

Article

Benchmarking Energy Efficiency of Supervised Machine Learning Models on Multi-Domain Classification Datasets

Aamir Ali ¹, Rohail Qamar ^{2,*}, Raheela Asif ² and Saman Hina ³
¹ Department of Global Data & Analytics, Aga Khan University, Karachi 74800, Pakistan

² Department of Computer Science & Information Technology, NED University of Engineering & Technology, Karachi 75270, Pakistan

³ Department of Computing, Imperial College London, London SW7 2AZ, UK

* Correspondence: rohailqamar@cloud.neduet.edu.pk

Abstract

Machine learning should be judged by how well it predicts, and computational resources are not accounted for in predictive accuracy. Given the growing emphasis on energy consumption and resource efficiency, decision-supporting frameworks should go beyond accuracy. This study presents an energy-based benchmarking approach for supervised learning models. Ten classical algorithms were evaluated on three textual and tabular datasets. The energy consumption of preprocessing, training, and inference was monitored with Intel RAPL via pyRAPL along with the runtime, peak memory usage, and predictive performance statistics (accuracy, precision, recall, F1-score, and AUC). Experiments were conducted in a controlled CPU-based environment to ensure comparability. The computational role of this feature is found to be appreciably diverse. Results show that Random Forest achieved the highest overall balance between predictive performance and efficiency (CI = 0.950, PPI = 0.907), while Logistic Regression provided a competitive trade-off (CI = 0.905, EI = 0.998). Gaussian Naïve Bayes was the most energy-efficient model with a mean energy consumption of 127 J, whereas Support Vector Classifier (SVC) incurred the highest computational cost, consuming 45,758 J and requiring 3925 s on average. The Pareto analysis identified Random Forest, Logistic Regression, Passive Aggressive, and Decision Tree as non-dominated solutions. These findings demonstrate that accuracy alone can be misleading for model evaluation and that integrating energy, runtime, and memory metrics enables more sustainable and resource-aware machine learning model selection. The proposed framework provides practical guidance for Green AI, Tiny Machine Learning (TinyML), edge computing, and other resource-constrained deployment environments.

Keywords: sustainable AI; machine learning; energy efficient; computational efficiency; model benchmarking; predictive analytics; explainable AI; Green AI; TinyML; pareto optimization

Academic Editor(s): Zhou Yang,
Fang Jin

Received: 14 May 2026

Revised: 29 June 2026

Accepted: 2 July 2026

Published: 4 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid expansion of information technology (IT) infrastructure has significantly increased its environmental impact. In 2020, the IT sector was responsible for approximately 2–6% of global carbon dioxide (CO₂) emissions, and projections suggest that this share may rise to nearly 20% by 2030 [1]. According to the International Energy Agency

(IEA), data centers consumed around 200 terawatt-hours (TWh) of electricity in 2018, accounting for nearly 1% of global electricity demand and contributing approximately 0.3% of worldwide CO₂ emissions [2]. Recent studies further indicate that the carbon footprint of digital technologies could grow two to nine times over the next decade, while electricity consumption may surpass 974 TWh by 2030 [3,4]. In addition, cryptocurrency mining has emerged as a major energy-intensive activity, consuming nearly 70 TWh of electricity annually by July 2021 through specialized mining facilities and hardware [5]. These trends highlight the urgent need to improve energy efficiency in computing systems and adopt sustainable computing practices to reduce the environmental impact of growing computational demands.

Energy consumption is another critical resource consideration in artificial intelligence systems, particularly during model training and deployment. Studies have shown that training a single large-scale AI model can generate carbon emissions comparable to those produced by several automobiles over their entire operational lifetimes [6]. This challenge becomes even more significant in edge computing environments, where energy efficiency is closely linked to the available computational resources. Unlike cloud-based infrastructures that typically have access to continuous power supplies, edge devices often rely on batteries and operate under strict energy constraints [7]. As a result, the energy requirements of AI models directly influence their feasibility for edge deployment. Deep learning architectures, especially Convolutional Neural Networks (CNNs), are known for their high computational complexity and substantial power consumption, which can restrict their practical use on resource-constrained devices [8]. Therefore, improving energy efficiency has become a key objective in the design and deployment of AI solutions for edge computing applications.

Sustainability refers to the concept of meeting present needs without compromising the ability of future generations to meet their own needs. It involves finding a balance between social, environmental, and economic considerations to ensure long-term well-being and harmony. Sustainable AI refers to the development and deployment of AI systems that are environmentally friendly and contribute to long-term sustainability. It involves considering the environmental, social, and economic impacts of AI technologies throughout their lifecycle. Sustainable AI is classified into two categories: Sustainability of AI and AI for Sustainability. Sustainability of AI deals with the extent to which the AI models are sustainable for the environment, e.g., reducing carbon footprints resulting from massive computations during model training/tuning. AI for Sustainability is a branch of sustainable AI making the environment sustainable by using AI models, e.g., AI4Good [9].

Computers pick up skills differently now through experience rather than fixed rules. That shift comes from a slice of artificial intelligence called machine learning. Patterns hidden in numbers get noticed by these smart programs without anyone spelling them out. Once spotted, such trends help shape guesses about what might happen next. Two decades have passed since this tech quietly moved into daily routines. Medical experts lean on it when spotting illness early. Financial institutions use its signals to flag odd transactions. Shopping sites suggest items based on past choices made elsewhere. Even city roads run smoother thanks to smarter movement tracking behind the scenes.

More computing muscle gets pulled in as machine learning spreads wider. How close predictions land near real outcomes shapes views on most training setups. Numbers such as precision, accuracy, recall, F1-score, and AUC step in to show how things perform. Their role is revealing how often guesses match actual values when fresh information arrives. Performance clarity comes from watching those scores shift across tests. Yet not every gain comes cheap. What gets measured typically misses the cost behind it. Performance numbers rarely show seconds burned or watts spent. Big data runs on heavy math,

needing powerful machines to keep up. Larger models demand more juice just to stay running. Power draw climbs as systems grow sharper and deeper.

These days, more scientists are looking into how AI affects the environment. People call this approach sustainable AI, creating systems that work well without wasting power or materials. Big machine learning setups, particularly ones hosted on remote servers, use massive amounts of electrical power and add to pollution levels. That reality pushes efficiency into focus, making speed, storage needs, and power draw key factors during model reviews. Right now, things are changing in how people see machine learning. Before, getting the right answer mattered most. Then came a time when doing tasks fast and with less work stood out. These days, it is about balancing correct results with how much power gets used across these systems [10,11].

1.1. Research Contribution

Although sustainability has received increasing attention in AI research, existing work on benchmark studies still focuses on predictive performance as the first priority of evaluation. Model comparison usually focuses on accuracy and its related metrics, at the expense of the computational cost as secondary or even omitted. Consequently, a slightly more accurate model can become an attractive alternative even if offered at much greater energy consumption.

- An energy-aware benchmarking framework is proposed for supervised machine learning models that jointly evaluates predictive performance, execution time, memory utilization, and energy consumption, moving beyond accuracy-centric evaluation.
- A comprehensive experimental evaluation of ten supervised machine learning algorithms is conducted across three multi-domain datasets (Fake News, Brain Stroke, and Diabetes) under a controlled CPU-based environment using Intel RAPL-based energy measurements.
- A multi-objective decision-making framework based on the Predictive Performance Index (PPI), Efficiency Index (EI), Composite Index (CI), and Pareto analysis is introduced to identify models that achieve the best balance between predictive effectiveness and computational efficiency.
- The results demonstrate that energy-efficient models can provide competitive predictive performance, with Random Forest achieving the highest overall balance (CI = 0.950), Logistic Regression offering near-optimal efficiency (EI = 0.998), and Gaussian Naïve Bayes exhibiting the lowest average energy consumption (127 J), highlighting the importance of sustainability-aware model selection for Green AI and TinyML applications.

1.2. Research Questions

RQ1: What does this research hope to uncover regarding machine learning model evaluation beyond accuracy? Traditional model evaluation in machine learning research has historically prioritized predictive performance, often treating computational cost as a secondary or omitted consideration. This narrow focus can mask significant trade-offs that matter in practical deployment, where execution time, memory usage, and energy consumption directly affect feasibility and sustainability. This research question establishes the motivating premise of this study: that a complete evaluation of supervised learning models requires considering computational efficiency alongside predictive accuracy, not as an afterthought but as a primary evaluation dimension.

RQ2: What differences exist in power consumption across various supervised machine learning algorithms? This research measures and compares the energy consumption of Decision Trees, Random Forests, Gaussian Naïve Bayes, Logistic Regression, K-Nearest

Neighbors (K-NN), Gradient Boosting, Passive Aggressive Classifier, Neural Networks, and Support Vector Classifier (SVC) using Intel RAPL and pyRAPL during preprocessing, training, and inference. Understanding these differences is important because energy consumption has become an increasingly significant factor in the deployment of machine learning systems, particularly in resource-constrained and environmentally sustainable computing environments.

RQ3: Could better predictions mean more power is used? Is there a relationship between predictive performance and energy consumption? This study examines whether improvements in predictive performance are associated with increased computational resource requirements. By analyzing the relationship between predictive metrics and computational cost, the research seeks to identify whether higher accuracy consistently requires greater energy consumption or whether efficient models can achieve competitive predictive performance. This has direct implications for model selection: if the relationship is strong, practitioners may need to explicitly trade off accuracy for efficiency; if weak or inconsistent, efficient models may be adopted without meaningful performance loss.

RQ4: Which machine learning models maintain strong predictive performance without excessive computational cost? This research question identifies the specific classifiers that achieve an effective balance between predictive performance (accuracy, precision, recall, F1-score, AUC) and computational efficiency (execution time, memory utilization, energy consumption). The objective is to move from establishing whether a trade-off exists to pinpointing which models occupy the most favorable position within it, providing concrete recommendations for deployment in computationally constrained or energy-sensitive environments.

RQ5: Could integrating predictive performance and computational efficiency provide a more effective benchmarking framework? This research proposes and evaluates a multi-objective benchmarking framework that integrates predictive performance, execution time, memory utilization, and energy consumption through the Predictive Performance Index (PPI), Efficiency Index (EI), Composite Index (CI), and Pareto analysis. The proposed framework aims to support more informed model selection by considering both predictive effectiveness and computational sustainability rather than relying solely on accuracy-based evaluation.

RQ1: What does this research hope to uncover regarding machine learning model evaluation beyond accuracy? It investigates supervised machine learning methods not only by accuracy but also by efficiency, including computation time and energy consumption, thereby assessing models across multiple real world demands and performance dimensions.

RQ2: What differences exist in power consumption across various supervised machine learning algorithms? We measure and compare the energy usage of different supervised learning algorithms, including Decision Trees, Random Forests, Gaussian Naïve Bayes (NB), Logistic Regression, K-Nearest Neighbors (K-NN), Gradient Boosting, Passive Aggressive Classifier, Neural Networks, and Support Vector Classifier, using tools like Intel RAPL and pyRAPL throughout preprocessing, training, and testing phases.

RQ3: Could better predictions mean more power used? Is there a relationship between model accuracy and energy consumption? We analyze whether higher predictive performance correlates with increased computational power usage, examining tradeoffs between model precision and resource consumption.

RQ4: Which systems remain accurate without consuming excessive power? We identify models that are able to reconcile goodness of prediction (accuracy, precision, recall, F1-score, AUC) and cost (runtime and energy), which shall provide a fee-free evaluation compared to the accuracy oriented testing.

RQ5: Could blending speed and power use provide a better way to judge models? We propose and implement a multi-objective benchmarking framework that combines runtime, energy consumption, and predictive performance, supporting energy-aware and efficient model selection.

