Green Machine Learning: Analysing the Energy Efficiency of Machine Learning Models

1st Samara O. S. Santos
Hamilton Institute
Maynooth University
Maynooth, Ireland
samara.santos.2024@mumail.ie

2nd Agustina Skiarski
Hamilton Institute
Maynooth University
Maynooth, Ireland
agustina.skiarski.2024@mumail.ie

3rd Daniel García-Núñez
Hamilton Institute
Maynooth University
Maynooth, Ireland
daniel.garcianunez.2024@mumail.ie

4th Victor Lazzarini
Dept Music
Maynooth University
Maynooth, Ireland
victor.lazzarini@mu.ie

5th Rafael de Andrade Moral
Dept Mathematics and Statistics
Maynooth University
Maynooth, Ireland
rafael.deandrademoral@mu.ie

6th Edgar Galvan
Dept Computer Science
Maynooth University
Maynooth, Ireland
edgar.galvan@mu.ie

7th André L. C. Ottoni
Dept Computer Science
Federal University of Ouro Preto
Ouro Preto, Brazil
andre.ottoni@ufop.edu.br

8th Erivelton Nepomuceno
Dept of Electronic Eng
Maynooth University
Maynooth, Ireland
erivelton.nepomuceno@mu.ie

2024 35th Irish Signals and Systems Conference (ISSC) | 979-8-3503-5298-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISSC61953.2024.10603302

***Abstract*—The consumption of energy by Machine Learning (ML) has increased significantly. There is growing concern about the sustainable use of ML, where choosing the best ML model should also consider energy efficiency. The main objective of the Green Machine Learning paradigm is the simultaneous optimisation of accuracy and energy consumption. The literature has presented some suggestions for metrics to be used. However, these metrics have not been extensively compared among different ML models. To address this aspect, in this paper, we have analysed six Machine Learning models applied to three benchmark datasets for binary classification tasks, focusing on performance and energy consumption. The results of the F1-Score show that the random forests model outperformed the other models, while logistic regression was more energy efficient. These results demonstrate the trade-offs between model performance and energy consumption, providing valuable guidance for algorithm selection. Performance metrics are an essential benchmark, with Python's Scikit-Learn suite of models often outperforming neural networks in classification tasks. Future research should extend energy analysis to other machine learning methods and consider metrics that balance performance and energy consumption.**

***Index Terms*—Green Computing, Green Machine Learning, Energy-efficient Machine Learning, Computer Arithmetic**

I. Introduction

Green Computing commonly refers to an artificial intelligence system that is environmentally friendly and helps reduce carbon emissions [1], [2]. Lannelongue *et al.* [3] presented ten simple guidelines that can effectively improve the sustainability of computing applications. The applications of green computing are extensive and diverse, spanning fields such as robotics [4], biology [5], IoT [6], [7] and smart cities [8], [9].

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6049 and 21/FFP-P/10065. Erivelton Nepomuceno was supported by Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq (Grant No. 425509/2018-4 and Grant No. 311321/2020-8) and FAPEMIG (Grant No. APQ-00870-17).

An emerging area of research that has received considerable attention is Green Machine Learning (GML) [10], [11], which aims to develop strategies and techniques to reduce the carbon footprint and computational costs associated with Machine Learning (ML) environment. It is essential to focus on this aspect, as ML is increasingly used in industry [12] and research [13], addressing these algorithms' energy consumption and carbon emissions is imperative.

To achieve this goal, the first step is to measure the energy consumption of ML algorithms. The simplest way is to measure the processing time using functions such as 'tic-toc' in Matlab [14]. However, this procedure is not precise as other simultaneous processes run in the background, which may influence computing time. Additionally, it does not consider CO2 emissions, which depend on many other factors, such as whether computation is done locally or in the cloud and the energy consumption period. Another example of measurement is explained in the work of García-Martín *et al.* [15], a review paper that focuses on calculating how much energy is used when ML systems are running, aiming to make them more energy-efficient and environmentally friendly by understanding their energy demands.

