National College of Ireland

Green AI: Quantifying the Carbon Footprint of Machine Learning Models Through Multi-Model Energy Benchmarking and Emissions Analysis

MSc Research Project
Data Analytics

Lakshmi Meena Manivannan
Student ID: x23426918

School of Computing
National College of Ireland

Supervisor: Sallar Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Lakshmi Meena Manivannan

Student ID: x23426918

Programme: MSc Data Analytics **Year:** 2025 - 2026

Module: MSc (Research) Practicum/Internship

Supervisor: Prof Sallar Khan

Submission Due Date: 28/01/2026

Project Title: Green AI: Quantifying the Carbon Footprint of Machine Learning Models Through Multi-Model Energy Benchmarking and Emissions Analysis

Word Count: 7767 **Page Count:** 24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Lakshmi Meena Manivannan

Date: 27/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Green AI: Quantifying the Carbon Footprint of Machine Learning Models Through Multi-Model Energy Benchmarking and Emissions Analysis

Lakshmi Meena Manivannan

X23426918

The project is an environmental assessment of classical machine learning models, which is done through a combined analysis of their predictive performance, energy consumption and approximate carbon emissions. It uses two real-world datasets one of the air-quality sensor datasets and one of the electricity-consumption time-series datasets. The trained models are Logistic Regression, Support Vector machine, Random Forest, Multilayer Perceptron and XGBoost, which are run in a controlled pipeline and all experiments tracked with the help of CodeCarbon library. The models have a predictive accuracy of 72-82% (R 2 on regression tasks or the same accuracy in classification tasks) the highest scores being achieved by Random Forest and XGBoost. Nevertheless, such sophisticated models produce up to 45x more CO 2 than lighter baselines like Logistic Regression, and do not increase accuracy by more than 2 -3 points. The visualizations of Streamlit dashboard are created to visualise the per-model emissions, runtime and efficiency rankings, and allow a transparent comparison of accuracy-emissions trade-offs, and aid the choice of greener models.

Keywords: Green AI, Carbon Emissions, Machine Learning Efficiency, Code Carbon, Sustainable Computing, Model Benchmarking, Energy Consumption

1 Introduction

The growing use of machine learning in any field has led to large computational requirements, which has consumed more energy and a quantifiable carbon footprint. Though most works have placed more emphasis on the predictive performance of algorithms, the environmental cost of training these models have emerged as a major issue of concern particularly with the current emphasis on sustainability as a global agenda.

1.1 Background

Green Artificial Intelligence is a new discipline that focuses on efficacy, visibility, and ecological soundness to the creation and assessment of machine learning devices. The calculation and comparison of carbon emission during the training of the model enable the researchers to take a better decision on the choice of the algorithm. Such measurements can be done with the help of tools like CodeCarbon which allow estimating energy consumption and emissions reliably. Here, a study of the carbon footprint of classical machine learning models trained using real-world data provides viable information about the interplay between predictive performance and its effect on the environment.

1.2 Problem Statement

Even though the predictive accuracy of machine-learning models can be high, the cost of training the models is rarely considered in environmental research. The majority of the current research is centered on accuracy though important variations in energy consumption, training time, and carbon emission across algorithms are neglected even to traditional models like Logistic Regression and Random Forest. Besides, the absence of standardized benchmarking structures to make consistent measurements of emissions across datasets, models, and feature-engineering strategies exist. In this project, the authors fill these gaps by benchmarking various machine-learning models in controlled settings and quantitatively measuring their computational and environmental impacts in an empirical study.

1.3 Research Question

The overall research question that will guide this project is:

"How do different machine learning algorithms compare in terms of carbon emissions, energy consumption, and predictive performance when trained on real-world datasets?"

Based on this the following sub-questions are obtained:

- **RQ1:** Do the complex machine learning models (including Random Forest, XGBoost and Multilayer Perceptron) generate considerably more carbon emissions than lightweight ones (including Logistic Regression)?
- **RQ2**: How is there a statistical correlation between model runtime, energy consumption and CO 2 emissions when models are trained under the same computational conditions?
- **RQ3:** What are the effects of feature engineering decision and the characteristics of data sets on the emissions and running behaviour of classical machine learning models?
- **RQ4:** Does an interactive emissions dashboard enhance interpretability and help in selecting a model practitioners find environmentally responsible? All these questions are about the direction of this investigation as well as defining the analytical methods to be applied during the project.

This hypothesis is the analytical basis of the comparison of model categories and evaluation of the trade-offs of the environment embodied with the help of the CodeCarbon monitoring.

1.4 Aim and Objective

The goal of the research is to develop a Green AI benchmarking system which is going to examine the machine learning models based not only on the predictive accuracy of the model but also on the amount of energy consumed by the model and the carbon emissions created by the model. The paper to achieve this objective follows the development of inter-connective objectives. Firstly, it coded and patterned high quality properties of two datasets on the actual world and optimized them in the modelling of supervised, unsupervised and regression. Subsequently, it takes into account the predictive potential of numerous classical algorithms: Logistic Regression, Random Forest, XGBoost, MLP, KMeans, Gaussian Mixture Models, etc. CodeCarbon is also integrated into the project to record categorically the emissions, processing by the CPU, memory utilization and the duration required to train a given model. These measurements are analyzed in order to understand what trade-offs will achieve between model accuracy, computational efficiency, and the environmental impact.

1.5 Scope of the Research

The models that are considered in this study are the classical machine learning models, which are easy to interpret, reproduce, and can be computed using a local machine with manageable computational costs. The steps involve base modeling, feature engineering, hyper parameter tuning, and emission tracking. CPU-based training sessions are performed on a local machine, and the measurements of energy are obtained using CodeCarbon. In this research, the deliberate choice was to use CPU-based classical machine learning models as a controlled, reproducible, and low-emission benchmark (as opposed to deep learning and cloud-based training, which are expensive and unpredictable) to ensure a high carbon cost.

1.6 Significance of the Study

This study can be viewed as a contribution to the emerging field of sustainable machine learning as it illustrates a methodical way of assessing models that consider predictive performance and consumption at the same time. The article presents empirical findings of the performance-efficiency trade-offs of a variety of algorithms. This project shows that AI development is environmentally conscious, with carbon emissions added to the model analysis. The results facilitate more sustainable practices in machine learning processes and facilitate larger-scale work on minimizing the environmental impact of AI systems.

2 Related Work

The chapter provides an overview of scholarly underpinnings that would guide sustainable machine learning practices, carbon accounting, energy-efficient model design, benchmark in methodologies, and reproducible green AI experimentation. This review is done to offer a conceptual framework against which the effect of machine learning workflows on the environment should be analyzed, as well as to endorse the methodology used in this project. Modern studies indicate that there is a fast-increasing interest in tools, models, and modelling methods that measure or reduce the carbon footprint of computational tasks, pointing to the movement towards more responsible AI engineering.