2. Related Work

This literature review is organized around four areas. First, it surveys measurement techniques and tools used to quantify energy consumption in machine learning, including on chip counters, external meters, and predictive power models. As sustainable AI is becoming an increasingly prominent area of research, it is important to note the current gaps in the field. Although there are methods for measuring energy consumption, the majority of the current methods have focused on deep learning techniques and GPU-based pipelines. There is a lack of multi-domain dataset analysis of classical models. In addition, the majority of current methods require some form of calibration and can have variability across hardware. Secondly, we look at current environmental characterization techniques in the ML community. This study focuses on benchmarking methods and required reporting standards, using power benchmarking as a case study along with first results obtained using new toolkits. Third, we provide empirical insights into runtime, memory usage, and the trade-offs between accuracy and energy for a suite of commonly used classical machine learning methods that are implemented and deployed in practice. Finally, we present an in-depth study of how data preprocessing, feature engineering, anonymization, and dataset scale affect energy consumption. This study investigates the performance of ten different supervised learning methods on three benchmark data sets. All experiments were run entirely on the CPU, as well as including an analysis of the energy used by the machine for data preprocessing, model training, and model usage. A multi-objective approach is then used to evaluate the results based on accuracy and efficiency.

Machine learning models are commonly evaluated through predictive performance measures such as accuracy, precision, recall, and the F1-score. There are many metrics that have been studied in order to model how well a model performs. However, there are other metrics that are important, such as energy usage, execution time, and memory usage. These variations, even for models with similar accuracy, are significant to consider in practice. As prior work has shown, the variations in energy consumption can differ by orders of magnitude even for models with similar accuracy and are dependent on a variety of factors such as algorithmic choices and solvers, dataset size and complexity, and the underlying hardware configuration [10,12].

The environmental impact of AI, in terms of both energy consumption and emitted CO₂, is increasingly studied. Energy-aware benchmarking has become a standard part of the machine learning researcher's workflow and is highly relevant in practice. This includes everything from Internet of Things devices and embedded systems to clinical environments, where limited power budgets and non-trivial costs play a significant role in deciding which models are applicable. The environmental implications of AI have recently come into focus due to the electricity consumed by computations and the increasing levels of CO₂ emissions associated with them [13,14].

Energy-aware benchmarking of AI has therefore become an important aspect of responsibly conducting research. Energy usage is a key consideration in the real-world deployment of AI, from IoT devices to embedded systems and clinical environments, with power budgets and tangible costs driving choices around which models are viable to use [11,15,16]. There are many learning models that can be used in classification problems. Some models, like logistic regression and naive Bayes, have the simplicity to get relatively good accuracy in classification problems and consume less energy in training compared to tree-based methods, distance-based methods, and ensemble methods such as [17,18].

Commonly used approaches for measuring energy consumption in machine learning include external power meters, on-chip counters like Intel's Running Average Power Limit (RAPL), and predictive models based on hardware performance counters. External meters are reliable system-level measures but require additional hardware and lack component-level granularity. Predictive models are adaptable but need to be calibrated and may differ from one platform to another. For CPU-based classical machine learning, there is some RAPL that was used as a standard due to low overhead and the ability to measure energy domains of CPU and DRAM [19–21]. This study demonstrates the application of controlled experimental conditions to ensure repeatability and uses RAPL to measure energy consumption during preprocessing, training, and testing following established methodological guidelines [22,23].

The widening interest in the energy behavior of machine learning systems has resulted in a number of benchmarking frameworks. These include Eco2AI, which tracks energy use and emissions in an effort to target research environments. Others focus on better reproducibility through specified reporting and experimental guidelines. MLPerf Power offers standardized rules for measuring and comparing energy usage across hardware platforms and workloads at industrial scale [22–24]. While they have different goals, these frameworks share common principles such as explicit workload definition, controlled hardware and software environments, warm-up runs, and transparent reporting.

But the research landscape is still not very clear. A lot of studies leave out important information that is needed for accurate energy measurement, like CPU frequency settings, governor configurations, background process isolation, or sampling rates that are high enough. Also, the size, sparsity, dimensionality, and preprocessing of the dataset all have a big effect on how much energy is used. Consequently, results from GPU-based deep learning benchmarks cannot be directly applied to CPU-based classical models or to datasets of varying modalities, such as text compared to tabular data [25–27]. These problems show that we need to make standards stronger so that reported results show how algorithms really work instead of how they were tested.

The study tackles these concerns by strictly following the rules for measuring things. To make sure that the reported performance and energy results are repeatable, clear, and comparable and that they show real algorithmic behavior instead of changes in the environment or method, the experimental scope is carefully controlled.

2.1 Sustainable AI and Carbon Accounting

Energy consumption and carbon reporting have become crucial in machine learning, as they expand model assessment from mere predictive accuracy to encompass environmental consequences. Research has shown that training and deployment can release a lot of carbon dioxide, but these costs are often ignored when the focus is only on accuracy or error rates [28,29]. Research further indicates that algorithm selection, hardware configuration, and preprocessing choices can substantially affect total energy consumption, underscoring the necessity for transparent reporting. Tools like Eco2AI help with this by keeping track of how much energy is used and estimating how much CO₂ is released using regional electricity profiles [30]. This is part of a larger move toward sustainable AI.

The impact of carbon cannot be figured out just by looking at energy use, since the carbon intensity of electricity changes from place to place and time to time, which means that the same amount of energy use can have different CO₂ outcomes. Also, operational energy is only a small part of a model's total footprint, and emissions from data center infrastructure and hardware manufacturing are often left out of real-world studies [31,32]. This research tackles these problems by using regional emission factors to turn CPU and DRAM energy measurements into CO₂ equivalents and reporting emissions separately for preprocessing, training, and inference. The method makes sure that traditional

machine learning models can be evaluated in a clear, repeatable, and comparable way by recording raw energy data, hardware settings, and emission factors.

2.2 Energy Efficiency of ML Algorithms and Architectures

This current work examines the performance of ten classical supervised learning algorithms, namely Logistic Regression, Gaussian NB, K-NN, Support Vector Classifiers, Decision Trees, Random Forests, Gradient Boosting, the Passive Aggressive Classifier, and two multi-layer perceptron models. These algorithms are tested on a variety of datasets to explore the correlation between prediction accuracy, computational complexity, and energy consumption. From existing literature, there is a large variation in the efficiency of classical algorithms, with linear models like Logistic Regression and Naïve Bayes being the most energy-efficient and distance-based and ensemble models like K-NN, Random Forests, and Gradient Boosting being less energy-efficient due to their higher memory requirements and computational complexity [17,18]. Research studies targeting CPU-centric settings have shown that traditional algorithms, such as linear models and boosted decision trees, can often outperform neural networks on structured tabular data while consuming significantly less power, thus making them applicable in resource-limited scenarios [26,27]. Ensemble methods typically provide strong predictive performance; on the other hand, they require higher computational costs and energy consumption, although they remain much more efficient than deep learning models, which consume significantly more power during training and inference [33].

Neuromorphic and spiking neural networks are also not considered in this CPU-only benchmark, as they are normally dependent on specialized hardware. Meanwhile existing works have shown that these networks can significantly lower the energy consumption of inference tasks using sparse and event-driven processing [34]. Our study focuses on classical machine learning algorithms running on standard CPUs, where the algorithmic complexity, dataset, and workload design have a direct and measurable impact on energy consumption. Complementary research proposes metrics such as EFLOP, which incorporate sparsity-aware operation counts to relate model structure more directly to computational cost in both spiking and conventional neural systems [35].

2.3 Data and Software Centric Optimization

The decisions made and enforced through data preparation and software configuration have a great impact on the energy consumption of machine learning tasks. Existing work has shown that changes to the dataset itself can lead to a dramatic reduction in energy consumption, with a reduction in dataset or feature size leading to a 90% reduction in energy consumption while maintaining similar accuracy [26]. Similarly, the use of k-anonymization has been shown to lead to a reduction in training energy for models such as Logistic Regression and Random Forests, with greater benefits as the dimensionality of the features increases [27]. Hyperparameter optimization is also important, as it has been shown that Bayesian optimization and Tree-Structured Parzen Estimators can be used to find hyperparameters that lead to faster convergence. For example, hyperparameter optimization using TPE resulted in a 44% reduction in carbon emissions for Random Forests and Support Vector Classifiers [36]. While this research does not propose dataset modification or hyperparameter optimization, existing work highlights the significance of these approaches in improving energy efficiency.

Software-level choices are essential in realizing fair, repeatable, and significant energy measurements. The literature emphasizes the need to manage CPU frequency, limit background processes, manage library versions, and normalize preprocessing to address variability in runtime, memory, and energy consumption [19,21]. Reproducibility best practices also emphasize the need for deterministic execution, detailed configuration

logging, and clear system documentation [22]. Following these best practices, this research uses fixed random seeds, normalized preprocessing, managed software versions, and CPU isolation to ensure that the measured energy differences are indicative of algorithmic characteristics and not influenced by system variables.

Collectively, these studies demonstrate increasing awareness of sustainability considerations in machine learning. However, most existing works focus on specific applications, individual datasets, deep learning architectures, or isolated phases of the machine learning pipeline. Comprehensive evaluations that simultaneously analyze preprocessing, training, and inference energy consumption together with predictive performance, runtime, and memory utilization across multiple heterogeneous datasets remain relatively limited (see Table 1).

Table 1. Comparative Review of Existing Green AI Studies.

Study	Application	Dataset	Methodology	Key Findings	Limitation
Liu and Yin (2024) [14]	LLM Carbon Footprint	LLM training benchmarks	Comparative carbon footprint analysis across hardware configurations	Hardware selection and training configuration are the dominant factors in total carbon emissions	Restricted to LLM training; inference and classical ML excluded
Qamar et al. (2026) [11]	Edge AI Benchmarking	Edge device benchmark datasets	Deep neural network architecture comparison on resource-constrained edge hardware	Significant energy efficiency differences observed among architectures with similar predictive accuracy	Restricted to deep learning models; classical ML
Santos et al. (2024) [17]	ML Energy Benchmarking	3 binary benchmark datasets	Six classical ML algorithms with energy monitoring via CodeCarbon	Random Forest achieved highest accuracy; Logistic Regression demonstrated best energy efficiency across all datasets	Limited to binary classification; single CPU hardware platform
Zemanek et al. (2024) [27]	Privacy-Preserving ML	UCI repository datasets	K-anonymization applied to Random Forest, Logistic Regression, and K-NN with pyRAPL energy tracking	Anonymization reduced training energy significantly, especially for Logistic Regression with high feature dimensionality	Limited algorithm variety; single anonymization technique evaluated
Zhang et al. (2024) [28]	NLP Energy Efficiency	GLUE benchmark	Spiking transformer architecture (Spiking-MiniLM)	Achieved comparable NLP accuracy to standard transformers at substantially lower energy cost	Requires specialized neuromorphic hardware; not applicable to standard CPUs
Knipper et al. (2024) [34]	NLP via Spiking Networks	Sentiment analysis datasets	Spiking Neural Networks (SNNLP) for NLP tasks	Achieved up to 32× inference energy reduction compared to conventional neural networks	Neuromorphic hardware-dependent; limited to NLP tasks
Narduzzi et al. (2025) [35]	Computational Cost Metrics	SHD and SSC neuromorphic datasets	EFLOP sparsity-aware metric for spiking and non-spiking networks	Sparsity-aware optimization reduces computational cost	Metric designed for neural networks;

				without sacrificing predictive accuracy	not applicable to classical ML
Rozam et al. (2025) [36]	Network Intrusion Detection	SDN-based DDoS datasets	Bayesian Optimization with carbon emission tracking via CodeCarbon	Bayesian Optimization reduced carbon emissions by approximately 44% while maintaining detection accuracy	Single security domain; generalizability to other tasks unvalidated

2.4. Research Gap

There are some limitations in the existing literature that this study attempts to fill. Although there is a rising interest in energy-efficient machine learning, the existing literature still largely focuses on deep learning and GPU-related workloads, and CPU-related studies on traditional supervised learning models are still relatively limited [18,19]. Even for traditional models, most studies focus on a very small set of models, a single dataset, and heterogeneous and poorly documented energy consumption methodologies. Moreover, most studies focus on either predictive accuracy, runtime, memory usage, or energy consumption, but very few studies simultaneously consider all these factors, although there have been many appeals to extend the model evaluation criteria beyond accuracy [13,35]. Cross-domain benchmarking is also limited, as most studies focus on either text or tabular data, but not both.

The reporting done in CPU-only energy benchmarking adds more complexity. Some research papers lack critical information regarding the experimental setup, including hardware information, CPU governor settings, sampling rate, preprocessing energy cost, and the use of fixed random seeds, making it difficult to reproduce and adding to the variation in the values of energy consumption reported in the literature [20,22]. Although efforts such as MLPerf Power encourage standardized reporting and methodology, this has not yet trickled down into traditional machine learning research communities [24]. More variation comes from the use of invalidated on-chip sensors or energy prediction models, which may introduce large errors [25].