This paper will employ an alternative approach to measuring energy consumption. It will use the CodeCarbon [16] library, which is available in Python. The main goal of this paper is to compare six machine learning models' performance and energy consumption metrics for binary classification on three benchmark datasets. The models being compared are Logistic Regression (LogReg), Support Vector Machine (SVM), Random Forests (RF), Gradient Boosting Classifier (XGBoost), Residual Neural Networks (RNN) and Multilayer Neural Networks (MLNN). Additionally, we investigate the impact of subsampling and feature reduction on predicting power and energy consumption. This analysis can provide valuable insights into ML models' performance and energy efficiency.

The contributions to Green Machine Learning are as follows:

- We thoroughly investigate six ML models' effectiveness and energy consumption using three benchmark datasets tailored to binary classification tasks. By systematically analysing the impact of pre-processing techniques, such as parameter and feature resolution methods, on power consumption, this research highlights strategies for optimising energy-efficient implementations of ML.
- This work goes beyond a simple performance evaluation by analysing model effectiveness and power consumption trade-offs. Through rigorous statistical testing, the most appropriate ML algorithms are identified based on both performance metrics and energy efficiency benchmarks. In particular, the results challenge conventional wisdom by highlighting traditional ML models' competitive performance and energy efficiency, such as logistic regression and random forests, compared to more complex neural network architectures.
- This work sets a precedent for future research efforts by advocating the extension of energy analysis to different ML modalities beyond classification tasks and the development of nuanced metrics that balance performance and energy considerations.

The remainder of this paper is divided into five sections. Section 2 discusses ML, Deep Learning, and the Friedman Test. Section 3 describes the methodology. Section 4 presents the results. Finally, Section 5 presents the conclusions.

II. Background

This section briefly discusses the topics covered in this paper: ML and Deep Learning, which are the core of this study, and the Friedman Test, which is used to analyse the results.

A. Machine Learning

Machine learning is a fundamental aspect of Artificial Intelligence (AI). Its goal is to enable algorithms to 'learn' from data, similarly to human learning processes. ML techniques aim to replicate cognitive abilities, allowing the machines to recognise patterns, make decisions and improve their performance over time. Examples of these techniques are supervised learning, unsupervised learning, reinforcement learning [17], [18], and deep learning, each with different objectives and data characteristics [19]. This study focuses on supervised learning models. The following ML models were used: Logistic Regression [20], Support Vector Machine [21], Random Forests [22], and Gradient Boosting Classifier [22].

B. Deep Learning

Deep Learning (DL) is an ML technique that uses layers of architecture inspired by the structure and function of the human brain [23]. Based on Artificial Neural Networks (ANNs), a data processing paradigm inspired by the way the biological nervous system processes data. It consists of interconnected layers and deep learning algorithms that excel at processing complex, high-dimensional data and delivering advanced results in tasks such as image recognition [24] and natural language processing [25]. The automatic extraction of complex features from raw data allows deep learning models to achieve performance in a wide range of domains. The following deep learning models are used in this study: Residual Neural Networks [26] and Multilayer Neural Networks [26].

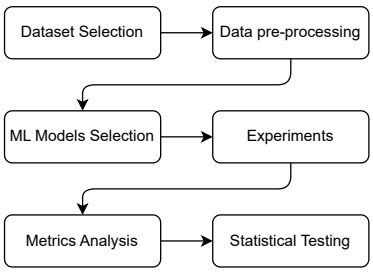

Fig. 1. Flowchart of the experiments performed to compare the performance of the ML models.

TABLE I
NUMBER OF SAMPLES AND FEATURES FROM THE DATASETS USED WITHIN THE EXPERIMENTS.

Name	# Samples	# Features
AIDS Clinical Trials Group Study 175 [28]	2139	23
TUANDROMD (Tezpur University Android Malware Dataset) [29]	4464	241
Predict students' dropout and academic success [30]	4424	36

III. Methodology

The methodology adopted in developing this paper is illustrated in Fig. 1. A detailed analysis of the diagram is presented in the subsequent sections.

A. Datasets

Three datasets from the UC Irvine repository for ML [27] were selected for the experiments based on the number of observations and the features. These datasets are suitable for investigating classification-oriented problems, and some of their parameters are presented in Table I.