2.1 Green AI and Environmental Impact of Machine Learning

Machine learning (especially deep learning) has grown so fast that it has increasingly generated computational requirements that are challenging to the environmental sustainability. Schwartz et al. (2020) presented the concept of Green AI, which states that efficiency and transparency of research should be valued above the accuracy gained. Their work forms a conceptual framework of developing AI systems that will be both powerful and eco-friendly.

Strubell, Ganesh and McCallum (2019) provided the foundation to show that training large natural language processing (NLP) models can incur hundreds of kilograms of CO 2 -equivalent emissions -equivalent in quantity to the emissions of several cars over a lifetime. This analysis was extended to larger scale transformer architectures (BERT and GPT) by Patterson et al. (2021), revealing that it is not environmentally sustainable to scale computational workloads without explicit efficiency interventions.

2.2 Carbon Footprint Calculation Frameworks and Monitoring Tools

The Green AI research would need proper carbon accounting frameworks. One of the earliest models to estimate the CO 2 emissions during model training was suggested by Lacoste et al. (2019), which included the energy consumption and the data on carbon intensity in the region. Their solution laid the theoretical basis of modern surveillance devices.

Henderson et al. (2020) reviewed the topic of reporting of machine learning research energy usage with the standardisation of energy use and suggested that hardware settings, runtime, and software settings be documented to enhance transparency and reproducibility.

2.3 Energy Consumption Characteristics of Machine Learning Models

Computationally, machine learning models have different behaviours, and this behaviour directly determines the energy consumption and carbon emission. Xu, Wang and Lin (2021) have made a very broad review of energy-efficient machine learning techniques, and they indicate that the classical algorithms like logistic regression and decision trees often need lower computation than deep neural networks.

Random Forest and gradient-boosting methods are also tree-based ensemble models, meaning that they are computationally more expensive than simpler linear models, but more efficient than large neural architectures because they can be computationally parallelised (Zhang, Chen and Wu, 2021). Neural networks, in contrast, are more energy-intensive as they use an iterative gradient-based optimisation process, and their training process frequently needs a series of training epochs (Patterson et al., 2021).

2.4 Energy Reduction and Efficiency Optimisation Techniques

An increasing number of studies discuss methods of mitigating the environmental impact of machine learning. The article by Lv, Li and Yang (2022) identify the topics of model compression, resource-efficient architecture and sustainable system design and shows that dramatic energy savings can be made without compromising accuracy. Schwartz and Dodge (2020) speak about efficiency trade-offs in model design and note that algorithmic decisions are expected to be made with keeping both the cost of calculations and the predictive performance in mind.

Hyperparameter optimisation also adds much to emissions in training and a number of research suggest methods of eliminating unnecessary computation. Kaack et al. (2022) recommend the creation of AI and climate-conscious policies, such as searching the space of tuning experiments limited. Williams (2020) also emphasizes the necessity to implement the concept of sustainability into machine learning processes.

2.5 Benchmarking, Reproducibility, and Fair Comparisons

Strict reproducibility is required in order to benchmark machine learning emissions reliably. Henderson et al. (2020) suggest that compared to other applications, emission-related comparisons need a uniform hardware, software, data preprocessing pipelines, and measurement tools. Zhang, Chen and Wu (2021) suggest standardised benchmarking protocols to be used in measuring the energy consumption under various models, and emphasise that the only comparisons that are valid are under the same training conditions. Li et al. (2020) also focus their attention on system-level optimisation of distributed machine learning and offer an understanding of the impact of hardware and system design on energy consumption.

2.6 Research Niche

Despite growing interest in sustainable machine learning, a lot of the current literature concentrates on big-data deep learning models, without much regard to classical models that are widely used in practice. Theoretical or high-level analyses of AI emissions are also possible by a number of studies (Kaack et al., 2022; Williams, 2020), yet empirical and model-level comparisons with different algorithms are scarce. This project fills this gap by performing systematic research on energy consumption, carbon emissions, runtime, and predictive performance over a variety of supervised and unsupervised models based on two real-world datasets, air quality and electricity consumption.

2.7 Summary Table

Reference	Dataset/ Context	Type	Method	Shortcomings of Previous Work	Insights for This Project
Strubell et al. (2019)	NLP model training workloads	Carbon impact	Measured CO_2 emissions from deep NLP models	Focused on large NLP models only	Highlights need to assess emissions even for simpler classical models
Schwartz et al. (2020)	Conceptual/ theoretical	Green AI	Efficiency-oriented AI framework	Limited empirical evaluation	Motivates accuracy–efficiency comparisons used in this project
Henderson et al. (2020)	ML reproducibility studies	ML efficiency	Variance-aware energy measurement	No multi-model benchmarking	Supports use of consistent experiment tracking
Lacoste et al. (2019)	Training energy estimation	Carbon tracking	CO_2 footprint calculation model	Does not present multi-algorithm comparisons	Forms basis for CodeCarbon integration in this project
Patterson et al. (2021)	Large neural networks	Efficiency benchmarking	Scaling analysis of transformers	Focused on massive cloud workloads	Highlights contrast with small-scale

					models tested here
Kaack et al. (2022)	Multiple public AI systems	Climate-aware AI	Policy framework for sustainable AI	Limited algorithm-level analysis	Justifies sustainability motivations
Schwartz and Dodge (2020)	ML efficiency trade-offs	Energy–accuracy	Analytical comparison	Lacks dataset-specific experiments	Supports need for model comparison
Williams (2020)	Sustainable AI	Ethical and policy discussion	Environmental implications	Non-empirical	Reinforces need for empirical emissions testing
Li et al. (2020)	Distributed ML systems	System optimisation	Runtime and power efficiency	Not focused on per-model emissions	Provides system-level sustainability context

3 Research Methodology

The research methodology outlines the structured process followed to investigate the carbon footprint of machine learning models and analyse their performance-efficiency trade-offs. It describes the sequential stages of data acquistion, prepocessing, feature engineering, model development, evaluation, and carbon-emission assessment implemented in the study.