It is clear from the literature that the use of robust tools such as Intel RAPL, together with best practice guidelines for benchmarking and carbon accounting, forms a good foundation for comparison. However, existing research also shows that data preparation, feature choice, hyperparameters, and software configuration can affect energy consumption even when the algorithm itself is unchanged. This study proposes a systematic and clear method for comparing accuracy, energy consumption, runtime, and memory usage for a range of algorithms and datasets.

3. Materials and Methods

3.1. Experimental Framework

The experimental framework developed for this study follows a clear and transparent process for evaluating supervised machine learning models with respect to both their predictive performance and their computational efficiency. The process starts with choosing the datasets and then moves through a set of planned steps that allow each model to be tested in a fair and uniform manner. Organizing the assessment in this structure helps the study produce results that can be checked again and understood as representing how the algorithms actually behave. All experiments are performed on a CPU-based system so that execution time, memory behavior, and energy usage can be measured precisely under uniform conditions.

The remaining steps are performed in the same manner for all models, but preprocessing is handled slightly differently for text and tabular data because of the different

datasets used in the experiment. This makes it more likely that any variations in performance are caused by the models themselves rather than by modifications to the way the experiments were conducted. The entire process is managed by a single benchmarking script, which ensures that each algorithm handles data, trains, and draws conclusions in precisely the same manner.

Performance is evaluated using accuracy, precision, recall, F1 score, and AUC, while efficiency is measured through execution time, memory utilization, and CPU and DRAM energy usage. Energy readings are obtained through the pyRAPL (Ver. 0.2.3.1) library, and runtime and memory statistics are captured using Python's (Ver. 3.12.3) time and tracemalloc modules. The outcome of this framework is a comprehensive set of benchmarking results that summarizes each model's strengths and limitations, allowing clear observation of the tradeoffs between predictive capability and computational cost. Figure 1 illustrates the fully automated pipeline used to capture both predictive and efficiency metrics. The diagram shows dataset acquisition, preprocessing, instrumented execution with pyRAPL, time and tracemalloc, logging to CSV, and the multi-objective analysis producing Predictive Performance Index (PPI), Efficiency Index (EI), and Pareto frontier results.

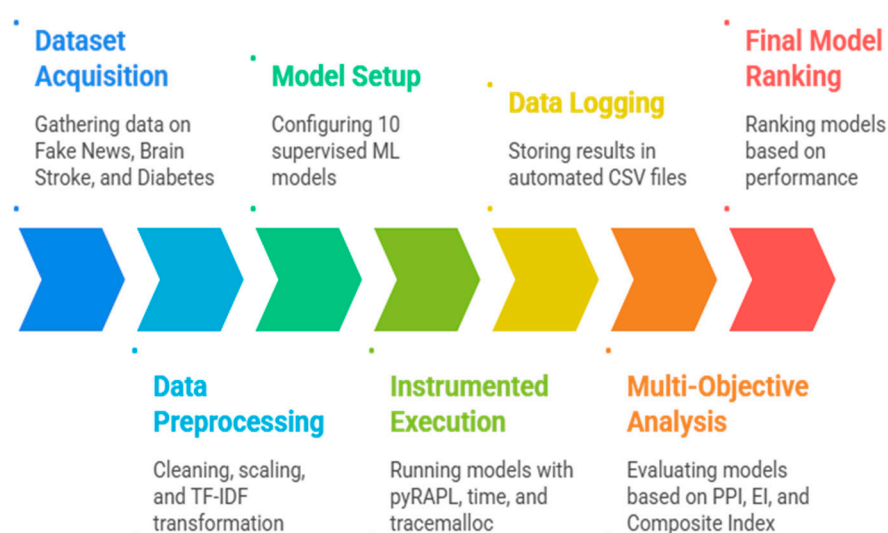


Figure 1. Framework for the CPU-based Benchmarking.

3.2. Dataset Selection

This study utilizes three diverse, publicly available datasets selected to represent different application domains and data modalities. The Fake News dataset belongs to the news and media domain and contains unstructured textual data, whereas the Brain Stroke and Diabetes datasets represent healthcare applications and contain structured tabular data. The selected datasets also vary considerably in scale, feature composition, and classification complexity, enabling a broader assessment of predictive performance and computational efficiency across heterogeneous machine learning workloads. The open-source data repository contains Fake News, Brain Stroke, and Diabetes, to evaluate machine learning models across different domains and data structures. By incorporating both unstructured text (news) and structured tabular data (healthcare), as well as both binary and multiclass classification tasks, the framework ensures a broad assessment of model behavior. The selected data ranges in scale from approximately 5000 to over 220,000 records, providing a robust foundation for observing how various computational loads impact

both predictive accuracy and resource efficiency. The attributes of the dataset used in this study are summarized in Table 2.

Table 2. Overview of Dataset.

Dataset	Domain	Rows	Features	Structure	Label
Fake News	News	6335	3	Text	0/1
Brain Stroke	Healthcare	4981	11	Tabular	0/1
Diabetes	Healthcare	229,781	22	Tabular	0/1/2

3.3. Data Preprocessing

Preprocessing was applied to all datasets to ensure they were in a suitable form for supervised learning while maintaining fairness in evaluation. A standard 80/20 training-test split was used for all experiments to provide a consistent basis for comparison.

Dataset 1 (Fake News): The text was cleaned of punctuation, tokenized, and filtered for stop words before being converted into numerical features using TF-IDF vectorization.

Dataset 2 (Brain Stroke): This tabular data was processed by handling missing values, encoding categorical attributes, and scaling numerical fields to maintain uniform feature ranges.

Dataset 3 (Diabetes): Similar to the stroke dataset, this large-scale data was cleaned of duplicates and standardized to ensure high data quality before model training.

3.4. Model Selection

Ten supervised machine learning models were selected to represent a diverse range of learning behaviors: Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Gaussian NB, K-NN, Gradient Boosting Classifier, Passive Aggressive Classifier, Support Vector Classifier, and two Neural Networks (one with a single hidden layer and one with five hidden layers). These models were implemented using Scikit-Learn with default hyperparameters to ensure the benchmarking results reflect the natural behavior of the algorithms rather than the effects of tuning (see Table A1). The detailed predictive and efficiency metrics are presented in Table 3.

Table 3. Predictive and Efficiency Metrics.

Indicator	Metric Notation
Accuracy	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$
	where
	• TP: True positive • TN: True negative • FP: False positive • FN: False negative
Precision	$P = \frac{TP}{(TP + FP)}$
Recall	$R = \frac{TP}{(TP + FN)}$
F1-Score	$F1 = \frac{2 \times P \times R}{P + R}$
Area Under Curve	AUC measures the model's ability to distinguish between classes across thresholds
Energy consumption	CPU and DRAM energy usage measured in Joules through pyRAPL interface

Memory Usage	Peak memory utilization in KB captured using the tracemalloc library
Execution Time	Total time in seconds for preprocessing, training, and inference measured via Python's time module.
Predictive Performance Index	$PPI = \frac{PPI_{raw} - \min(PPI_{raw})}{\max(PPI_{raw}) - \min(PPI_{raw})}$ Normalizes the average of all predictive metrics to a scale of 0 to 1
Resource Efficiency Components	$\text{Metric}_{eff} = 1 - \frac{\text{Metric} - \min(\text{Metric})}{\max(\text{Metric}) - \min(\text{Metric})}$ Inverts and normalizes resource usage so that higher values represent better efficiency (lower consumption)
Efficiency Index (EI)	$EI = \frac{\text{Time}_{eff} + \text{Memory}_{eff} + \text{Energy}_{eff}}{3}$ Provides a single normalized score representing the model's overall computational economy
Composite Index (CI)	$CI = \frac{PPI + EI}{2}$ Combines accuracy and efficiency into a final score to rank the most balanced model
Pareto Analysis	$PPI_A \geq PPI_B \text{ and } EI_A \geq EI_B$ Identifies “non-dominated” models that offer the best available trade-offs without being strictly inferior to others.

3.5. System Environment

Experiments were conducted on a controlled CPU-based system featuring an Intel Core i5 8th Generation processor with 16 GB RAM and Ubuntu 22.04 LTS. This environment ensures RAPL support for precise energy readings while avoiding the variables introduced by GPU acceleration. The experimental workflow follows a 7-step automated pipeline:

1. Dataset acquisition from an open source repository;
2. Data cleaning and formatting, i.e., TF-IDF for text, scaling for tabular;
3. Initializing the 10 models;
4. Sequential running of benchmarks;
5. Capturing predictive and efficiency data;
6. Logging results to CSV files;
7. Analysis and identifying Pareto-optimal models.

4. Analysis and Discussion

This section looks at ten machine learning models that are supervised. It checks how well they work on the Fake News, Brain Stroke and Diabetes datasets. It considers how well they can predict things and how efficient they are. The things it looks at are accuracy, precision, recall, and F1-score, and AUC for predictive performance. It also looks at execution time, memory usage, and energy consumption for efficiency. This shows that just because a model is very accurate, it does not mean it uses a lot of energy. The amount of energy a model uses depends on how complicated the algorithm is, how it uses memory, and how much computation it does.

The results are broken down into parts: how well each model does on each dataset, how efficient each model is, and how accuracy and resource usage balance out. The study also ranks the models based on things to find the ones that do well and use resources

efficiently. Figure 2 shows how the characteristics of a dataset, the architecture of a model, and the workload of computation are related. It shows that energy consumption is more about complexity and memory activity than accuracy. The main thing the current research finds is that some machine learning models can be very accurate and also use resources efficiently, which is good for AI and the Fake News, Brain Stroke and Diabetes datasets.

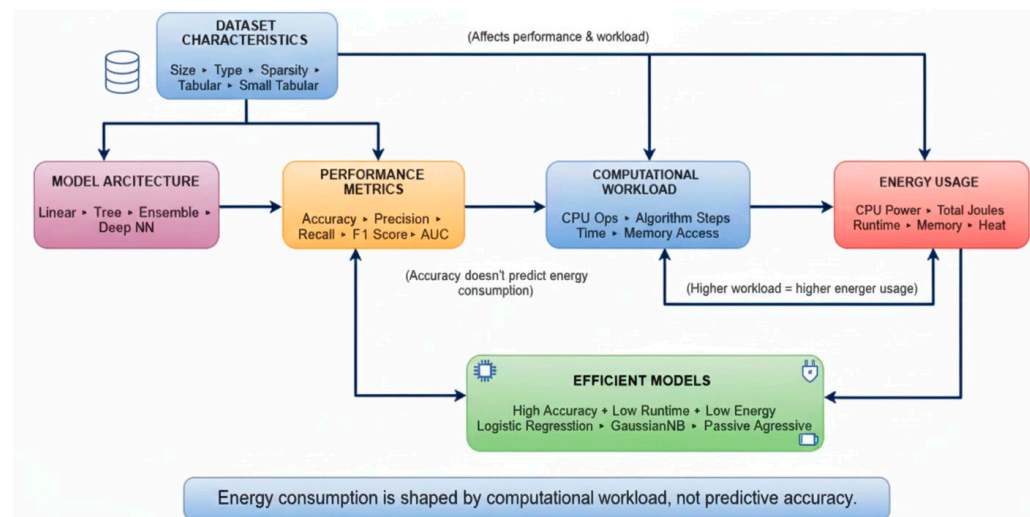


Figure 2. Conceptual Overview of Benchmarking Result Highlighting Performance and Efficiency Trade-offs.

4.1. Performance Matrices of Machine Learning Classifiers

The performance of the evaluated machine learning classifiers was assessed from both predictive and computational perspectives across the Fake News, Brain Stroke, and Diabetes datasets. Predictive effectiveness was measured using Accuracy, Precision, Recall, and Area Under the Curve (AUC), while computational efficiency was evaluated using energy consumption, execution time, memory utilization, and F1-Score. This combined evaluation provides a comprehensive understanding of the trade-offs between classification performance and resource utilization.