Four different types of experiments were carried out with each of the datasets. The difference in each experiment relied on the data pre-processing step. These experiments are outlined as follows:

- Full datasets
- Sub-sampled datasets
- Full datasets with a reduced number of features
- Sub-sampled datasets with a reduced number of features

When reducing features, 95% of the variance was retained, and only 20% remained when reducing the number of observations.

B. Machine Learning Models

Six ML models were selected based on the work presented by [31]. The models were implemented using Python Version 3.11. Four of these models were implemented from the Scikit-Learn [32] Library Version 1.4, with the library's default hyperparameters. These models are now described: i) Logistic

TABLE II
ARCHITECTURE OF THE NEURAL NETWORKS USED

	RNN	MLNN
No. layers	4 (in residual block)	5
Activation function	ReLu	ReLu
Output layer activation function	Sigmoid	Sigmoid
Neurons per layer	256, 128, 64, 64	128, 256, 512, 256, 128

Regression, ii) Support Vector Machine, iii) Random Forests, and iv) Gradient Boosting Classifier. Two neural network models were designed specifically for this purpose: i) Residual Neural Networks and ii) Multilayer Neural Networks.

The construction of the neural network algorithms is explained in Table II in terms of the chosen architectures.

C. Experiments

To conduct the experiments in this paper, a computer with the following specifications was used:

- Model: Dell Latitude 5540 Laptop
- Platform System: Linux 6.2.0
- RAM Memory: 16 GB
- CPU model: 13th Gen Intel(R) Core(TM) i7-1365U

As mentioned in Section III-A, four different pre-processing methods were established for each dataset, resulting in 12 pre-processed datasets. The experiments were carried out immediately after the pre-processing and used stratified 10-fold cross-validation with four repetitions for each of the 6 ML algorithms, 12 pre-processed datasets, and 40 iterations per dataset, totalling 2,880 experiments.

D. Metrics

The metrics employed to assess the performance of the ML models were divided into two categories: classification performance and energy consumption. For classification performance analysis, the F1-Score was selected, represented by Eq. (1):

$$\mathrm{F1-Score} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{1}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

The experiment separately measured energy consumption in three stages: data pre-processing, training, and testing. The energy consumption (measured in kWh) was also evaluated using the CodeCarbon [16] library.

CodeCarbon employs a global carbon intensity of electricity per cloud provider or country for energy measurement. This data is augmented by a tracking system that monitors the power supply to the underlying hardware at frequent intervals, usually by default at every 15 seconds. A standardised ratio of 3 watts per 8 GB RAM is employed to ensure the accuracy of its calculations. Furthermore, the energy consumption of Intel processors is tracked by drawing from Intel RAPL files. It is crucial to highlight that the measurement is initiated upon the initial invocation of the start method and ceases operation upon the call to the stop method.

E. Statistical Test

The Friedman and Post-Hoc Tests were used to conduct statistical analyses and compare the energy consumption between the different experiments.

1) Friedman Test: The Friedman Test, a non-parametric statistical test that allows for comparisons between different data samples, was used to determine if there was a statistically significant difference among the metrics produced by each ML model.

The test was conducted using the *friedmanchisquare* function from the *Scipy Python* library [33]. Three assumptions were examined, leading to the consideration of a null hypothesis:

- The group (ML Models) is measured on three or more different occasions;
- The group is a random sample of the population;
- The dependent variable is measured at the ordinal or continuous level.

We used a significance level of 0.05.

2) Post-Hoc Tests: After performing the Friedman test and observing a significant result, it is conventional to perform post-hoc tests to identify specific pairwise groups that exhibit a considerable difference. This paper employed Miller's method to compare all pairs comprehensively [34].The scikit-posthocs [35] library was employed to perform the post hoc analysis.

Finally, the Condorcet contest was computed to compare the algorithms' performance. This method counts the number of positive encounters between two contenders, in this case, the number of positive significant differences found using the post hoc test between two models.

IV. RESULTS

To facilitate understanding, the results have been systematically categorised into different sections that examine the designated metrics, namely classification performance, energy consumption, and the statistical analyses carried out using the Friedman and post-hoc tests.