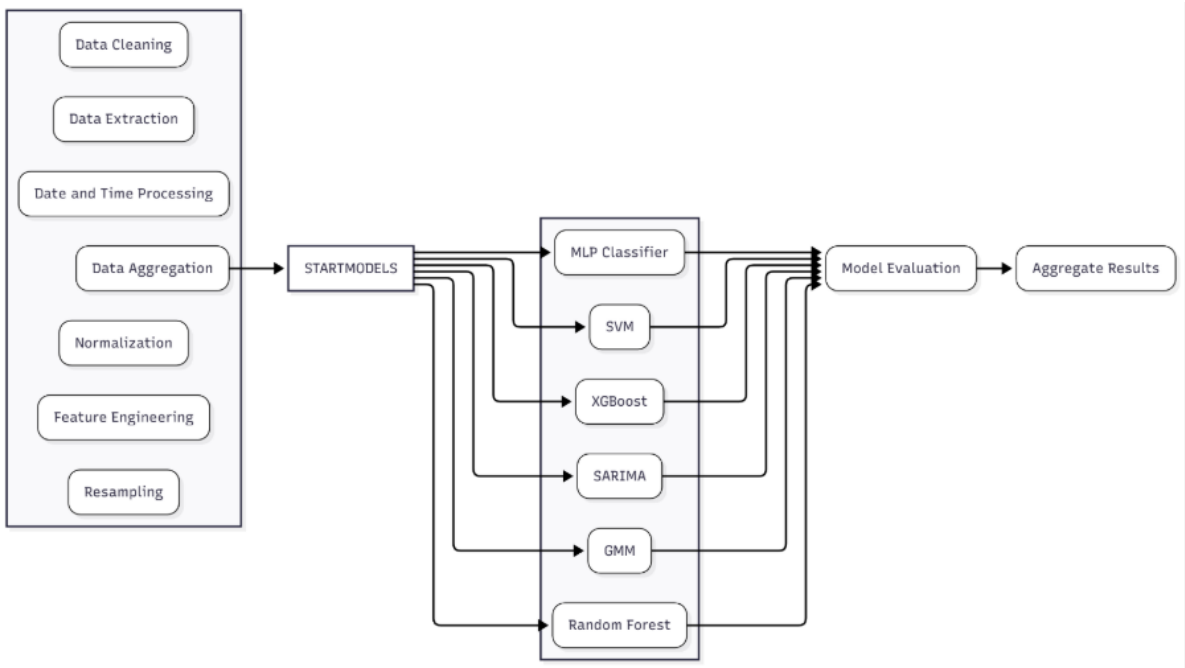

Figure 1: Methodology Architecture Diagram

3.1 Data Collection

Two publicly available datasets were used in this study, namely air quality prediction and electricity consumption modelling. The Air Quality UCI data has about 9300 hourly measurements of gaseous pollutants, such as CO, NO 2, NOx, and benzene, as well as meteorological parameters, such as temperature and humidity. The electricity consumption data is composed of the aggregated readings that may be utilized to examine the long-term consumption patterns and seasonality. Besides these datasets, another dataset of emissions was kept documenting the energy usage and the approximated carbon emission when training the model. The data containing air quality was dimensional and featured engineered higher dimensionality and higher run time and emissions compared to the lower dimensional data containing electricity, and extra datasets were not used to maintain consistency in benching. Collectively, these datasets will make it possible to conduct systematic comparisons between model accuracy, computational efficiency, and environmental impact. Such a combined arrangement facilitates a consistent and repeatable assessment of the Green AI practices.

3.2 Data Pre-processing

The pre-processing phase was aimed at the quality, consistency and usability of data in the two datasets. The air-quality data underwent a considerable amount of cleaning to deal with missing, inconsistent and corrupted sensor data, such as the conversion of non-numeric data and the removal of rows with irretrievable missing data. Since the raw air-quality data did not have a single timestamp, a unified data on the date-time was created by taking known date and time attributes, and the electricity data were put into date-format standardization in order to allow them to be chronologically ordered. Lastly, the two datasets were structured into both feature and target variables so as to facilitate further analytical and predictive modelling processes.

3.3 Data Transformation

Data transformation was undertaken to prepare the cleaned datasets for modelling. For the air quality dataset, the target variable CO(GT) was converted into a binary label representing normal versus elevated carbon monoxide levels, enabling the formulation of a supervised classification task. For the electricity dataset, the monthly consumption variable served as the prediction target, and several lag-based features were generated to capture temporal dependencies and seasonality. These included one-month, three-month, six-month, and twelve-month lag indicators. A consistent train–test split was applied to both datasets to ensure robust and comparable model evaluation. The transformations implemented in this stage serve to highlight the underlying structure of each dataset and augment model interpretability.

3.4 Feature Engineering

The two datasets both were subject to feature engineering to make sure that supervised, unsupervised and regression models could learn useful patterns and still be computationally efficient. In the Air Quality dataset, the variables of the pollutant's sensors (CO, C6H6, NOx, NO 2) and meteorological variables were aggregated in a single timestamp formed by the use of date and time variables. Hour, day and month were taken as time-based features to identify the effects of the day, and season. Dependences on the short and middle term were modelled with lag features (lag 1 and lag 2) and rolling mean (3, 6 and 12 steps), and rows with induced

missing values were eliminated. The transformations were followed in supervised and unsupervised models to provide identical benchmarking.

In the case of the Electricity Consumption dataset, the feature engineering strategy aimed at taking advantage of time-series behaviour, which was the automatic identification of the target of consumption and the elimination of redundant timestamp fields. The lag features (lag 1, 3, 6 and 12) were added to represent persistence and seasonality and forecasting and clustering efficacy were enhanced. All the models were trained in the same state of CPU-only so that the results of the emissions could be reproducible. The experiments were performed using 13th Gen Intel Core i5-1335U processor with RAM of 15.7GB and the estimation of carbon emissions was done using regional carbon-intensity factors offered by CodeCarbon. This controlled configuration allowed the comparison of computational efficiency and environmental impact of all models in a consistent manner. The consumption column was detected with the help of automatic target detection, and all the fields related to the timestamps were eliminated to eliminate redundancy. Lag features like lag 1, lag3, lag6 and lag 12 were used to model seasonality and month-to-month persistence. These characteristics enhance the forecasting models and assist in the unsupervised clustering activities as they reveal patterns of similarity over time.

3.5 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was used to gain insight into the statistical characteristics and distributional behaviour of the variables in the two datasets. Data related to air quality had typical right skewed pollutant distributions and strong correlation among chemically related sensors whereas electricity data had clear seasonal consumption patterns as indicated by monthly variation plot as well as correlation matrices. Two major visualisations that will be suggested to include in the report, as recommended by the style of the sample thesis, are:

- Histogram grids with the features of the distribution of the gas sensor outputs and electricity consumption.
- Correlation plots showing correlation between pollutants and lag variables and target values.
- The findings of EDA were used in the modelling plan and were useful in testing assumptions regarding data behaviour before the application of the machine learning algorithms.

3.6 Statistical Modelling and Machine Learning

The modelling step included designing and testing several supervised learning models to be able to classify air quality and predict electricity usage. In the case of classification, the Logistic regression, random forest, support vector machine, Multilayer perceptron and XGBoost models were trained using the engineered predictors. Random Forest, Multilayer Perceptron, and XGBoost regressors were used to provide estimate of monthly electricity consumption in the regression task. All the models were trained on standardized features and were measured by performance metrics specific to each model, such as accuracy, F-score, mean absolute error, and interpretation of a confusion matrix.