The predictive performance results are presented in Table 4. For the Fake News dataset, Random Forest, Support Vector Classifier (SVC), and the neural network with 5 hidden layers achieved the strongest classification performance, attaining approximately 94% accuracy, precision, and recall. Logistic Regression and Passive Aggressive also demonstrated competitive performance, achieving classification accuracies above 92%. These findings suggest that both ensemble learning methods and neural network architectures are highly effective for text-based fake news classification.

On the Brain Stroke dataset, Passive Aggressive achieved the highest accuracy (93%), followed closely by Logistic Regression (92%), Gradient Boosting (91%), and the neural network models (90–91%). Most classifiers achieved precision values above 90%, demonstrating strong capability in identifying stroke-related outcomes. In contrast, K-NN and SVC exhibited comparatively lower predictive performance on this dataset.

The Diabetes dataset proved more challenging than the other datasets, resulting in lower predictive performance across most classifiers. The neural network with 1 hidden layer achieved the highest accuracy (82%), followed by Logistic Regression (79%) and Decision Tree (77%). Although predictive performance declined relative to the other datasets, the results reveal that simpler machine learning models remain competitive for structured healthcare data.

Table 4. Predictive Performance for Fake News, Brain Stroke and Diabetes.

Classifier	Fake News				Brain Stroke				Diabetes			
	Accuracy	Precision	Recall	AUC	Accuracy	Precision	Recall	AUC	Accuracy	Precision	Recall	AUC
Decision Tree	0.80	0.80	0.80	0.80	0.89	0.91	0.89	0.55	0.77	0.79	0.77	0.76
Gaussian Naïve Bayes	0.86	0.87	0.86	0.94	0.86	0.92	0.86	0.69	0.60	0.79	0.60	0.69
Gradient Boosting	0.92	0.92	0.92	0.98	0.91	0.92	0.91	0.82	0.61	0.83	0.61	0.79
K-NN	0.88	0.88	0.88	0.91	0.57	0.93	0.57	0.74	0.48	0.84	0.48	0.76
Logistic Regression	0.92	0.92	0.92	0.97	0.92	0.91	0.92	0.79	0.79	0.80	0.79	0.79
NN 1 Hidden Layer	0.89	0.89	0.89	0.96	0.90	0.92	0.90	0.81	0.82	0.79	0.82	0.80
NN 5 Hidden Layers	0.94	0.94	0.94	0.98	0.91	0.93	0.91	0.81	0.62	0.83	0.62	0.78
Passive Aggressive	0.93	0.94	0.93	0.98	0.93	0.92	0.93	0.80	0.61	0.82	0.61	0.75
Random Forest	0.94	0.94	0.94	0.98	0.83	0.93	0.83	0.79	0.53	0.84	0.53	0.78
SVC	0.94	0.94	0.94	0.98	0.69	0.95	0.69	0.83	0.58	0.83	0.58	0.77

While predictive performance provides insight into classification effectiveness, practical deployment also requires consideration of computational efficiency. Table 5 summarizes the energy consumption, execution time, memory utilization, and F1-score achieved by each classifier across the three datasets. For the Fake News dataset, SVC achieved the highest F1-score (0.942), followed closely by the neural network with 1 hidden layer (0.940), Passive Aggressive (0.935), and the neural network with 5 hidden layers (0.934). However, SVC also incurred the highest computational cost, consuming 3431 J of energy, requiring 261 s of execution time, and utilizing 396,125 KB of memory. In contrast, Logistic Regression achieved a competitive F1-score of 0.921 while consuming only 340 J of energy and requiring 31 s of execution time.

On the Brain Stroke dataset, the neural network with 5 hidden layers achieved the highest F1-score (0.922), followed by Random Forest (0.918) and Logistic Regression (0.917). Although deep neural networks provided strong predictive performance, they required substantially greater computational resources. Logistic Regression achieved comparable predictive performance while consuming only 2 J of energy and requiring 0.19 s of execution time. For the Diabetes dataset, Gradient Boosting achieved the highest F1-score (0.807), followed by Random Forest (0.795) and Decision Tree (0.779). The neural network with 5 hidden layers and SVC demonstrated poor efficiency–performance balance, requiring substantial energy consumption and execution time while providing only moderate predictive gains. These findings illustrate that increasing model complexity does not necessarily translate into superior predictive performance.

Table 5. Efficiency Performance for Fake News, Brain Stroke and Diabetes.

Classifier	Fake News				Brain Stroke				Diabetes			
	Energy (J)	Time (s)	Mem (KB)	F1-Score	Energy (J)	Time (s)	Mem (KB)	F1-Score	Energy (J)	Time (s)	Mem (KB)	F1-Score
Decision Tree	387	36	99,322	0.796	2	0.21	5200	0.901	80	6	98,235	0.779
Gaussian Naïve Bayes	324	31	198,567	0.879	1.1	0.09	5015	0.682	56	4	130,965	0.583
Gradient Boosting	796	76	99,588	0.887	57	5	5059	0.913	3978	389	98,224	0.807
K-NN	338	31	247,838	0.859	3	0.14	6202	0.887	1110	33	130,966	0.662
Logistic Regression	340	31	199,765	0.921	2	0.19	5017	0.917	638	26	130,964	0.692
NN 1 Hidden Layer	1305	63	17,795	0.940	330	31	5015	0.917	1889	178	93,551	0.693
NN 5 Hidden Layers	650	41	29,195	0.934	1921	71	5482	0.922	71,726	3638	93,549	0.680
Passive Aggressive	331	32	198,225	0.935	3	0.24	4988	0.868	66	5	130,964	0.623
Random Forest	387	37	99,534	0.919	70	7	5127	0.918	1219	114	98,224	0.795

SVC	3431	261	396,125	0.942	415	37	5664	0.778	133,429	11,478	131,551	0.664
-----	------	-----	---------	-------	-----	----	------	-------	---------	--------	---------	-------

4.2. Efficiency Analysis

The mean efficiency metrics presented in Table 6 reveal substantial differences in computational resource utilization among the evaluated machine learning classifiers. The results demonstrate that lightweight models such as Gaussian Naïve Bayes, Passive Aggressive, Logistic Regression, and Decision Tree consistently require lower energy consumption, shorter execution times, and relatively modest memory resources compared with more computationally intensive approaches. Among all evaluated classifiers, Gaussian Naïve Bayes demonstrated the lowest average energy consumption (127 J) and one of the shortest execution times (12 s), although the predictive performance was comparatively lower than that of several other models. Similarly, Passive Aggressive achieved an average energy consumption of 134 J and an execution time of 12 s while maintaining competitive classification performance. Decision Tree and Logistic Regression also exhibited favorable efficiency characteristics, requiring only 156 J and 327 J of average energy consumption, respectively, while completing execution in less than 20 s on average. Furthermore, both models maintained relatively low memory utilization, making them suitable candidates for deployment in resource-constrained environments.

Table 6. Mean Accuracy and Mean Efficiency Metrics across Three Datasets.

Classifier	Mean Accuracy	Mean F1-Score	Mean Energy (J)	Mean Time (s)	Mean Memory (KB)
Decision Tree	0.818	0.826	156	14	2205
Gaussian Naïve Bayes	0.641	0.715	127	12	3070
Gradient Boosting	0.871	0.869	1610	157	2258
K-NN	0.772	0.803	483	21	91,428
Logistic Regression	0.815	0.843	327	19	2136
NN 1 Hidden Layer	0.822	0.850	1175	91	3602
NN 5 Hidden Layers	0.823	0.845	24,766	1250	6701
Passive Aggressive	0.765	0.809	134	12	3020
Random Forest	0.878	0.877	559	53	2321
SVC	0.736	0.795	45,758	3925	62,389

K-NN required a moderate amount of energy (483 J) and execution time (21 s); however, it exhibited the highest memory requirement among the traditional machine learning algorithms, consuming an average of 91,428 KB of memory. This behavior is expected because K-NN stores the entire training dataset and performs instance-based comparisons during prediction. Random Forest achieved the highest mean accuracy (0.878) and mean F1-score (0.877) among all evaluated classifiers while maintaining moderate computational requirements, consuming 559 J of energy and requiring 53 s of execution time on average. Gradient Boosting also demonstrated strong predictive performance, achieving a mean F1-score of 0.869, but at the expense of higher computational cost, with an average energy consumption of 1610 J and an execution time of 157 s.

The neural network models required substantially greater computational resources. The neural network with 1 hidden layer consumed an average of 1175 J of energy and required 91 s of execution time, whereas the neural network with 5 Hidden Layers consumed 24,766 J of energy and required 1250 s of execution time. Although both neural network architectures achieved competitive predictive performance, the considerable increase in computational cost highlights the trade-off between model complexity and resource efficiency. SVC was identified as the most computationally expensive classifier. It consumed an average of 45,758 J of energy and required 3925 s of execution time, while

also exhibiting the highest memory utilization (62,389 KB). Despite this substantial computational overhead, its mean F1-score (0.795) remained lower than those achieved by Random Forest, Gradient Boosting, Logistic Regression, and the neural network models. This finding suggests that the additional computational cost associated with SVC does not necessarily translate into superior predictive performance across the evaluated datasets.

Overall, the efficiency analysis reveals that Passive Aggressive, Gaussian Naïve Bayes, Decision Tree, and Logistic Regression provide the most favorable balance between computational efficiency and predictive performance. In contrast, deep neural network architectures and SVC require substantially greater computational resources, making them less suitable for resource-constrained and edge-computing environments where energy consumption, execution time, and memory utilization are critical considerations.

4.2.1. Execution Time

Figure 3 illustrates the mean execution time of the evaluated machine learning classifiers across the three datasets. Significant differences can be observed among the models due to variations in algorithmic complexity and computational requirements. The fastest models were Gaussian Naïve Bayes and Passive Aggressive, requiring average execution times of 11.54 s and 12.25 s, respectively. Decision Tree and Logistic Regression also demonstrated high computational efficiency, completing execution in 14.31 s and 18.93 s, respectively. These models benefit from relatively simple learning mechanisms and lower computational complexity. K-NN required 21.26 s on average, while Random Forest required 52.66 s due to the additional computational overhead associated with constructing and evaluating multiple decision trees. Gradient Boosting exhibited a substantially higher execution time of 156.67 s because of its sequential boosting process, in which multiple weak learners are trained iteratively.

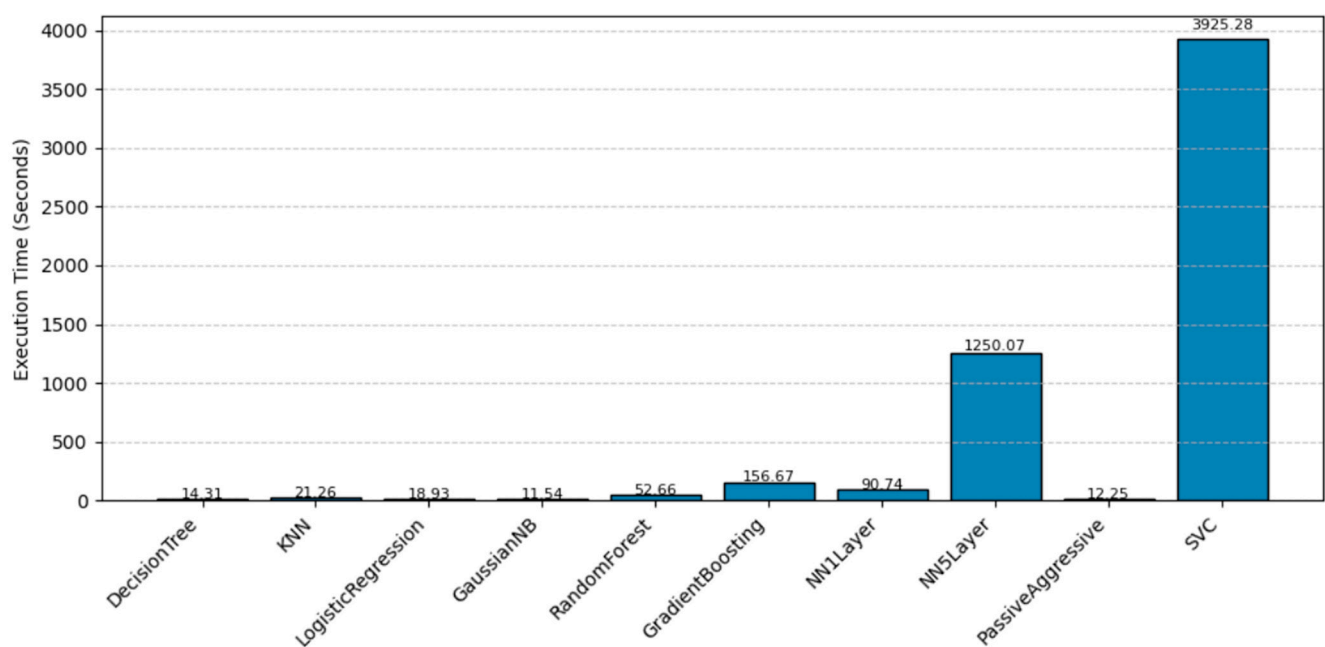


Figure 3. Mean Model Execution Time.