A. Classification Performance

Table III shows each model's mean and standard deviation of F1-Score values across all datasets. The best values for each dataset were highlighted. For the AIDS dataset, XGBoost showed the highest mean value using the full and the subsampled dataset. When using PCA, the MLNN had the highest mean value, and when employing subsampling and PCA, RF outperformed the rest of the models. On the other hand, for the MALWARE dataset, RF had the highest mean value independently of the pre-processing steps used. Finally, for the students' dataset, RF and XGboost obtained the highest mean value using the Full dataset, while logistic regression obtained a higher value when employing subsampling, PCA and both. The highest values were obtained for all the datasets when neither subsampling nor PCA was used.

TABLE III
MEAN AND STANDARD DEVIATION OF F1-SCORE VALUES FOR EACH MODEL ACROSS ALL DATASETS.

Dataset	Subsampling	PCA	LogReg	SVM	RF	XGBoost	MLNN	RNN
AIDS			0.6943 (0.05)	0.7329 (0.04)	0.7554 (0.04)	0.7683 (0.04)	0.7008 (0.04)	0.6990 (0.06)
	✓		0.6715 (0.05)	0.6634 (0.06)	0.6898 (0.05)	0.7307 (0.04)	0.1180 (0.15)	0.5262 (0.12)
		✓	0.5852 (0.05)	0.6359 (0.06)	0.6670 (0.06)	0.6776 (0.05)	0.6948 (0.05)	0.6675 (0.08)
	✓	✓	0.5783 (0.06)	0.5697 (0.06)	0.6342 (0.05)	0.6240 (0.06)	0.5334 (0.19)	0.5406 (0.17)
MALWARE			0.9919 (0.00)	0.9903 (0.00)	0.9964 (0.00)	0.9915 (0.00)	0.9905 (0.00)	0.9755 (0.04)
	✓		0.9860 (0.00)	0.9867 (0.00)	0.9917 (0.00)	0.9876 (0.00)	0.9768 (0.01)	0.9706 (0.02)
		✓	0.9889 (0.00)	0.9958 (0.00)	0.9966 (0.00)	0.9949 (0.00)	0.9913 (0.00)	0.9860 (0.01)
	✓	✓	0.9867 (0.00)	0.9879 (0.00)	0.9890 (0.00)	0.9867 (0.00)	0.9771 (0.01)	0.9768 (0.01)
Students'			0.9116 (0.01)	0.9079 (0.01)	0.9121 (0.01)	0.9121 (0.01)	0.9030 (0.01)	0.8959 (0.02)
	✓		0.9058 (0.01)	0.8909 (0.01)	0.9025 (0.01)	0.8989 (0.01)	0.8882 (0.02)	0.8892 (0.02)
		✓	0.8087 (0.00)	0.8084 (0.00)	0.7349 (0.02)	0.7957 (0.01)	0.8087 (0.00)	0.8086 (0.00)
	✓	✓	0.8086 (0.00)	0.8082 (0.00)	0.7163 (0.02)	0.7815 (0.01)	0.8081 (0.00)	0.8070 (0.00)

TABLE IV
MEAN AND STANDARD DEVIATION OF ENERGY CONSUMPTION DURING THE PRE-PROCESSING STEP

Dataset	Subsampling	PCA	Energy Consumption (mWh)
AIDS			0.0143 (0.01)
	✓		0.0397 (0.03)
		✓	0.0471 (0.04)
	✓	✓	0.0563 (0.03)
MALWARE			0.0575 (0.03)
	✓		0.0807 (0.05)
		✓	0.6520 (0.15)
	✓	✓	0.4141 (0.16)
Students'			0.0414 (0.05)
	✓		0.1063 (0.09)
		✓	0.1325 (0.08)
	✓	✓	0.1861 (0.13)

B. Energy Consumption

Table IV shows the energy consumption of the pre-processing step across all experiments. The pre-processing step was independent of the Machine Learning models used, so the mean and standard deviation were calculated using the values from all experiments. The best results can be observed when neither PCA nor subsampling is used. When analysing each experiment separately, it can be observed that the PCA method has a higher value than the subsampling. Usually, the value of the energy in the combination of PCA and subsampling is greater than when analysed separately, but the MALWARE dataset shows a different result.