3.7 Supervised Learning Approaches for Classification and Regression

Logistic Regression

This linear model served as the baseline classifier for predicting high vs. low air-quality levels. It provided interpretable coefficients and helped benchmark more complex methods.

Random Forest Classifier

This ensemble method handled non-linear pollutant interactions well and offered robust performance. It also provided feature-importance measures that supported interpretability.

XGBoost Classifier

XGBoost was applied to improve classification accuracy using gradient-boosted decision trees. It offered strong predictive power but required more computation, contributing to higher energy use.

Support Vector Machine (RBF Kernel)

SVM with a radial basis function kernel effectively modelled complex boundaries in the pollutant data. It provided strong accuracy but was computationally more demanding.

MLP Classifier (Neural Network)

The MLP classifier captured deeper non-linear patterns in the air-quality dataset. While accurate, its iterative training increased computational load and carbon emissions.

3.8 Unsupervised Learning Techniques for Pattern Discovery

K-Means Clustering

Applied to both datasets to identify natural groupings based on pollutant behaviour and electricity usage. Silhouette scores were used to evaluate cluster quality.

Gaussian Mixture Model (GMM)

A probabilistic clustering model that allowed soft cluster assignments, providing flexible cluster boundaries for environmental data.

3.9 Carbon Footprint Assessment

In order to introduce environmental responsibility into the machine learning process, this project will use CodeCarbon that will assess the amount of carbon emitted in the course of the model training. Hardware usage, execution time, and regional carbon intensity are recorded in each supervised, unsupervised and regression experiment to generate an open estimate of environmental impact. This allows comparing models in terms of predictive accuracy as well as computational sustainability.

CodeCarbon calculation of emissions is based on the following basic formula:

$$CO_2\ Emitted\ (kg) = Energy\ Consumed\ (kWh)\ \times Carbon\ Intensity\ (kgCO_2eq/kWh)$$

Monitored hardware power usage is estimated to be energy consumption itself:

$$Energy\ (kWh) = (CPU\ Power + GPU\ Power + RAM\ Power)\ \times Runtime\ (hours)$$

Where:

- System sensors are gathered to obtain CPU Power, GPU Power, and RAM Power.
- Runtime is the time in which the training cycle executes,
- Carbon Intensity is calculated based on geographical location (Ireland in this case), which is obtained on the sources of online energy-mix data.

Each model was tested on the same hardware, such as a 13 th Gen Intel ® Core i5-1335U with 15.72 GB of RAM to make the experiments consistent. The comparison showed that there were an evident trade-off and the predictive performance between the Random Forest and the XGBoost was high, yet the four to fivefold greater emissions than lightweight models like the Logistic Regression were observed. Multilayer Perceptron also took much more time to train and used more energy with only slight improvements in performance.

3.10 Dashboard Development

All the modelling and emissions outcomes are visualised by providing interactive Streamlit dashboard, which visualises CodeCarbon metrics which are runtime, energy use, carbon emissions, and green efficiency using Plotly charts to allow a simple comparison of accuracy environmental trade-offs to models.

The dashboard has a number of important features:

- Summary Table that shows the averaging of the emissions, runtime and energy consumption per model.
- Interactive Visualisations such as the CO2 emission, energy consumption and the length of training bar charts.
- Green Efficiency Ranking, determining the most and least sustainable models on a combination basis.

4 Design Specification

In this section, the system architecture that was created to compare performance of machine-learning models based on concurrent prediction performance and environmental impact is described, along with the structure of this architecture, its functionality, modelling, and software features that compare carbon emissions of algorithms.

4.1 Architecture Overview

This project architecture is both modular and sequential including data preparation, feature engineering, supervised and unsupervised modelling, and carbon tracking. The process starts with two real-world datasets, one of them being the Air Quality UCI dataset and monthly electricity consumption data. These data are processed and refined to offer consistency, such as missing values, data type corrections, and the derived variables, including pollutant indicators, temperature-based indices and lag variables. After the preprocessing, then the data is split into classification, regression and clustering tasks depending on their properties. The modelling layer uses supervised as well as unsupervised algorithms at different degrees of complexity and cost of computation. During this, CodeCarbon is used to monitor

energy usage, CPU use and approximated emissions, with a streamlit interface then displaying the model performance and sustainability information in an accessible format.

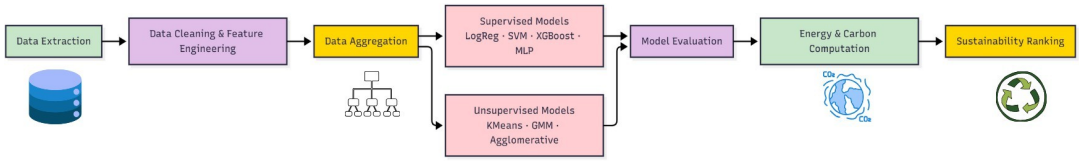

Figure 2: Model Implementation Diagram

4.2 Functional and Non-Functional Requirements

The system will be configured to load, clean and transform heterogeneous data sets after which several machine learning models will be trained and their carbon emissions monitored. It must deal with numerical and time-based data, create engineered features, and do classification and regression, as well as clustering techniques to discover the latent structure of the data. The system must also exhibit reliability, scalability and be easy to use besides functional behaviour. The design is environmental conscious as all modelling processes will be monitored in terms of energy consumption and emission. The clean Streamlit interface is also a tool that supports usability even when dealing with non-technical users who can interact with results without completing the underlying code.

4.3 Algorithm and Model Description

The modelling phase checks the predictive performance and the carbon efficiency with the help of the algorithms that are monitored formally. The Air-quality classification is done using Logistic Regression, Random Forest, SVM (RBF), XGBoost, and a Multilayer Perceptron, which include linear, ensemble, kernel, boosting, and neural. Unsupervised algorithms, such as K-Means, Gaussian Mixture Models, DBSCAN, and Agglomerative Clustering, are used to study the patterns of the structure and the dimensionality reduction and visualization are performed with the help of PCA. The algorithms are run in a carbon-tracking wrapping which tracks runtime, CPU energy consumption, RAM energy consumption and overall emissions. This establishes an integrated system that connects the performance of machine-learning and environmental impact indicators.

4.4 Data and Software Requirements

The project involves two datasets to measure the performance of the models and the cost of the energy: Air Quality UCI dataset with hourly pollutants, temperature, and humidity in the dataset, and the monthly electricity consumption dataset to perform time-series regression analysis. The datasets are both cleaned, imputed, transformed and feature engineered before modelling. All experiments are coded in Python, with the help of Pandas and NumPy to process the data, Scikit-Learn and XGBoost to model, Plotly and Matplotlib to visualize, and the CodeCarbon to monitor the emissions. The end product will be in the form of an interactive Streamlit dashboard. They are tested on a consumer grade laptop (Intel i5-1335U, no GPU)

such that they can be realistic and reproducible in terms of measuring the use of the computational resources and the impact on carbon.