The neural network models required considerably longer execution times. The neural network with 1 hidden layer completed execution in 90.74 s, whereas the neural network with 5 hidden layers required 1250.07 s due to the increased number of trainable parameters and optimization iterations. SVC demonstrated the highest computational burden, requiring an average execution time of 3925.28 s. This substantial increase is primarily

attributed to the computational complexity of kernel-based optimization and support vector calculations. The results indicate that increasing model complexity generally leads to longer execution times and greater computational cost. Therefore, lightweight classifiers such as Gaussian Naïve Bayes, Passive Aggressive, Decision Tree, and Logistic Regression are more suitable for applications requiring rapid model execution.

4.2.2. Memory Usage

Figure 4 presents the mean peak memory utilization of the evaluated classifiers. Considerable variation can be observed across the models, reflecting differences in data storage requirements and model architecture complexity. Logistic Regression (2136 KB), Decision Tree (2205 KB), Gradient Boosting (2258 KB), Random Forest (2321 KB), Passive Aggressive (3020 KB), and Gaussian Naïve Bayes (3070 KB) demonstrated relatively low memory requirements. The neural network models required moderate memory resources, with the neural network having 1 hidden layer consuming 3602 KB and the neural network with 5 hidden layers consuming 6701 KB. Although these values are higher than those of traditional machine learning models, they remain substantially lower than those observed for K-NN and SVC. K-NN exhibited exceptionally high memory utilization, requiring 91,428 KB on average. This behavior is expected because the algorithm stores the complete training dataset and performs distance calculations during prediction. SVC also required substantial memory resources, consuming 62,389 KB due to the storage and processing requirements associated with support vectors and kernel computations. Therefore, the results indicate that simpler linear and tree-based models are significantly more memory-efficient than instance-based and kernel-based approaches. Consequently, Logistic Regression, Decision Tree, Gaussian Naïve Bayes, and Passive Aggressive represent attractive choices for deployment in memory-constrained environments.

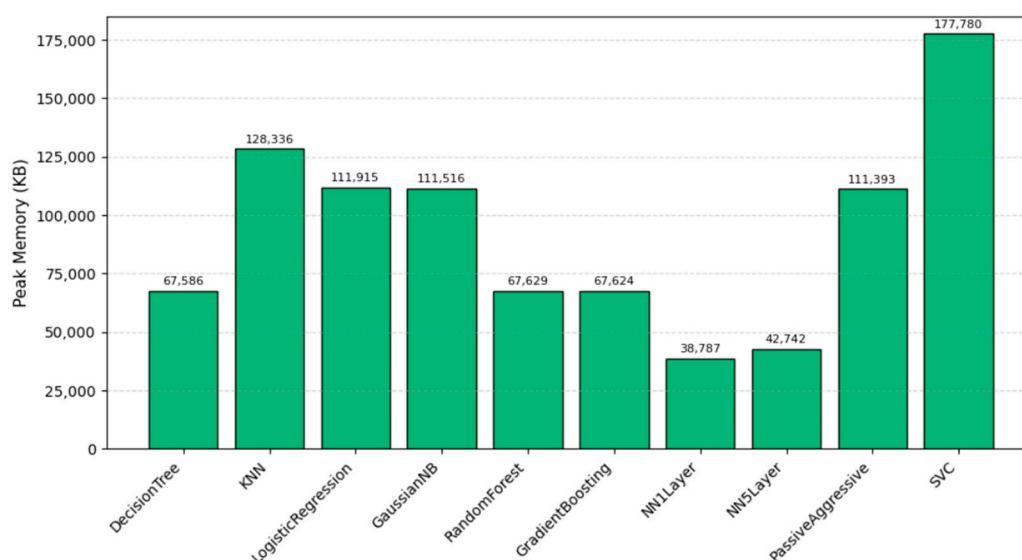


Figure 4. Mean Peak Memory Utilization.

4.2.3. Energy Consumption

The mean energy consumption results, shown in Figure 5, reveal substantial differences in the computational energy requirements of the evaluated classifiers. Gaussian Naïve Bayes, Passive Aggressive, Decision Tree, and Logistic Regression emerged as the most energy-efficient models, consuming only 127 J, 134 J, 156 J, and 327 J on average, respectively. These low energy requirements are primarily attributable to their relatively simple model structures and efficient learning processes. Moderate energy consumption

was observed for K-NN (484 J) and Random Forest (559 J), which achieved competitive predictive performance while maintaining reasonable computational efficiency. Gradient Boosting consumed considerably more energy, requiring 1610 J on average due to its iterative boosting mechanism. Similarly, the neural network with 1 hidden layer consumed 1175 J as a result of repeated optimization during training.

The most energy-intensive models were the neural network with 5 hidden layers and SVC. The deeper neural network consumed 24,766 J on average, reflecting the substantial computational cost associated with training multiple hidden layers. SVC exhibited the highest energy consumption by a large margin, requiring 45,758 J, nearly twice that of the deep neural network model. This high energy demand is attributable to the complexity of kernel-based optimization and support vector computations. Collectively, these findings demonstrate a clear relationship between model complexity and energy consumption. Lightweight classifiers such as Gaussian Naïve Bayes, Passive Aggressive, Decision Tree, and Logistic Regression offer substantial energy savings while maintaining competitive predictive performance. In contrast, deep neural networks and SVC impose significant computational costs, making them less suitable for energy-constrained and sustainability-focused computing environments.

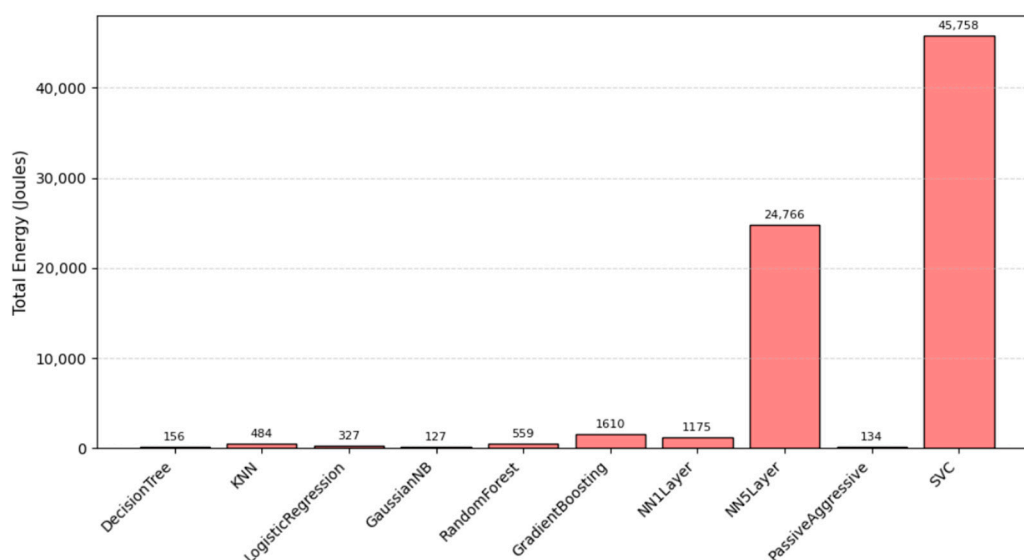


Figure 5. Mean Energy Consumption.

4.2.4. Training and Inference Cost Analysis

While the previous analysis examined overall computational efficiency, Table 7 provides a more detailed breakdown of training and inference costs across the Fake News, Brain Stroke, and Diabetes datasets. Separating these phases is important because training and inference place different demands on computational resources. Training typically represents a one-time cost associated with model construction, whereas inference cost directly affects deployment performance and operational efficiency. The current research confirms that training energy consumption was the dominant contributor to total computational cost for most classifiers. In particular, SVC, Passive Aggressive, and deep neural network models required substantially greater training time and energy than traditional machine learning approaches. On the Diabetes dataset, SVC exhibited the highest computational burden, requiring 10,262 s of training time and consuming 118,464 J of energy. Similarly, the Passive Aggressive classifier required 3635 s of training time and consumed 71,672 J of energy. In contrast, lightweight classifiers such as Decision Tree, K-NN,

Gaussian Naïve Bayes, and Random Forest completed training within only a few seconds while consuming comparatively low amounts of energy.

The inference results reveal a different pattern. For most classifiers, inference time and energy consumption remained considerably lower than training costs. Decision Tree, Random Forest, K-NN, and Gradient Boosting demonstrated particularly efficient inference behavior across all datasets, requiring less than one second of inference time and only a few joules of energy. However, some classifiers exhibited substantial inference overhead. For example, Passive Aggressive required 71 s and 1921 J for inference on the Fake News dataset, while SVC required 1214 s and 14,917 J for inference on the Diabetes dataset. Gaussian Naïve Bayes also showed unusually high inference energy consumption on the Diabetes dataset, consuming 1055.08 J despite relatively low training requirements. The results highlight that training and inference costs can differ substantially across machine learning algorithms. While some models incur high training costs but remain efficient during deployment, others continue to consume considerable resources during inference. These findings highlight the importance of evaluating both phases independently when selecting machine learning models for resource-constrained and sustainability-focused environments.

Table 7. Training vs. Inference Results for Fake News, Brain Stroke and Diabetes.

Classifier	Fake News				Brain Stroke				Diabetes			
	Train Time (s)	Train Energy (J)	Inference Time (s)	Inference Energy (J)	Train Time (s)	Train Energy (J)	Inference Time (s)	Inference Energy (J)	Train Time (s)	Train Energy (J)	Inference Time (s)	Inference Energy (J)
Decision Tree	6.23	70.15	0.16	1.97	0.16	1.97	0.02	0.03	2.96	31	0.01	0.25
Gaussian Naïve Bayes	0.15	2.60	0.02	0.48	0.02	0.50	0.08	1.84	0.70	6.50	28.73	1056
Gradient Boosting	0.80	20.48	0.14	1.50	0.14	1.50	0.01	0.14	22.19	589	0.01	0.30
K-NN	0.36	5.82	0.04	0.61	0.04	0.61	0.02	0.21	0.58	7	0.02	0.28
Logistic Regression	6.46	68.91	6.59	68.09	6.59	68	0.20	1.78	108.38	1146	2.75	24.28
NN 1 Hidden Layer	45.56	479.07	5.38	56.26	5.38	56	0.03	0.34	385.02	3925	0.46	4.58
NN 5 Hidden Layers	33.12	985.03	31.15	330	31.15	330	0.02	0.14	174.30	1840	0.03	0.35
Passive Aggressive	10.97	330.67	71	1921	71	1921	0.03	0.50	3635	71,672	0.18	5.86
Random Forest	1.36	14.29	0.19	2.16	0.19	2.16	0.01	0.15	1.65	18	0.01	0.22
SVC	216	2923	35.97	404	35.97	404	0.92	11.12	10,262	118,464	1214	14,917

4.3. Hyperparameter Sensitivity Analysis

Furthermore, to assess the robustness of the reported findings, a hyperparameter sensitivity analysis was conducted for three representative classifiers—Random Forest, Neural Network with 5 Hidden Layers, and Support Vector Classifier (SVC)—as presented in Table A3. Alternative configurations were evaluated across the Fake News, Brain Stroke, and Diabetes datasets. The objective was to determine whether variations in model complexity significantly altered the observed trade-offs between predictive performance and computational efficiency.