Table V shows the energy expended during training. The logistic regression model exhibited significantly higher power consumption when compared to other ML methods, regardless of the pre-processing technique chosen. Contrary to the results observed in the pre-processing step, the combined use of PCA and subsampling in the training step gives a better result, using less energy than when no method is used, or each technique is used alone. Another notable observation is that the SVM model took significantly more time without using any method and only using PCA, but this was reduced using subsampling.

Table VI shows the energy consumption during the prediction step. Regardless of the dataset or the pre-processing method, the XGBoost model performs better than the other models analysed and achieves the best results. Again, some of the considerations made in the previous table can be observed, such as the lower energy consumption when both PCA and subsampling are used. In addition, the SVM model shows high values when the subsampling method is not used on the Students' dataset.

C. Post-Hoc Tests

Tables VII and VIII show the values obtained from the Miller-Friedman post-hoc test for the F1-Score and the sum of energy consumption during training and prediction, respectively, allowing a more comprehensive comparison between the different ML models. In these tables, the up arrow (↑) indicates that a positive statistically significant difference was found between the model in the row and the model in the column, the down arrow (↓) indicates a negative considerable difference, and the left/right arrow (↔) suggests that no significant difference was found. The arrows are sorted from left to right for each dataset: AIDS, MALWARE and Students'. The last column shows the result of the Condorcet contest, which sums the positive significant difference (↑) for each model.

Table VII shows the results for the F1-Score metric. When evaluating the F1-Score metric, the Random Forest model shows a positive and significant discrepancy compared to its counterparts within at least one of the datasets. This variation indicates the RF model performance advantage when evaluated using the F1-Score metric. The RF model obtained a result of 10 for the Condorcet method, and the second best result, the XGBoost model, was 6. The RNN model performs the lowest result.

Table VIII shows the statistical test results for the sum of the energy consumption during training and prediction. The logistic regression model is the most efficient regarding energy consumption, while the random forests model also performs well on this metric. Both models can be strong contenders, as there is no notable difference between them in the Condorcet Method. Once again, the worst result is seen for a Neural Network model, now in the MLNN model.

TABLE V
MEAN AND STANDARD DEVIATION OF ENERGY CONSUMPTION (mWh) DURING TRAINING STEP.

Dataset	Subsampling	PCA	LogReg	SVM	RF	XGBoost	MLNN	RNN
AIDS			0.0909 (0.05)	0.5174 (0.03)	0.5245 (0.02)	0.9150 (0.05)	1.8268 (0.17)	1.3575 (0.17)
	✓		0.0497 (0.02)	0.0604 (0.02)	0.2453 (0.04)	0.3046 (0.05)	1.4780 (0.23)	1.2008 (0.21)
		✓	0.0546 (0.02)	0.6364 (0.05)	0.9359 (0.05)	0.9197 (0.05)	2.0476 (0.21)	1.4248 (0.24)
	✓	✓	0.0385 (0.02)	0.0848 (0.02)	0.4210 (0.06)	0.4113 (0.07)	1.6352 (0.29)	1.2370 (0.27)
MALWARE			0.3419 (0.11)	1.8123 (0.08)	0.5303 (0.04)	1.6644 (0.05)	2.4698 (0.07)	1.7267 (0.58)
	✓		0.1632 (0.10)	0.2047 (0.03)	0.2978 (0.04)	0.4270 (0.04)	1.6093 (0.06)	1.3768 (0.70)
		✓	0.1627 (0.05)	0.8268 (0.05)	1.0557 (0.05)	2.7180 (0.11)	2.6874 (0.68)	1.3508 (0.11)
	✓	✓	0.1018 (0.05)	0.2581 (0.03)	0.4955 (0.05)	0.8137 (0.05)	1.8618 (0.79)	1.0933 (0.10)
Students'			0.1565 (0.07)	31.5021 (7.88)	1.0609 (0.06)	1.5637 (0.09)	2.4111 (0.07)	1.5509 (0.34)
	✓		0.1616 (0.14)	0.4614 (0.16)	0.3231 (0.04)	0.4523 (0.04)	1.6560 (0.48)	1.2722 (0.49)
		✓	0.0581 (0.02)	18.4786 (2.04)	1.0359 (0.11)	0.6075 (0.09)	2.6496 (0.39)	1.4675 (0.41)
	✓	✓	0.0388 (0.01)	0.4113 (0.12)	0.4959 (0.05)	0.3391 (0.04)	1.7399 (0.09)	1.1291 (0.10)

TABLE VI
MEAN AND STANDARD DEVIATION OF ENERGY CONSUMPTION (mWh) DURING PREDICTION STEP.

