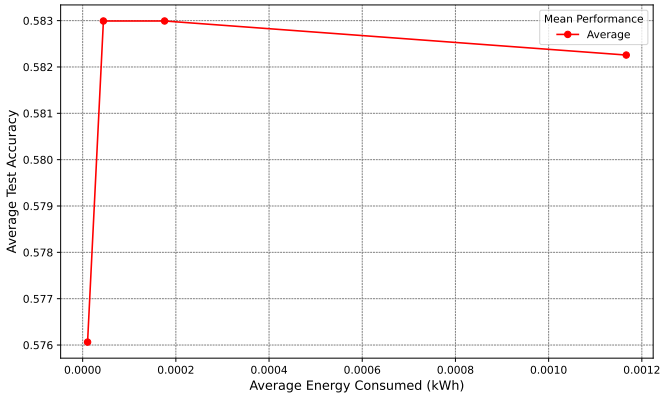

Fig. 2. Decision Tree's Average Energy Consumption vs. Accuracy for each Grid Search in the Census Income dataset

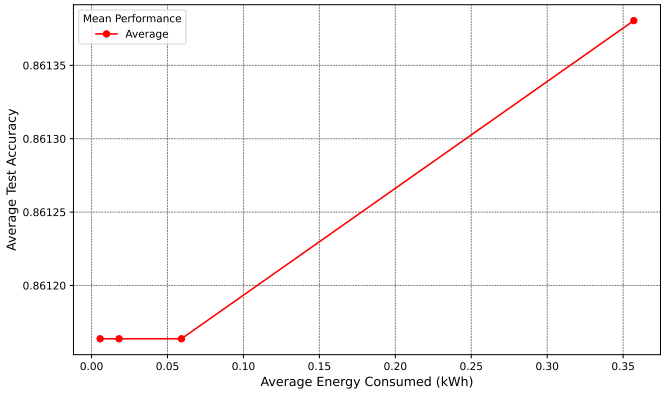

Fig. 3. K-Nearest Neighbors' Average Energy Consumption vs. Accuracy for each Grid Search in the CDC Health Indicator dataset

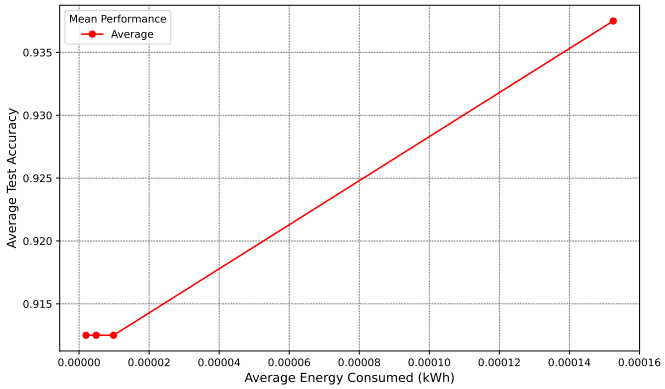

Fig. 4. K-Nearest Neighbors' Average Energy Consumption vs. Accuracy for each Grid Search in the Olivetti Faces dataset

C. Multilayer Perceptrons (MLPs)

For the census dataset in Figure 5, the MLP algorithm exhibited a pattern similar to the decision trees, with an initial jump in accuracy from the first grid search iteration to the second. Subsequent grid searches yielded minimal improvements in accuracy with steady increases in energy consumption. Conversely, for the CDC Health Indicator dataset in Figure 6, the MLP showed no improvement in accuracy across grid search iterations. However, it is worth noting that the initial accuracy for this dataset was already relatively high.

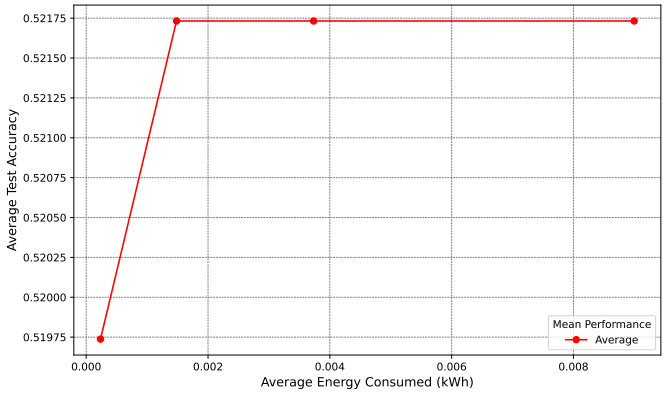

Fig. 5. MultiLayer Perceptrons' Average Energy Consumption vs. Accuracy for each Grid Search in the Census dataset