The hyperparameter sensitivity analysis demonstrates in Table A3 that, although certain parameter configurations produced modest improvements in predictive performance, these gains were frequently accompanied by substantial increases in execution time and energy consumption. For Random Forest, increasing the number of trees and model depth generally increased computational cost while providing only marginal

improvements in accuracy and F1-score. Similar trends were observed for the Neural Network with 5 Hidden Layers, where larger network architectures occasionally improved predictive performance but required considerably greater computational resources. For SVC, the impact of hyperparameter selection varied across datasets; however, more complex kernel configurations did not consistently outperform simpler alternatives and often incurred significantly higher computational costs.

The overall ranking of classifiers remained largely unchanged across the evaluated configurations, indicating that the primary conclusions of this study regarding the trade-off between predictive performance and computational efficiency are robust to reasonable variations in hyperparameter settings. Figures A1–A3 visualize the relationship between predictive performance, energy consumption, and execution time for the evaluated hyperparameter configurations, where bubble size represents total execution time and the horizontal axis represents total energy consumption.

4.4. Correlation Analysis

The correlation heatmap presented in Figure 6 illustrates the relationships among predictive performance and computational efficiency metrics across the evaluated machine learning models. Specifically, the analysis examines the associations between accuracy, energy consumption, memory utilization, and execution time. The results reveal a weak-to-moderate negative relationship between accuracy and computational resource requirements. Accuracy exhibits negative correlations with energy consumption ($r = -0.36$) and execution time ($r = -0.33$), indicating that higher computational cost does not necessarily translate into improved predictive performance. The relationship between accuracy and memory utilization is negligible ($r = -0.03$), suggesting that memory requirements have little direct influence on classification performance across the evaluated models.

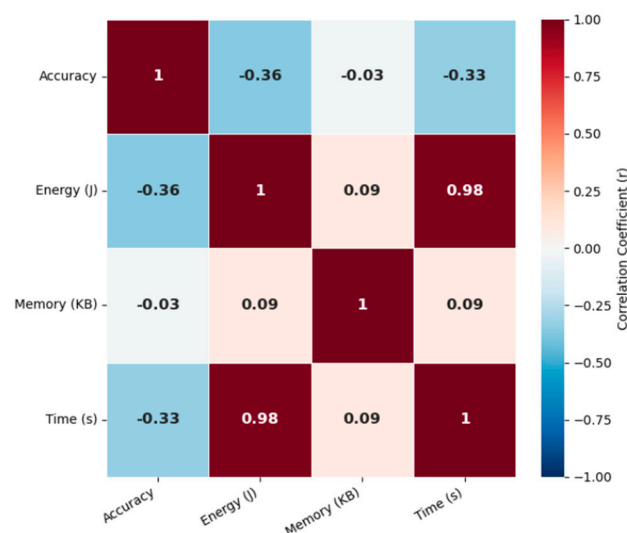


Figure 6. Correlation Heatmap of Model Performance and Efficiency Metrics.

The strongest relationship observed in the analysis is between energy consumption and execution time ($r = 0.98$). This near-perfect positive correlation indicates that execution time is the primary factor driving energy consumption in the evaluated CPU-based machine learning workloads. Models requiring longer processing times consistently consume greater amounts of energy, regardless of their predictive performance. In contrast, memory utilization exhibits only weak positive correlations with both energy consumption ($r = 0.09$) and execution time ($r = 0.09$). These findings propose that memory usage

contributes relatively little to overall computational efficiency when compared with processing time. Consequently, reducing execution time is likely to yield substantially greater energy savings than focusing solely on memory optimization.

Finally, the correlation analysis demonstrates that predictive performance and computational efficiency are largely independent objectives. While some computationally intensive models achieve strong predictive performance, they do not consistently outperform simpler alternatives. Models such as Logistic Regression, Decision Tree, and Passive Aggressive provide favorable trade-offs by maintaining competitive classification performance while requiring substantially lower energy consumption and execution time. These findings emphasize the importance of considering both predictive effectiveness and computational efficiency when selecting machine learning models for practical and sustainability-conscious applications.

4.5. Trade-Off Analysis

Table 6 highlights the trade-offs between predictive performance and computational efficiency across the evaluated machine learning classifiers. The results demonstrate that achieving higher predictive accuracy often requires increased computational resources, although this relationship is not absolute. Random Forest achieved the highest mean accuracy (0.878) and mean F1-score (0.877) among all evaluated classifiers while maintaining moderate computational requirements, consuming 559 J of energy and requiring 53 s of execution time on average. Gradient Boosting produced comparable predictive performance, achieving a mean accuracy of 0.871 and a mean F1-score of 0.869. However, this improvement came at a substantially higher computational cost, with average energy consumption reaching 1610 J and execution time increasing to 157 s. Lightweight classifiers such as Logistic Regression, Decision Tree, and Passive Aggressive demonstrated favorable trade-offs between predictive performance and computational efficiency. Logistic Regression achieved a mean accuracy of 0.815 and a mean F1-score of 0.843 while consuming only 327 J of energy and requiring 19 s of execution time. Similarly, Decision Tree achieved a mean accuracy of 0.818 and a mean F1-score of 0.826 while consuming only 156 J of energy. Passive Aggressive was among the most computationally efficient models, requiring only 134 J of energy and 12 s of execution time, although its predictive performance was lower than that of the best-performing ensemble methods.

In contrast, more computationally intensive models did not consistently deliver proportional gains in predictive performance. The neural network with 5 Hidden Layers achieved a mean accuracy of 0.823 and a mean F1-score of 0.845 but required 24,766 J of energy and 1250 s of execution time. Similarly, SVC exhibited the highest computational cost among all evaluated classifiers, consuming 45,758 J of energy and requiring 3925 s of execution time while achieving a mean accuracy of only 0.736. These findings suggest that increased model complexity does not necessarily result in superior predictive performance. Therefore, the results indicate that no single classifier simultaneously optimizes predictive performance and computational efficiency. While ensemble methods such as Random Forest and Gradient Boosting provide the strongest predictive performance, lightweight models such as Logistic Regression, Decision Tree, and Passive Aggressive offer more favorable efficiency–performance trade-offs. Consequently, the choice of classifier should be guided by the specific requirements of the target application, including constraints related to energy consumption, execution time, and available computational resources.

The radar chart presented in Figure 7 further illustrates the trade-offs among predictive accuracy, energy efficiency, and runtime efficiency for the top-performing classifiers. Random Forest exhibits the most balanced profile, combining the highest predictive accuracy with strong efficiency characteristics. Gradient Boosting achieves comparable

predictive performance but demonstrates lower efficiency due to its greater computational requirements. The neural network with 1 hidden layer provides moderate predictive performance with reasonable efficiency, whereas the neural network with 5 hidden layers suffers from substantial reductions in both energy and runtime efficiency because of its high computational demands. Decision Tree maintains a highly balanced profile, achieving competitive predictive performance while requiring significantly fewer computational resources than more complex models.

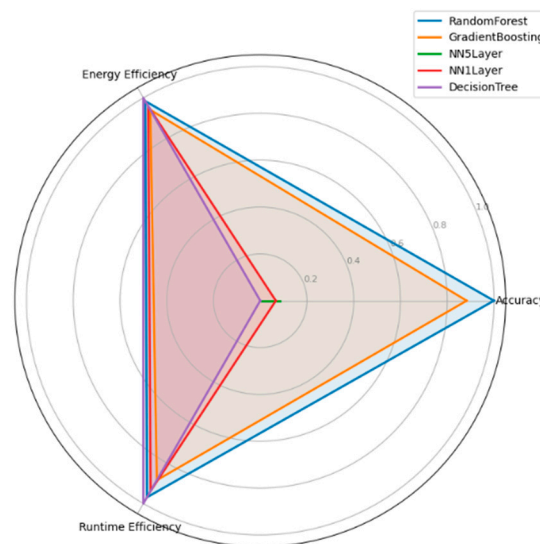


Figure 7. Comparison of Top Five Supervised Classifiers.

4.6. Multi-Objective Benchmarking Analysis

The proposed multi-objective benchmarking framework integrates predictive performance and computational efficiency into a unified evaluation methodology for supervised machine learning models. While existing benchmarking approaches such as Eco2AI primarily focus on energy consumption and carbon emissions, and frameworks such as MLPerf Power emphasize hardware-level performance and power measurements, the framework recommended in this study evaluates models from both predictive and sustainability perspectives. This is achieved through the Predictive Performance Index (PPI), Efficiency Index (EI), Composite Index (CI), and Pareto analysis. By simultaneously considering classification performance, execution time, memory utilization, and energy consumption, the framework provides a more comprehensive basis for model selection than predictive metrics alone.

The Composite Index (CI) provides an overall measure of how effectively each classifier balances predictive performance and computational efficiency. As shown in Table 8, Random Forest achieved the highest composite score (CI = 0.950), indicating the most favorable overall balance between classification effectiveness and resource efficiency. This result is consistent with its strong predictive performance, reflected by the highest mean accuracy and mean F1-score among the evaluated models, while maintaining moderate energy consumption and execution time requirements. Gradient Boosting ranked second with a composite score of 0.919, followed closely by the neural network with 1 hidden layer (CI = 0.909) and Logistic Regression (CI = 0.905). These models achieved strong predictive performance while maintaining acceptable computational efficiency. Passive Aggressive also performed well, achieving a CI of 0.846, largely due to its exceptionally low energy consumption and execution time requirements. Decision Tree (CI = 0.717) and Gaussian Naïve Bayes (CI = 0.630) occupied the middle range of the ranking. Although

these models demonstrated excellent computational efficiency, their comparatively lower predictive performance reduced their overall composite scores. K-NN achieved a CI of 0.571, primarily due to its high memory utilization despite moderate predictive performance.

At the lower end of the ranking, the neural network with 5 hidden layers achieved a CI of 0.747. Although it demonstrated competitive predictive performance, its substantial energy consumption and execution time significantly reduced its overall efficiency. SVC achieved the lowest composite score (CI = 0.383), reflecting its extremely high computational requirements in terms of energy consumption, execution time, and memory utilization. Despite requiring considerably greater computational resources, SVC did not provide corresponding improvements in predictive performance, making it the least favorable model from a multi-objective perspective. Overall, the findings show that Random Forest, Gradient Boosting, Logistic Regression, Passive Aggressive, and the neural network with 1 hidden layer provide the strongest balance between predictive performance and computational efficiency. In contrast, computationally intensive models such as the neural network with 5 hidden layers and SVC struggle to maintain a favorable sustainability profile due to their excessive resource requirements.

Figure 8 visually reinforces these findings by comparing the Composite Index values across all evaluated classifiers. Models with higher CI values achieve a more effective balance between predictive effectiveness and computational efficiency, while lower CI values show that improvements in one objective are offset by increased computational costs in another. The results demonstrate that Random Forest represents the most balanced solution among the evaluated classifiers, whereas SVC exhibits the least favorable trade-off between performance and efficiency.

Table 8. Experimental Results of Multi-Objective Benchmarking Analysis

Classifier	Predictive Performance	Efficiency Index (EI)	Composite Index (CI)
	Index (PPI)		
Decision Tree	0.435	0.999	0.717
Gaussian Naïve Bayes	0.263	0.997	0.630
Gradient Boosting	0.863	0.976	0.920
K-NN	0.480	0.663	0.571
Logistic Regression	0.813	0.998	0.905
NN 1 Hidden Layer	0.838	0.980	0.909
NN 5 Hidden Layers	0.796	0.698	0.747
Passive Aggressive	0.696	0.997	0.846
Random Forest	0.907	0.993	0.950
SVC	0.657	0.108	0.383

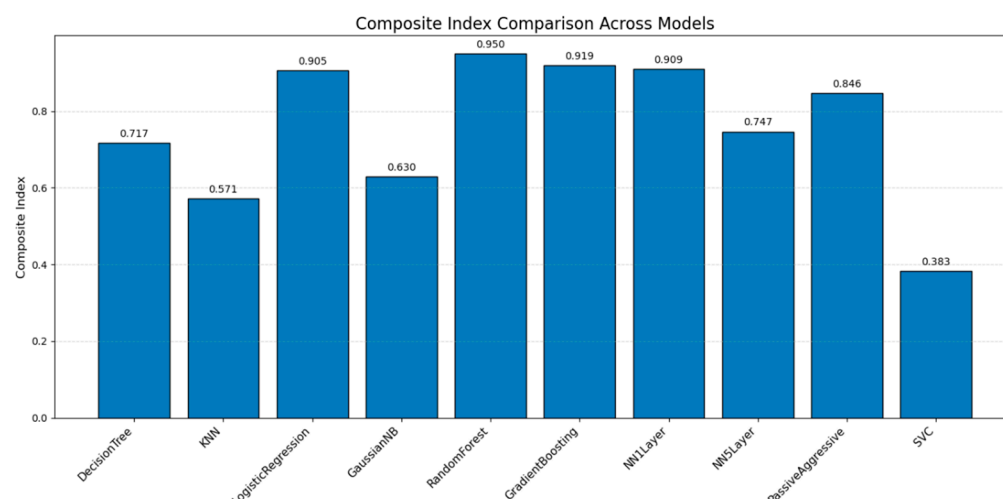


Figure 8. Composite Index Comparison across Supervised Learning Models.