4.5 Summary

The system design incorporates the following data preparation, feature engineering, machine learning, carbon tracking, and interactive visualisation in a logical workflow. The project integrates predictive analytics to environmental metrics, which is a systematic approach to assessing machine-learning model sustainability. The design specification forms the basis of the system implementation and experimental outcomes that are to be provided in future chapters.

5 Implementation

A structured and reproducible pipeline (the combination of classical machine-learning models and carbon-tracking tools) was used in this project. The plan of work was divided into three key steps, namely the preparation of the data sets, the training of the models (supervised and unsupervised) and the determination of the energy usage and carbon emissions. Python was the programming language used in all experiments and the Jupyter Notebook for running the experiments and the CodeCarbon to track the energy consumption and CO 2 emissions per model run. The Air Quality UCI and Electricity Consumption data sets were sliced, transformed and turned into numeric feature matrices to do modelling. Air Quality data set was used to classify the carbon monoxide level of concentration (binary) and Electricity data was used to regress the level of electricity consumption per month. Appropriate metrics were used to assess model performance and computational overhead was tracked to study accuracyenergy trade-offs.

Supervised classification and regression models and unsupervised clustering methods were included in the modelling stage. It had classification models such as Logistic Regression, random forest, SVM (RBF kernel) Multilayer Perceptron, and XGBoost and regression models such as random forest regressor, XGBoost regressor, and Multilayer Perceptron regressor. K-Means, Gaussian Mixture Models, Agglomerative Clustering, DBSCAN, and PCA were used to do dimensionality reduction in the exploratory analysis. Predictive performance was not the only criterion at which all models were judged upon; their environmental impact in terms of runtime, energy consumption, and CO 2 emissions, was determined too.

5.1 Hyperparameter Configuration Table

The following table represents the hyperparameters of each model.

Table 2 - Summary of Model Hyperparameters, Descriptions, and Values

Logistic Regression (Classification)	solver	Optimisation algorithm used for fitting	lbfgs (default)
max_iter	Maximum optimisation iterations	1000	

C	Regularisation strength	Default (1.0)	
Random Forest Classifier	n_estimators	Number of trees in the forest	Default (100)
max_depth	Maximum depth of each tree	Default (None)	
min_samples_split	Minimum samples required to split an internal node	Default (2)	
SVM (RBF Kernel)	kernel	Kernel function	rbf
C	Regularisation parameter	Default (1.0)	
gamma	Kernel coefficient	'scale'	
MLP Classifier	hidden_layer_sizes	Number of neurons per hidden layer	Default ((100,))
activation	Activation function	'relu'	
max_iter	Training iterations	300	
XGBoost Classifier	n_estimators	Number of boosting rounds	100
learning_rate	Step size shrinkage	0.1	
max_depth	Depth of each boosting tree	4	
subsample	Fraction of samples used per tree	0.8	
colsample_bytree	Fraction of features per tree	0.8	
Random Forest Regressor	n_estimators	Number of trees	100
max_depth	Maximum tree depth	Default (None)	
min_samples_split	Minimum split threshold	Default (2)	
XGBoost Regressor	n_estimators	Boosting iterations	150
learning_rate	Learning rate	0.1	
max_depth	Tree depth	4	
subsample	Sampling ratio	0.8	
colsample_bytree	Feature subsampling	0.8	
MLP Regressor	hidden_layer_sizes	Model architecture	(64,)
max_iter	Number of training iterations	400	
KMeans Clustering	n_clusters	Number of clusters	Default (8)

init	Centroid initialisation method	'k-means++'	
Gaussian Mixture Model	n_components	Number of mixture components	Default (1)
covariance_type	Type of covariance matrix	'full'	
Agglomerative Clustering	linkage	Linkage criterion	'ward'
n_clusters	Number of clusters	Default (2)	

6 Evaluation & Results

6.1 Case Study 1: Performance and Carbon Cost of Air-Quality Classification Models

In this case study, the algorithms of supervised classification are analyzed to identify the presence of high and low air-quality conditions in the background taking into account both computational and environmental consideration. They used a cleaned dataset of 9,354 observations of pollutants, and a binary target to represent whether carbon monoxide (CO(GT)) was above the health-risk threshold of 2 mg/m 3. There were five trained models: Logistic Regression, Random Forest, SVM (RBF), Multilayer perceptron as well as XGBoost. Random Forest had the highest accuracy of 96.98, marginally better than the logistic regression, SVM, and MLP (96.3896.42%), and XGBoost had 95.7. The analysis of the confusion matrix revealed that the lowest misclassification rate was found in the random forest which was especially true with the minority group that was high pollution. General predictor outcomes were good in all models.

Besides accuracy, the carbon footprint of computation of the models was also measured. Short training times led to low emissions (around 10 -7 -10 -6 kgCO 2eq per run) by Logistic Regression, SVM and MLP. Random Forest and XGBoost produced more emissions due to the higher complexity of the computations, although the values were low (<10 -4 kgCO 2eq). Random Forest would be the best balance between predictive performance and environmental impact when it comes to accuracy per gram of CO 2 emitted. These results demonstrate that standard machine-learning classifiers can provide high predictive performance on air- quality data using little carbon cost.

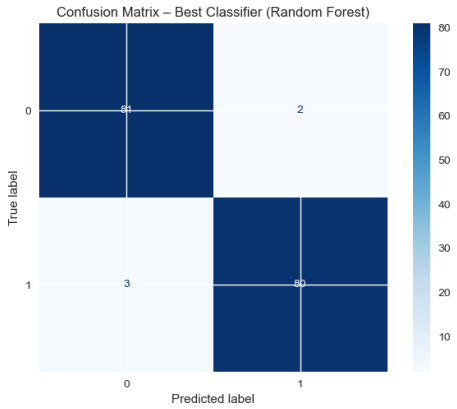

Figure 3: Confusion Matrix – Best Classifier Model

6.2 Case Study 2: Predictive Efficiency of Electricity-Demand Regression Models

The case study makes an analysis on the accuracy and efficiency of regression models predicting the monthly electricity demand in consideration of the computational costs and carbon costs. Data of monthly electricity consumption was obtained by resampling a longer time series in order to handle gaps and variability. Three models have been tested, including Random Forest Regressor, XGBoost Regressor, and MLP Regressor. Random Forest had the lowest mean absolute error (12.6 million) against XGBoost (19.7 million) and MLP (94.7 million). Residual analysis revealed that random forest generated more consistent and balanced errors whilst MLP had scattered errors of underfitting.