Dataset	Subsampling	PCA	LogReg	SVM	RF	XGBoost	MLNN	RNN
AIDS			0.0252 (0.02)	0.0656 (0.01)	0.0313 (0.01)	0.0087 (0.00)	0.1942 (0.01)	0.1632 (0.02)
	✓		0.0427 (0.04)	0.0331 (0.01)	0.0277 (0.01)	0.0093 (0.00)	0.1924 (0.01)	0.1723 (0.02)
		✓	0.0278 (0.01)	0.0571 (0.02)	0.0333 (0.01)	0.0089 (0.00)	0.1899 (0.01)	0.1665 (0.02)
	✓	✓	0.0210 (0.01)	0.0470 (0.01)	0.0329 (0.01)	0.0099 (0.01)	0.2052 (0.03)	0.1669 (0.03)
MALWARE			0.0274 (0.03)	0.1335 (0.01)	0.0367 (0.01)	0.0094 (0.00)	0.2115 (0.03)	0.2029 (0.03)
	✓		0.0291 (0.02)	0.0678 (0.02)	0.0282 (0.01)	0.0091 (0.00)	0.2042 (0.02)	0.2345 (0.04)
		✓	0.0261 (0.01)	0.0550 (0.02)	0.0290 (0.01)	0.0113 (0.01)	0.2056 (0.02)	0.1818 (0.03)
	✓	✓	0.0302 (0.03)	0.0376 (0.02)	0.0247 (0.01)	0.0075 (0.00)	0.2117 (0.03)	0.2954 (0.76)
Students'			0.0322 (0.03)	2.6525 (0.77)	0.0516 (0.01)	0.0102 (0.01)	0.1937 (0.02)	0.1740 (0.03)
	✓		0.0441 (0.04)	0.3550 (0.11)	0.0380 (0.01)	0.0087 (0.00)	0.2212 (0.05)	0.1777 (0.03)
		✓	0.0312 (0.01)	2.0865 (0.35)	0.0635 (0.03)	0.0071 (0.00)	0.2042 (0.03)	0.1719 (0.03)
	✓	✓	0.0267 (0.01)	0.3685 (0.12)	0.0474 (0.02)	0.0067 (0.00)	0.2109 (0.03)	0.1851 (0.03)

TABLE VII
MILLER-FRIEDMAN POST HOC TEST AND CONDORCET METHOD RESULT FROM THE F1-SCORE USING THE FULL DATASETS.

Model	LogReg	SVM	RF	XGBoost	MLNN	RNN	Con.
LogReg	X	↓↔↔	↓↓↔	↓↔↔	↔↔↑	↔↑↑	3
SVM	↑↔↔	X	↔↓↔	↔↔↔	↑↔↔	↔↑↑	4
RF	↑↑↔	↔↑↔	X	↔↑↔	↑↑↑	↑↑↑	10
XGBoost	↑↔↔	↔↔↔	↔↓↔	X	↑↔↑	↑↑↑	6
MLNN	↔↔↓	↓↔↔	↓↓↓	↓↔↓	X	↔↑↔	1
RNN	↔↓↓	↔↓↓	↓↓↓	↓↓↓	↔↓↔	X	0

TABLE VIII
MILLER-FRIEDMAN POST HOC TEST AND CONDORCET METHOD RESULTS FROM THE ENERGY CONSUMPTION DURING TRAINING AND PREDICTION USING THE FULL DATASETS.