The Pareto front analysis provides a visual representation of the trade-off between predictive performance and computational efficiency. Models located on or near the Pareto frontier represent the most desirable solutions because improvements in predictive performance cannot be achieved without sacrificing efficiency, and vice versa. As illustrated in Figure 9, Random Forest achieved the highest Predictive Performance Index (PPI = 0.907) while maintaining a very high Efficiency Index (EI = 0.993). This position places Random Forest on the Pareto frontier and confirms its status as the most balanced model among the evaluated classifiers. Logistic Regression (PPI = 0.813, EI = 0.998) and Passive Aggressive (PPI = 0.696, EI = 0.997) also lie on the Pareto frontier, demonstrating that strong computational efficiency can be achieved while maintaining competitive predictive performance. Decision Tree occupies the efficient end of the frontier, exhibiting near-perfect efficiency (EI = 0.999) while maintaining moderate predictive performance (PPI = 0.435).

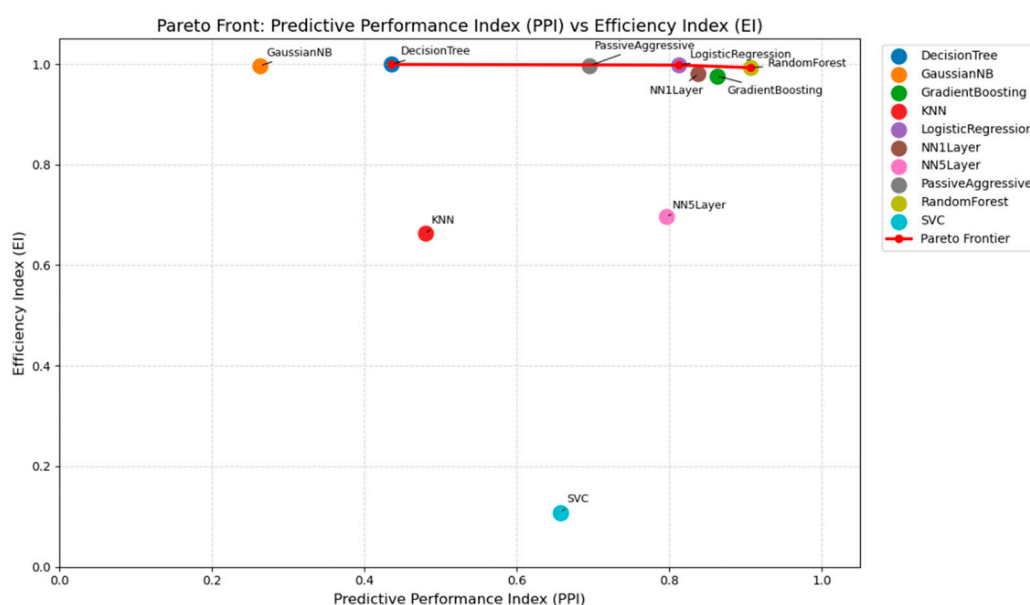


Figure 9. Pareto Front Analysis Showing the Relationship between Predictive Performance Index (PPI) and Efficiency Index (EI).

Several models were located below the Pareto frontier, indicating less favorable trade-offs between predictive performance and computational efficiency. Although

Gradient Boosting achieved strong predictive performance (PPI = 0.863), its lower efficiency (EI = 0.976) prevented it from dominating the frontier solutions. Similarly, the neural network with 1 hidden layer achieved competitive predictive performance (PPI = 0.838) but required greater computational resources than Logistic Regression and Random Forest. The neural network with 5 hidden layers and SVC exhibited the least favorable trade-offs. Despite achieving moderate predictive performance, their substantially lower efficiency scores (EI = 0.698 and EI = 0.108, respectively) reflect the significant computational costs associated with these models. In particular, SVC required considerably higher energy consumption, execution time, and memory resources without providing corresponding improvements in predictive performance. The Pareto analysis highlights that no single metric should be used in isolation when selecting machine learning models. While predictive performance remains important, computational efficiency plays a critical role in determining the practical suitability of a model. The results demonstrate that Random Forest, Logistic Regression, Passive Aggressive, and Decision Tree provide the most favorable balance between predictive effectiveness and computational efficiency, making them particularly attractive for real-world deployments where resource constraints and sustainability considerations are important.

The Pareto frontier confirms the findings of the composite index analysis, demonstrating that models with balanced predictive performance and computational efficiency provide greater practical value than models that optimize only a single objective. Table 9 summarizes the comparative analysis of all classifiers

Table 9. Key Findings of Machine Learning Models.

Classifier	Predictive Level	Efficiency Level	Findings
Decision Tree	Medium–High	Very High	Delivers good predictive performance with excellent computational efficiency.
Gaussian Naïve Bayes	Low–Medium	Very High	Extremely efficient and lightweight, but predictive performance is comparatively lower.
Gradient Boosting	Very High	High	One of the strongest performers, achieving high predictive accuracy while maintaining good efficiency.
K-NN	Medium	Low	Provides moderate predictive performance but suffers from lower efficiency due to higher execution costs.
Logistic Regression	High	Very High	Offers an excellent balance between predictive performance and computational efficiency.
NN 1 Hidden Layer	Very High	High	Achieves near-top predictive performance with good efficiency, making it a strong practical choice.
NN 5 Hidden Layers	High	Moderate	Delivers strong predictive performance but requires greater computational resources than simpler neural architectures.
Passive Aggressive	High	Very High	Combines strong predictive capability with excellent efficiency, making it suitable for resource-constrained environments.
Random Forest	Highest	Very High	Achieves the best overall balance of predictive performance and efficiency, ranking first in the Composite Index.
SVC	Medium	Very Low	Shows reasonable predictive capability but poor efficiency, limiting its practicality for resource-sensitive deployments.

4.7. Limitations of Study

Although the findings provide useful insights, the study does have its limitations. The evaluation was carried out entirely on a CPU-based system, which means that the

results do not reflect how models might behave in GPU or multi-node settings. The number of datasets, while diverse in type and scale, is still limited, and expanding to more domains may reveal trends in energy behavior. Hyperparameter tuning was intentionally kept minimal additional to avoid injecting variability into the energy measurements, but this also means that some models were not operating at their absolute optimal configuration. Energy measurements were restricted to CPU and DRAM consumption using Intel RAPL, which does not capture the full system-wide energy profile. These factors place some boundaries on the generalizability of the results, but they do not undermine the overall conclusions about the trade-off between performance and efficiency.

5. Conclusions

This study benchmarked ten supervised machine learning classifiers across three multi-domain classification datasets, namely Fake News, Brain Stroke, and Diabetes, while simultaneously evaluating predictive performance and computational efficiency. In addition to traditional classification metrics, the analysis incorporated execution time, memory utilization, and energy consumption to provide a comprehensive assessment of model behavior. Furthermore, a multi-objective benchmarking framework based on the Predictive Performance Index (PPI), Efficiency Index (EI), Composite Index (CI), and Pareto analysis was proposed to evaluate the trade-offs between predictive effectiveness and computational sustainability. The experimental results demonstrate that higher computational cost does not necessarily result in superior predictive performance. Among the evaluated classifiers, Random Forest achieved the strongest overall performance, obtaining the highest mean accuracy and mean F1-score while maintaining moderate computational requirements. The Composite Index and Pareto Front analyses consistently identified Random Forest as the most balanced model, offering an effective compromise between predictive performance and computational efficiency.

Lightweight models such as Logistic Regression, Passive Aggressive, and Decision Tree also demonstrated favorable efficiency–performance trade-offs. These classifiers achieved competitive predictive performance while maintaining low energy consumption, short execution times, and modest memory requirements. The results signify that effective machine learning solutions can often be achieved without relying on computationally intensive algorithms. In contrast, more complex models such as the neural network with 5 hidden layers and Support Vector Classifier (SVC) required substantially greater computational resources. These models consumed significantly more energy, required longer execution times, and utilized greater memory resources while providing limited improvements in predictive performance. In some cases, their predictive performance was inferior to that of simpler and more efficient alternatives. These findings suggest that increased model complexity does not necessarily translate into proportional gains in classification effectiveness. The correlation analysis further revealed a near-perfect relationship between energy consumption and execution time, highlighting runtime as the primary driver of computational energy usage in CPU-based machine learning workloads. The results also showed only weak relationships between predictive performance and computational resource requirements, reinforcing the importance of evaluating machine learning models using both performance and efficiency criteria.

These outcomes demonstrate that machine learning model selection should not be based solely on predictive performance. Energy consumption, execution time, and memory utilization are equally important considerations, particularly in sustainability-focused and resource-constrained environments. By integrating predictive and computational metrics into a unified benchmarking framework, this work provides practical guidance for selecting machine learning models that achieve strong predictive performance while minimizing computational cost.

The selected models were predominantly machine learning algorithms that are designed to operate efficiently on CPUs, and several of them do not inherently support GPU acceleration. Therefore, CPU-based benchmarking provides a fair and consistent evaluation framework across all investigated classifiers. There are a number of meaningful directions in which this work can be extended. One useful next step would be to expand the evaluation to include more model families, such as unsupervised learning or regression-based, and incorporate GPU-level energy measurement, in particular for transformer-based models and large language models that dominate many current applications. Another direction worth exploring is how energy consumption behaves in distributed or cloud-based training environments, since many real systems run at large scale rather than on a single CPU machine. Further studies would be useful in the future to more closely analyze the optimization strategies such as pruning, quantization, and early stopping to determine how much computational efficiency can be improved without affecting predictive performance. A wider range of domains, particularly those involving real-time or streaming data, could be added to the collection of datasets used in these kinds of studies. This would give a better illustration of the performance of different models under conditions similar to those that they undergo in daily operations. Finally, the advancement of the tools that assist in measuring carbon impact or the tools that help to design the cost-conscious scheduling may encourage the development of machine learning systems that would be both effective in operation and environmentally responsible.

Author Contributions: Conceptualization, R.Q. and A.A.; methodology, R.Q. and R.A.; validation, S.H. and R.A.; formal analysis, R.Q. and A.A.; investigation, R.Q., A.A. and R.A.; writing—original draft preparation, R.Q. and R.A.; writing—review and editing, R.A. and S.H.; project administration, R.Q. and S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Fake News, Brain Stroke and Diabetes datasets at (https://github.com/GeorgeMcIntire/fake_real_news_dataset, <https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset>, <https://doi.org/10.24432/C53919> (accessed on 10 February 2026)).

Acknowledgments: The authors sincerely thank the reviewers for their constructive feedback, which significantly improved the quality of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A presents the model configurations and hyperparameter sensitivity experiments conducted to evaluate the robustness of the reported findings. Table A1 summarizes the training parameters used in the primary experiments, while Table A2 outlines the alternative hyperparameter configurations evaluated during the sensitivity analysis. Table A3 presents the corresponding results across the three benchmark datasets. The objective of these experiments was to assess whether variations in model complexity significantly affected predictive performance, energy consumption, execution time, and the overall trade-offs between predictive effectiveness and computational efficiency.

Table A1. Training Parameters of Machine Learning Models.