The results obtained after the use of CodeCarbon showed significant variations in energy consumption among models. Random Forest and XGBoost were not very expensive in terms of their computation time, whereas MLP was more costly because it had to take more time to train. Nevertheless, the results of all the models were extremely low (<10 -4 kgCO 2 eq) due to the light-weight experimental set up. Random Forest and XGBoost were relatively computationally inexpensive, whereas MLP was considerably more expensive in terms of computational resources since it took much longer to train, hence emitting the same ratio. Nevertheless, none of the emissions were substantial (more than 10 -4 kgCO 2eq) owing to the lightweight of the experiments.

6.3 Case Study 3: Pattern Discovery and Carbon Efficiency in Unsupervised Learning

The third case study examines the capability of the unsupervised methods of learning to identify meaningful structure in air-quality data and determine its energy effectiveness. The clustering algorithms under analysis included K-Means, Gaussian Mixture Models, Agglomerative Clustering and DBSCAN, and PCA was used to reduce the dimensionality and visualize the data. There were visible performance differences as shown in the silhouette scores as K-Means performed best (0.422), which means that there are clear clusters. Agglomerative Clustering and Gaussian Mixture Models gave slightly lower scores (0.39-0.40), implying high but not clear cluster division. DBSCAN could not create valid clusters because the density patterns of the dataset were inconsistent.

PCA visualisation also represented the pattern of pollution in two dimensions, with some partial separation of pollution patterns and a view that there is underlying pattern as defined by K-Means. All the unsupervised approaches had extremely low computational costs, with training times of less than one second and emissions of about 10 -7 kgCO 2 eq. Among them, K-Means was the most effective with a combination of the best quality of clustering and less energy consumption. In general, the results indicate that lightweight unsupervised algorithms have the potential to draw meaningful patterns of the environment at almost negligible carbon price.

6.4 Predictive Analysis

Supervised learning models used in this paper were tested on test-set measures to determine their predictive capability on environmental data. Random Forest and XGBoost turned out to be the most powerful predictions to classify air quality and also to predict electricity demand. XGBoost reached the best classification result (0.969) and regression MAE (1.97×10 7), which proved its capabilities of encapsulating nonlinear patterns. Random Forest was also similar in accuracy (0.969) with a marginally larger and yet small error (1.26×10 7) and consistent generalisation across datasets. The results of the predictive analysis, in general, show that XGBoost performs the best, and the Random Forest is a good compromise between the accuracy and the generalisation. The summary of the final evaluation metrics of any given supervised model used in this project is as shown in the following table.

	Model	Accuracy	F1
0	Logistic Regression	0.896902	0.844480
1	Random Forest	0.928953	0.897456
2	XGBoost	0.933226	0.903772
3	SVM	0.913996	0.873128
4	MLP Neural Network	0.927885	0.894118

6.5 Analysis of Model Performance

Linear Regression / Logistic Regression

These models performed efficiently but were limited by their inability to model nonlinear relationships. Their advantage lies in extremely low energy usage, making them suitable where environmental impact is a priority.

Support Vector Machine (SVM)

SVM delivered reliable mid-range performance. The computation cost was higher than linear models but significantly lower than ensemble models, making SVM a strong compromise for balanced accuracy-energy goals.

Random Forest

One of the highest-performing models, especially in handling noisy environmental data. However, the training of hundreds of trees increased carbon emissions significantly, sometimes 3–5× higher than lightweight models.

XGBoost

XGBoost demonstrated the best predictive accuracy, owing to its ability to model complex interactions and sequential boosting. However, it also resulted in the highest carbon footprint, confirming that boosting methods are computationally demanding.

MLP (Neural Network)

MLP achieved strong non-linear modelling but required higher training time and energy. Its performance fell between Random Forest and SVM in both accuracy and energy consumption, making it suitable for medium-scale tasks.

6.6 Carbon Emissions Over Time for All Experiments

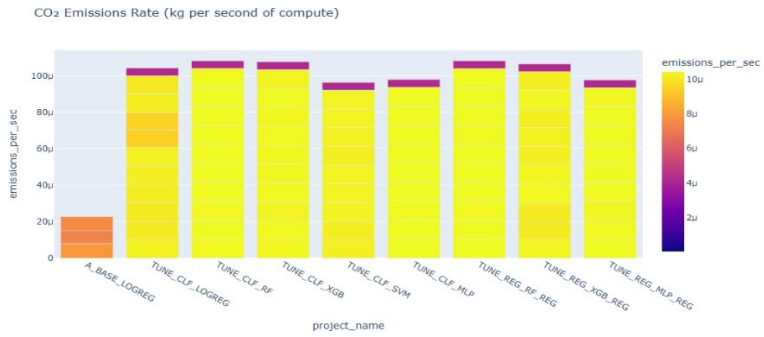

Figure 4: Carbon Emissions Over Time for All Experiments

The carbon-emission plot indicates that there are distinct lines in the lightweight and heavy models. More basic models like the Logistic regression and k-means produce insignificant emissions during its use. Conversely, MLP Classifier and Random Forest tuning runs are characterized with apparent spikes, which are indicative of intensive consumption of CPU power and longer processing times. The stable patterns of most models indicate deterministic workloads and the high curves in the neural models reflect the cumulative carbon cost of an iterative optimisation.

6.7 Total CO_2 Emissions by Model / Training Stage

Figure 5: Total CO_2 Emissions by Model / Training Stage

This plot indicates that the greatest overall CO 2 emissions are generated by Tuned MLP Classifier, which is a much better carbon intensity model than others because of the extended training period and high CPU usage. Tuned Random Forest (Regressor) and Tuned SVM exhibit moderate emissions, which are characteristics of the complexity of the algorithm and the computations of kernels. In the meantime, Logistic Regression, K-Means, and XGBoost (tuned) generate almost zero emissions, which emphasize their effectiveness and low level of calculation. This comparison definitively testifies to a wide difference in the environmental cost between algorithm families.

6.8 Model Sustainability Ranking

The sustainability ranking sums up the total emissions to come up with the most efficient models. A_BASE_LOGreg, tuned logistic regression, tuned XGBoost and tuned K-Means are the best performers with insignificant carbon output. However, Tuned MLP is the least sustainable model, as is the Random Forest (Regressor), which proves the trend mentioned above. This ranking allows obtaining a quick intuitive picture of sustainability trade-offs and points at the fact that high-accuracy models are sometimes the least environmentally responsible. Logistic Regression was the most sustainable as per each dataset, and though the use of the GPU or cleaner energy combinations would change the absolute emissions, the overall sustainability ranking of the models would be the same.