Model	LogReg	SVM	RF	XGBoost	MLNN	RNN	Con.
LogReg	X	↑↑↑	↔↔↔	↑↑↑	↑↑↑	↑↑↑	12
SVM	↓↓↓	X	↔↓↓	↔↓↓	↑↔↔	↑↔↓	2
RF	↔↔↔	↔↑↑	X	↑↔↔	↑↑↑	↑↑↑	9
XGBoost	↓↓↓	↔↑↑	↓↔↔	X	↑↑↑	↔↔↔	5
MLNN	↓↓↓	↓↔↔	↓↓↓	↓↓↓	X	↔↓↔	0
RNN	↓↓↓	↓↔↑	↓↓↓	↔↔↔	↔↑↔	X	2

V. CONCLUSIONS

This study analysed six ML models' energy consumption and classification performance in binary classification tasks. The experiments used three benchmark datasets with tabular numeric data from the UCI database. Furthermore, four experiments were performed using different pre-processing steps for each machine learning model: (i) full dataset, (ii) subsampling 20% of the dataset, (iii) PCA maintaining 95% of covariance, (iv) and performing both subsampling and PCA steps.

The experiments showed that PCA consumes more energy than the subsampling step during the pre-processing step. Moreover, the subsampling step, in most cases, reduces the energy consumption during training more effectively than PCA. When using both, the energy consumption increased during the pre-processing in two datasets but decreased in one. This could be justified because the subsampling step is performed before PCA, reducing energy consumption. Nevertheless, the energy consumption during the training step was mainly reduced using both pre-processing steps.

Regarding the F1-Score, the subsampling step performed better in the AIDS and Students dataset than when using PCA in most of the experiments. At the same time, PCA outperformed in the MALWARE dataset. This was justified because the latter dataset has a larger number of features.

Finally, a post hoc Friedman test was computed to investigate statistical significance, and the Condorcet method was employed to compare the different models. The F1 Score and the sum of the training and prediction energy consumption from the Full dataset experiments were used in this comparison. The results showed that the Random Forest was the model with the best performance regarding the F1 Score, while the Logistic Regression was regarding the energy consumption. However, when comparing both models, Random Forest was statistically significantly better in two datasets relating to the F1-Score. At the same time, no significant difference was found regarding energy consumption. Therefore, Random

Forest was shown to be the most robust algorithm within the task at hand.

In the future, this energy analysis will be extended using more datasets with different properties and tasks. Additionally, it would be highly interesting to include other metrics to compare the models' performance.

ACKNOWLEDGMENTS

The authors are grateful to Maynooth University and Science Foundation Ireland. This publication has emanated from research supported partly by a grant from Science Foundation Ireland under Grant number 18/CRT/6049 and 21/FFP-P/10065. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. Erivelton Nepomuceno was supported by Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq (Grant No. 425509/2018-4 and Grant No. 311321/2020-8) and FAPEMIG (Grant No. APQ-00870-17).

REFERENCES

[1] T. Nazaré, J. Gadelha, E. Nepomuceno, and R. Lozi, "Green computing for energy transition: A survey," *IEEE Latin America Transactions*, vol. 21, no. 9, pp. 937–948, 2023.

[2] T. Nazaré, K. Mahmoodi, and E. Nepomuceno, "Green-Box System Identification and Carbon Footprint Analysis for Sustainable Computing," in *OCEANS 2023 - Limerick*, pp. 1–6, June 2023.

[3] L. Lannelongue, J. Grealey, A. Bateman, and M. Inouye, "Ten simple rules to make your computing more environmentally sustainable," *PLOS Computational Biology*, vol. 17, p. e1009324, Sept. 2021.

[4] S. E. Alves Filho, A. M. F. Burlamaqui, R. V. Aroca, L. M. G. Gonçalves, and S. T. de Lima Sa, "Green robotics: Concepts, challenges, and strategies," *IEEE Latin America Transactions*, vol. 16, no. 4, pp. 1042–1050, 2018.

[5] S. Kumar, "Embracing green computing in molecular phylogenetics," 2022.

[6] S. B. Othman, F. A. Almalki, C. Chakraborty, and H. Sakli, "Privacy-preserving aware data aggregation for iot-based healthcare with green computing technologies," *Computers and Electrical Engineering*, vol. 101, p. 108025, 2022.