D. Random Forests

In Figures 7 and 8, the random forest algorithm followed a trend similar to decision trees and MLPs. An initial increase in accuracy was observed between the first and second grid search iterations. However, further grid searches resulted in negligible improvements in accuracy, despite the steady increase in energy consumption associated with more exhaustive hyperparameter tuning.

V. Findings

The comprehensive evaluation of decision trees, k-nearest neighbors (KNN), multilayer perceptrons (MLPs), and random

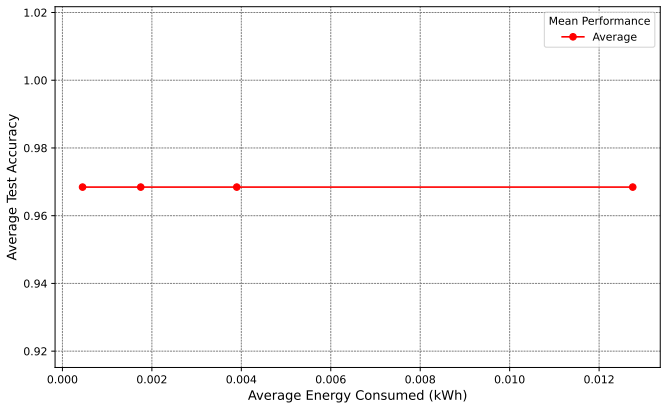

Fig. 6. MultiLayer Perceptrons' Average Energy Consumption vs. Accuracy for each Grid Search in the RT-IoT2022 dataset

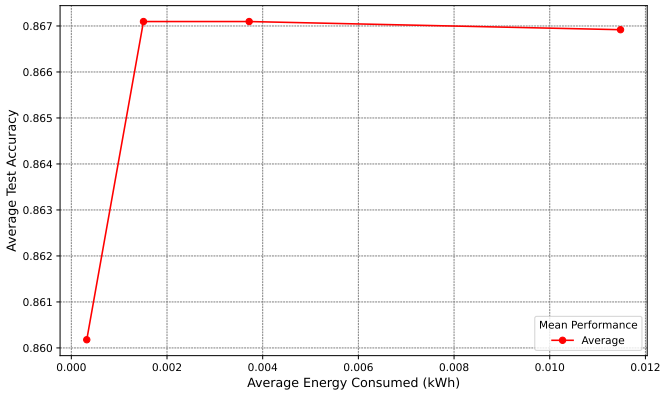

Fig. 7. Random Forest's Average Energy Consumption vs. Accuracy for each Grid Search in the CDC Health Indicator dataset

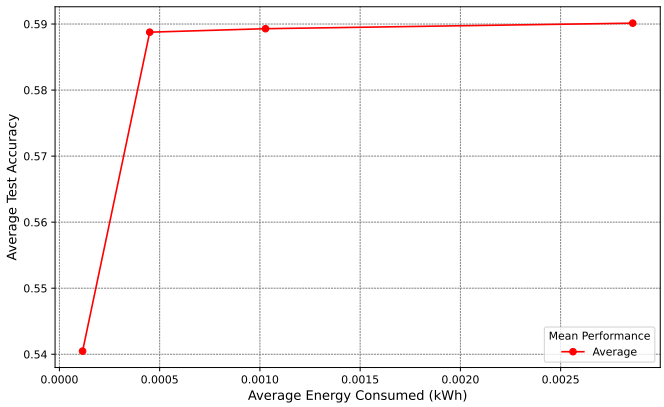

Fig. 8. Random Forest's Average Energy Consumption vs. Accuracy for each Grid Search in the Census Income dataset

forests, revealed notable insights into the relationship between accuracy and energy consumption. Most algorithms exhibited minimal increases in accuracy, despite the considerable increases in energy consumption. On average, each additional grid search iteration led to a mere 0.01% increase in accuracy. On the other hand, the data reveals that testing additional hyperparameters increased energy consumption by an average of 309.64% per additional grid search iteration. This staggering increase in energy consumption associated with a negligible gain in performance highlights the inefficient nature of grid search hyperparameter tuning.