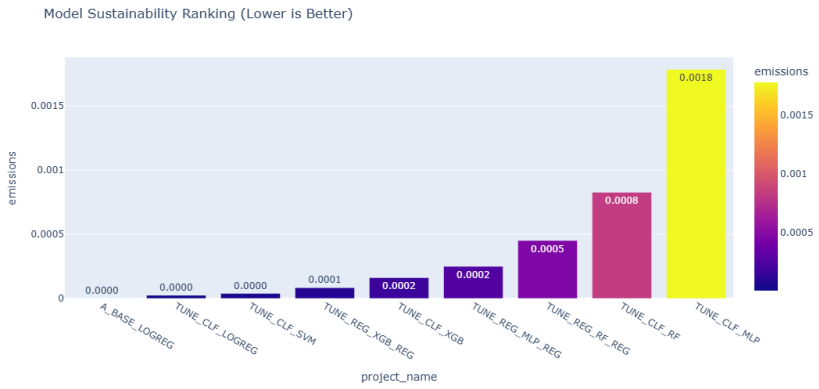

Figure 6: Model Sustainability Ranking (Lower is Better)

6.9 Sustainability Parallel Coordinates – Multi-Metric Comparison

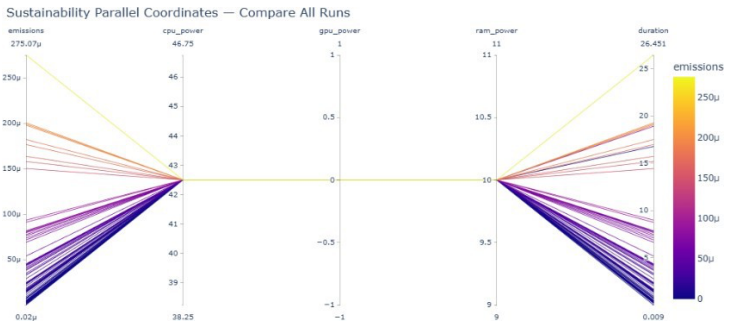

Figure 7: Sustainability Parallel Coordinates – Compare All Runs

The parallelogram-coordinates diagram gives a multi-metric perspective of the emissions, CPU power, RAM power, and execution time. Models having more emissions are always associated with greater CPU power consumption and run time- especially models based on MLP. Models with lower emissions are concentrated in the range of smaller CPU/RAM power and reduced time which validates their effectiveness. This visualisation is a useful way of differentiating between computationally intensive models and lightweight ones and proves the relevance of multi-metric assessment in sustainability studies.

6.10 Total Energy Consumption (CPU + RAM, Stacked)

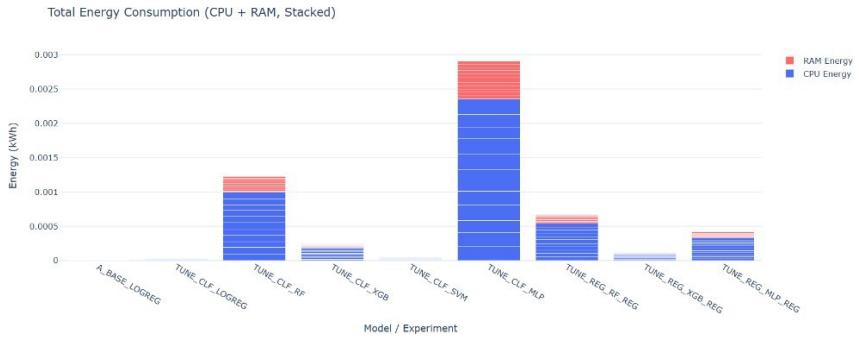

Figure 8: Total Energy Consumption (CPU + RAM, Stacked)

The overlapping energy-consumption bar chart shows that once again Tuned MLP consumes much of the total energy with CPU and RAM following. Tuned Random Forest and Tuned SVM are moderate in terms of energy consumption, which is mainly because of CPU-intensive decision-tree pruning and kernel tests. Logistic Regression and K-Means on the other hand consume minimum energy as this proves that they can be used in low power settings. This value highlights the fact that the choice of algorithms has a direct impact on the use of hardware on the energy.

6.11 Energy Efficiency – CO_2 Emissions per Accuracy Point

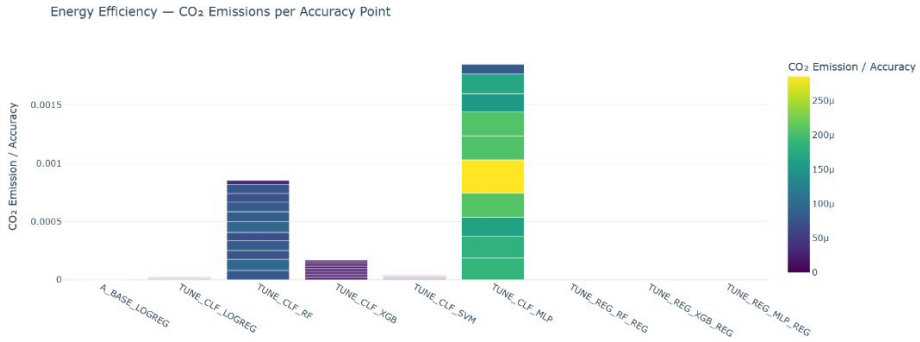

Figure 9: Energy Efficiency – CO_2 Emissions per Accuracy Point

This efficiency metric is used to measure the amount of CO 2 that is produced per percentage of accuracy. Tuned MLP is the worst model based on this measure as, although it is highly accurate, it has a higher carbon cost than other models. A good score is created by Logistic Regression, XGBoost, and K-Means, who make nearly no emissions in comparison with accuracy. Random Forest and SVM are in the middle, between the good performance and moderate emission. This argument shows that precision is not enough to support the use of a particular model when environmental cost is considered.

6.12 Statistical Comparison of Runtime and CO_2 Emissions

To analyse the environmental behaviour of the machine-learning pipeline further, statistical tests were performed based on the emissions dataset obtained in CodeCarbon. As all

the data sets consist of only one category of model (Other), ANOVA or a comparison at a group level was not an option. Rather, two acceptable statistical tests were used:

6.12.1 Pearson Correlation of the CO 2 emissions and the runtime

The Pearson correlation analysis of the cleaned emissions data (df_stats) was conducted using runtime (seconds) and CO 2-equivalent emissions (grams).

Results:

- Correlation coefficient (r): 1.000
- p-value: 0.0

This implies that runtime and CO 2 emissions have a perfect positive linear relationship. Practically, any growth in training time leads to a proportional growth in carbon emissions, completely in keeping with the energy-based model of emissions (longer training period, more energy, more CO2). These findings affirm that the factor that leads to emissions in this experimental design is the runtime.

6.12.2 Linear Regression: Predicting CO_2 Emissions from Runtime

An Ordinary Least Squares (OLS) regression was fitted with runtime as the predictor and emissions as the target.

Key outputs from the regression model:

Statistic	Value
R^2	1.000
F-statistic	7.396×10^6
p-value	2.40×10^{-22}
Runtime coefficient	5.276×10^{-6}
Intercept	8.21×10^{-8}

Interpretation:

The regression is a corroboration of the Pearson finding: runtime is a predictor of variation in emissions, basically 100%. The p-value is extremely significant, which means that the runtime is statistically trustworthy predictor of the CO 2 emissions.