[7] M. S. Batta, H. Mabed, and Z. Aliouat, "State of health optimization based unequal clustering in iot networks," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 125–130, IEEE, 2022.

[8] X. Huang, S. Chen, D. Xiong, C. Xu, and Z. Yang, "Analysis and prediction of influence factors of green computing on carbon cycle process in smart city," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[9] H. Ye, F. Li, Z. Liu, and X. Deng, "A green energy consumption policy of bluetooth mobile devices for smart cities," *Computing*, vol. 102, no. 4, pp. 1077–1091, 2020.

[10] L. Lannelongue, J. Grealey, and M. Inouye, "Green Algorithms: Quantifying the Carbon Footprint of Computation," *Advanced Science*, vol. 8, no. 12, p. 2100707, 2021.

[11] T. Tornede, A. Tornede, J. Hanselle, F. Mohr, M. Wever, and E. Hüllermeier, "Towards Green Automated Machine Learning: Status Quo and Future Directions," *Journal of Artificial Intelligence Research*, vol. 77, pp. 427–457, June 2023.

[12] D. A. Narciso and F. Martins, "Application of machine learning tools for energy efficiency in industry: A review," *Energy Reports*, vol. 6, pp. 1181–1199, 2020.

[13] A. Birhane, P. Kalluri, D. Card, W. Agnew, R. Dotan, and M. Bao, "The values encoded in machine learning research," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 173–184, 2022.

[14] P. Steno, A. Alsadoon, P. W. C. Prasad, T. Al-Dala'in, and O. H. Alsadoon, "A novel enhanced region proposal network and modified loss function: Threat object detection in secure screening using deep learning," *The Journal of Supercomputing*, vol. 77, pp. 3840–3869, Apr. 2021.

[15] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, 2019.

[16] V. Schmidt, K. Goyal, A. Joshi, B. Feld, L. Conell, N. Laskaris, D. Blank, J. Wilson, S. Friedler, and S. Luccioni, "CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing," *Zenodo*, 2021.

[17] P. M. Oliveira, J. M. Palma, E. G. Nepomuceno, and M. J. Lacerda, "Reinforcement learning for control design of uncertain polytopic systems," *Information Sciences*, vol. 625, pp. 417–429, 2023.

[18] A. L. Ottoni, E. G. Nepomuceno, M. S. d. Oliveira, and D. C. d. Oliveira, "Reinforcement learning for the traveling salesman problem with refueling," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2001–2015, 2022.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education Limited: London, UK, 4 ed., 2022.

[20] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, pp. 135–139, IEEE, 2019.

[21] M. Awad, R. Khanna, M. Awad, and R. Khanna, "Support vector machines for classification," *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pp. 39–66, 2015.

[22] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[24] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.

[25] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[26] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020.

[27] D. Newman, S. Hettich, C. Blake, and C. Merz, "Uci repository of machine learning databases," 1998.

[28] National Institute of Allergy and Infectious Diseases (NIAID), "A randomized, double-blind phase II/III trial of monotherapy vs. combination therapy with nucleoside analogs in HIV-infected persons with CD4 cells of 200-500/mm3." submitted: November 2, 1999.

[29] P. Borah, D. Bhattacharyya, and J. Kalita, "Malware dataset generation and evaluation," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, pp. 1–6, IEEE, 2020.

[30] M. V. Martins, D. Tolledo, J. Machado, L. M. Baptista, and V. Realinho, "Early prediction of student's performance in higher education: A case study," in *Trends and Applications in Information Systems and Technologies: Volume 1 9*, pp. 166–175, Springer, 2021.

[31] M. Z. Naser, "Do We Need Exotic Models? Engineering Metrics to Enable Green Machine Learning from Tackling Accuracy-Energy Trade-offs," *Journal of Cleaner Production*, vol. 382, p. 135334, Jan. 2023.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] P. Virtanen and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[34] R. G. Miller *et al.*, *Simultaneous statistical inference*, vol. 58. Springer, 1981.

[35] M. Terpilowski, "scikit-posthocs: Pairwise multiple comparison tests in python," *The Journal of Open Source Software*, vol. 4, no. 36, p. 1169, 2019.