VI. Conclusion

Our research underscores the necessity of a balanced approach to hyperparameter tuning, wherein energy efficiency is considered in tandem with accuracy. We have seen that using grid search resulted in clear diminishing returns in regards to accuracy, but had significant increases in energy consumption. By demonstrating the potential for significant energy reductions without substantially impacting model performance, we aim to advocate for a shift towards more sustainable machine learning methodologies. Future research endeavours should explore the practical application of these findings in real-world settings, further refining the balance between energy consumption and model accuracy.

However, it is essential to recognize that in specific critical contexts where even minute improvements in accuracy are of paramount significance, extensive grid search methods may remain a viable option. The medical domain serves as an example, where the potential ramifications of marginal accuracy gains can highly impact patient outcomes and survival rates. Moreover, in certain research or specialized industrial environments where computational resources and energy budgets are not primary limiting factors, the exhaustive exploration of hyperparameters through grid search remains a valuable tool for model optimization. Specific industries that may prioritize maximizing accuracy over energy efficiency render grid search a suitable approach for achieving optimal model accuracy.

Nonetheless, it is imperative to have a carefully calculated balance of accuracy and energy efficiency. In scenarios where incremental accuracy gains are less critical, and environmental or economic factors take precedence, the findings of this study strongly advocate for the adoption of more energy-efficient hyperparameter tuning strategies, thereby promoting sustainable machine learning practices without compromising model performance to an unacceptable degree.

Acknowledgment

This paper was supported by the use of ChatGPT-4, developed by OpenAI. ChatGPT-4 assisted in research, code generation, and the writing process.

References

[1] R. Boutaba, M. A. Salahuddin, N. Limam, et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," Journal of Internet Services and Applications, vol. 9, no. 16, Jun. 2018. [Online]. Available: https://doi.org/10.1186/s13174-018-0087-2)

[2] M. Badawy, N. Ramadan, and H. A. Hefny, "Healthcare predictive analytics using machine learning and deep learning techniques: a survey," J. Electr. Syst. Inf. Technol., vol. 10, no. 40, Aug. 2023. [Online]. Available: https://doi.org/10.1186/s43067-023-00108-y.
[3] Wang W, Kiik M, Peek N, Curcin V, Marshall IJ, Rudd AG, et al. (2020) A systematic review of machine learning models for predicting outcomes of stroke with structured data. PLoS ONE 15(6): e0234722. https://doi.org/10.1371/journal.pone.0234722
[4] A. Brownlee, J. Adair, S. Haraldsson, and J. Jabbo, "Exploring the Accuracy -Energy Trade-off in Machine Learning." Accessed: Apr. 12, 2024. [Online]. Available: https://dspace.stir.ac.uk/bitstream/1893/32312/1/GI2021_Machine_Learning_Energy.pdf
[5] E. Strubell, A. Ganesh, and A. Mccallum, "Energy and Policy Considerations for Deep Learning in NLP," 2019. Available: https://arxiv.org/pdf/1906.02243.pdf
[6] Olivetti Faces dataset, sklearn.datasets.fetch_olivetti_faces, scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html
[7] Census Income dataset, UCI Machine Learning Repository, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/census+income
[8] CDC Diabetes Health Indicators dataset, UCI Machine Learning Repository, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/CDC+Diabetes+Health+Indicators
[9] RT-IoT2022 dataset, UCI Machine Learning Repository, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/RT-IoT2022
[10] Scikit-learn: Machine Learning in Python, Scikit-learn developers, [Online]. Available: https://scikit-learn.org
[11] S. Alisneaky, et al., "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN." Mathematics in Engineering, vol. 2022, Article ID 8513719, 2022. Accessed: Apr. 24, 2024. [Online]. Available: https://www.hindawi.com/journals/mpe/2022/8513719/
[12] "CodeCarbon," CodeCarbon. Online. Available: https://codecarbon.io/.

APPENDIX

For detailed insights into the methodology and findings presented within this paper, the corresponding codebase and comprehensive results are hosted on GitHub. Interested parties are encouraged to review the repository for an in-depth exploration of the experimental framework and data. The repository is accessible via the following link:

https://github.com/Abdomash/ml-energy-consumption