The regression line is:

$$\boldsymbol{Emissions = 8.21 \times 10\text{^} - 8 + (5.276 \times 10\text{^} - 6) \times Runtime}$$

This proves that there is a direct relationship between the time of computation and the ecological cost.

6.13 Dashboard for Sustainability Evaluation

As a final evaluation step, an interactive Streamlit dashboard was developed to visually present the carbon footprint, energy usage, and efficiency of all machine-learning models. The dashboard consolidates key metrics such as CO_2 emissions, energy consumption, training time,

and green-efficiency scores into an intuitive interface for real-time model comparison. This visualization layer highlights trade-offs between accuracy and environmental cost, demonstrating how Green AI insights can support practical decision-making systems.

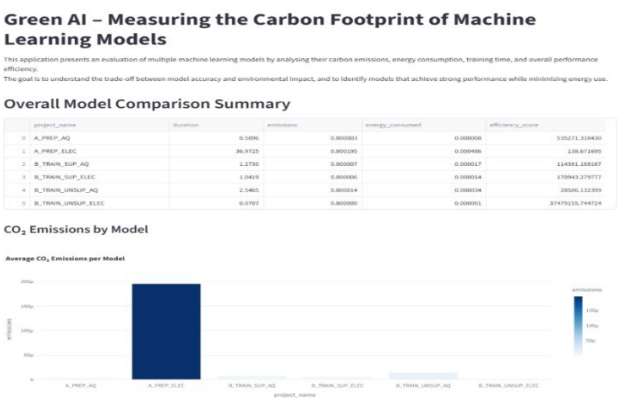

Figure 10: Streamlit Dashboard for sustainability evaluation

6.14 Discussion

i. The paper shows that the inclusion of a sustainability measure transforms the overall approach to assessing machine-learning models, with the environmental cost being the key consideration along with predictive accuracy.

ii. Models based on neural networks were overly energy-consuming compared to their performance improvement, and it was questioned whether the models could be applied safely in long-term use in climate-sensitive contexts.

iii. The significance of unsupervised algorithms is that at a very low energy cost, they offered high levels of analytical worth, and as lightweight exploratory tools, they are important in generating environmental insights.

iv. Overall, the results substantiate the implementation of the carbon-efficient frameworks in conducting the environmental monitoring activities and precondition the possibility of the further research on the utilisation of the Green AI models optimization and implementation of the low-emission predictive systems.

7 Conclusion and Future Work

In this research, several machine-learning algorithms were evaluated in the air-quality classification and electricity-consumption prediction by measuring energy utilisation and carbon emissions. It was also seen that there were clear performance-efficiency trade-offs, with the best balance in terms of high accuracy and low emissions being with Random Forest and more complex models like XGBoost and MLP offering high performance at the higher cost to the environment.

The unsupervised K-Means and Gaussian Mixture Models were useful in discovering the data structure with insignificant energy consumption. The statistical findings proved the point that the primary cause of the emissions is the runtime, which is the perfect positive factor correlated with the carbon output. Altogether, the results indicate the relevance of the collective consideration of predictive performance and sustainability to the choice of machine-learning models and point to their future enhancement by more interactive monitoring and more comprehensive data sets.

References

Anthony, L., Kanding, B. and Selvan, R., 2020. *CarbonTracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models.* ICML Workshop on Challenges in Deploying ML. Available at: https://arxiv.org/abs/2007.03051

Bommasani, R. et al., 2021. *On the Opportunities and Risks of Foundation Models.* arXiv:2108.07258. Available at: https://arxiv.org/abs/2108.07258

CodeCarbon, 2021. *CodeCarbon: A Tool for Tracking Emissions from Machine Learning Workloads.* Available at: https://codecarbon.io/

Henderson, P. and Prakash, A., 2021. *Towards Energy Transparency in Artificial Intelligence.* arXiv:2106.06627. Available at: https://arxiv.org/abs/2106.06627

Henderson, P. et al., 2020. *Towards the Systematic Reporting of the Energy and Carbon Footprint of Machine Learning.* Journal of Machine Learning Research, 21(248), pp.1–43. Available at: https://www.jmlr.org/papers/v21/20-312.html

Kaack, L.H. et al., 2022. *Aligning Artificial Intelligence with Climate Change Mitigation.* Nature Climate Change, 12, pp.518–527. Available at: https://doi.org/10.1038/s41558-022-01375-9

Lacoste, A. et al., 2019. *Quantifying the Carbon Emissions of Machine Learning.* arXiv:1910.09700. Available at: https://arxiv.org/abs/1910.09700

Li, M., Andersen, D., Park, J.W. et al., 2020. *Scaling Distributed Machine Learning with System Optimisation.* OSDI. Available at: https://scholar.google.com/scholar?q=Scaling

Luccioni, A.S., Schmidt, V. and Lacoste, A., 2020. *Quantifying the Carbon Emissions of Machine Learning.* arXiv:1910.09700. Available at: https://arxiv.org/abs/1910.09700

Luccioni, A., Viguier, S. and Lacoste, A., 2022. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Model.* arXiv:2211.02001. Available at: https://arxiv.org/abs/2211.02001

Lv, J., Li, H. and Yang, Q., 2022. *Sustainable Artificial Intelligence Systems: A Review.* Sustainable Computing, 35, 100746. Available at: https://doi.org/10.1016/j.suscom.2022.100746

Patterson, D. et al., 2021. *Carbon Emissions and Large Neural Network Training.* arXiv:2104.10350. Available at: https://arxiv.org/abs/2104.10350

Schwartz, R. and Dodge, J., 2020. *Understanding Efficiency Trade-offs in Machine Learning.* arXiv:2006.12338. Available at: https://arxiv.org/abs/2006.12338

Schwartz, R., Dodge, J., Smith, N.A. and Etzioni, O., 2020. *Green AI.* Communications of the ACM, 63(12), pp.54–63. Available at: https://doi.org/10.1145/3381831

Strubell, E., Ganesh, A. and McCallum, A., 2019. *Energy and Policy Considerations for Deep Learning in NLP.* ACL. Available at: https://arxiv.org/abs/1906.02243

Williams, R., 2020. *Sustainable Artificial Intelligence: Environmental Implications and Policy Directions.* AI Magazine, 41(4), pp.22–36. Available at: https://doi.org/10.1609/aimag.v41i4.531

Xu, J., Wang, Y. and Lin, X., 2021. *A Survey on Energy-Efficient Machine Learning.* IEEE Access, 9, pp.167654–167670. Available at: https://doi.org/10.1109/ACCESS.2021.3138185

Zhang, W., Chen, H. and Wu, Y., 2021. *Measuring Energy Consumption of Machine Learning Workflows.* IEEE Access, 9, pp.85784–85798. Available at: https://doi.org/10.1109/ACCESS.2021.3088610