Classifier	Hyperparameters
------------	-----------------

Decision Tree	criterion = gini; splitter = best; max_depth = 10; min_samples_split = 5; min_samples_leaf = 5
Gaussian Naïve Bayes	var_smoothing = 1×10^{-9}
Gradient Boosting	• loss = log_loss; learning_rate = 0.1; n_estimators = 100; max_depth = 5
K-NN	n_neighbors = 10; weights = distance; algorithm = auto; metric = minkowski
Logistic Regression	penalty = l2; C = 1.0; max_iter = 1000; solver = lbfgs; random_state = 42
NN 1 Hidden Layer	hidden_layer_sizes = (10); max_iter = 500; activation = relu; solver = adam
NN 5 Hidden Layers	hidden_layer_sizes = (64, 64, 32, 32, 16); max_iter = 500; activation = relu; solver = adam; alpha = 0.0001
Passive Aggressive	C = 1.0; max_iter = 1000; random_state = 42
Random Forest	n_estimators = 200; max_depth = 20; min_samples_split = 5; min_samples_leaf = 5; max_features = sqrt; random_state = 42
SVC	• C = 1.0; kernel = rbf; gamma = 0.1; probability = False; random_state = 42

Table A2. Hyperparameter Configurations Evaluated.

Classifier	Configuration	Hyperparameters
Random Forest	RF_Default	n_estimators = 100, max_depth = None, random_state = 42
Random Forest	RF_200_20	n_estimators = 200, max_depth = 20, random_state = 42
Random Forest	• RF_300_30	• n_estimators = 300, max_depth = 30, random_state = 42
Random Forest	RF_500_None	n_estimators = 500, max_depth = None, random_state = 42
NN5	NN5_Default	hidden_layer_sizes = (64, 64, 32, 32, 16), activation = 'relu', solver = 'adam', max_iter = 500
NN5	NN5_Small	hidden_layer_sizes = (32, 32, 16, 16, 8), activation = 'relu', solver = 'adam', max_iter = 500
NN5	NN5_Medium	hidden_layer_sizes = (128, 64, 32, 16, 8), activation = 'relu', solver = 'adam', max_iter = 500
NN5	NN5_Large	hidden_layer_sizes = (128, 128, 64, 32, 16), activation = 'relu', solver = 'adam', max_iter = 500
SVC	SVC_Linear	kernel = 'linear', probability = False
SVC	• SVC_RBF	• kernel = 'rbf', C = 1, gamma = 'scale', probability = False
SVC	• SVC_Poly	• kernel = 'poly', degree = 3, probability = False
SVC	• SVC_RBF_C10	• kernel = 'rbf', C = 10, gamma = 'scale', probability = False

Table A3. Hyperparameter Sensitivity Results across Datasets.

Model Configuration	Fake News				Brain Stroke				Diabetes			
	Accuracy	F1-Score	Energy (J)	Time (s)	Accuracy	F1-Score	Energy (J)	Time (s)	Accuracy	F1-Score	Energy (J)	Time (s)
RF_Default	0.919	0.919	403	36.7	0.924	0.918	68	6.8	0.803	0.793	804	74.7
RF_200_20	0.901	0.9	423	38.4	0.925	0.919	141	13.5	0.797	0.797	1309	122.9
RF_300_30	0.903	0.903	498	45.3	0.924	0.918	211	20	0.803	0.794	2230	209
RF_500_None	0.919	0.919	714	64.6	0.925	0.919	349	33.3	0.804	0.794	3795	352.6
NN5_Default	0.934	0.934	665	40.4	0.927	0.922	1915	72.6	0.607	0.68	74,381	3694
NN5_Small	0.94	0.94	684	41.1	0.888	0.903	384	33.1	0.604	0.685	15,875	1543
NN5_Medium	0.938	0.938	899	48.1	0.903	0.911	983	35.7	0.621	0.686	133,041	5710
NN5_Large	0.933	0.933	770	44.8	0.94	0.926	2624	101.3	0.659	0.712	6252	8317
SVC_Linear	0.938	0.938	1961	149	0.95	0.926	1003	99.7	0.582	0.67	117,246	9424
SVC_RBF	0.942	0.942	3579	274	0.691	0.778	92	7.8	0.568	0.659	135,852	11,641
SVC_Poly	0.898	0.898	4654	350	0.818	0.864	79	6.5	0.564	0.657	22,845	9050
SVC_RBF_C10	0.944	0.944	4337	337	0.852	0.885	79	6.8	0.574	0.663	143,011	12,203

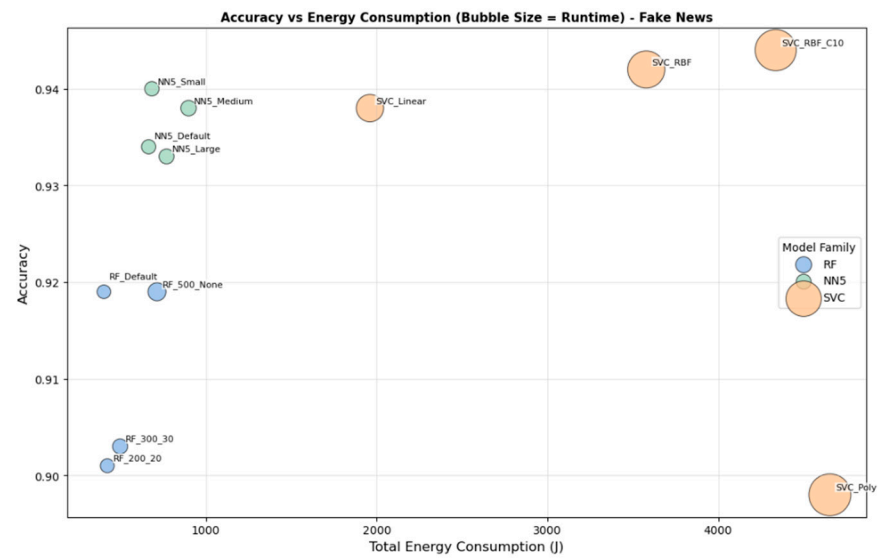


Figure A1. Hyperparameter Sensitivity Analysis on Fake News Dataset.

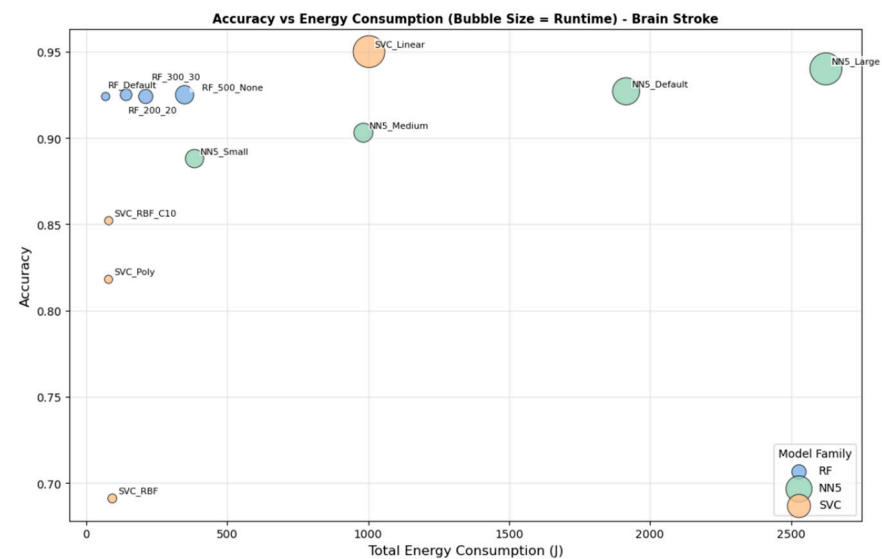


Figure A2. Hyperparameter Sensitivity Analysis on Brain Stroke Dataset.

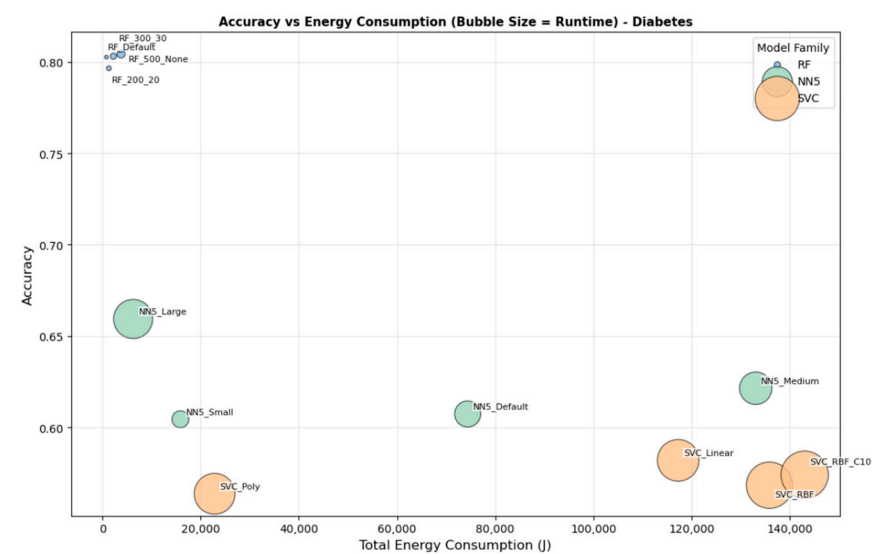


Figure A3. Hyperparameter Sensitivity Analysis on Diabetes Dataset.

References

1. Malmodin, J.; Belkhir, L.; Elemeligi, A.Y.; Hilty; Aebischer; Andrae, A.; Elder, R.; Lundén, D.; Bergmark, P.; Jones, M.; et al. Greenhouse Gas Emissions in the ICT Sector. 2020. Available online: <https://c2e2.unepccc.org/wp-content/uploads/sites/3/2020/03/greenhouse-gas-emissions-in-the-ict-sector.pdf> (accessed on 15 March 2026).
2. Jones, N. How to stop data centres from gobbling up the world's electricity. *Nature* **2018**, *561*, 163–166. <https://doi.org/10.1038/d41586-018-06610-y>.
3. Liu, Y.; Wei, X.; Xiao, J.; Liu, Z.; Xu, Y.; Tian, Y. Energy consumption and emission mitigation prediction based on data center traffic and PUE for global data centers. *Glob. Energy Interconnect.* **2020**, *3*, 272–282. <https://doi.org/10.1016/j.gloi.2020.07.008>.
4. Andrae, A.S.G. New perspectives on internet electricity use in 2030. *Eng. Appl. Sci. Lett.* **2020**, *3*, 19–31. <https://doi.org/10.30538/psrp-easl2020.0038>.
5. Cambridge Blockchain Network Sustainability Index: CBECI. Available online: <https://ccaf.io/cbnsi/cbeci> (accessed on 17 March 2026).
6. Hao, K. Training a Single AI Model Can Emit as Much Carbon as Five Cars in Their Lifetimes. MIT Technology Review. 7 December 2020. Available online: <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/> (accessed on 21 April 2026).
7. Su, Z.; Li, Q.; Kaneko, H.; Li, H.; Meng, L. Optimization and Deployment of DNNs for RISC-V-based Edge AI. In *2024 IEEE International Conference on Real-Time Computing and Robotics (RCAR)*; IEEE: Piscataway, NJ, USA, 2024; pp. 200–205. <https://doi.org/10.1109/RCAR61438.2024.10671295>.
8. Wu, Y.; Wu, C.; Yuan, G.; Li, Y.; Guo, W.; Rao, J.; Shen, X.; Ren, B.; Wang, Y. DACO: Pursuing Ultra-Low Power Consumption via DNN-Adaptive CPU-GPU CO-Optimization on Mobile Devices. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–2. <https://doi.org/10.23919/DATE58400.2024.10546884>.
9. Dhiman, R.; Miteff, S.; Wang, Y.; Ma, S.-C.; Amirikas, R.; Fabian, B. Artificial Intelligence and Sustainability — A review. *Analytics* **2024**, *3*, 140–164. <https://doi.org/10.3390/analytics3010008>.
10. Gutiérrez, M.; Moraga, M.Á.; García, F. Analysing the Energy Impact of Different Optimisations for Machine Learning Models. In *2022 International Conference on ICT for Sustainability (ICT4S)*; IEEE: Piscataway, NJ, USA, 2022; pp. 46–52. <https://doi.org/10.1109/ICT4S55073.2022.00016>.
11. Qamar, R.; Asif, R.; Jameel, S.M. Towards Sustainable AI: Benchmarking energy efficiency of deep neural networks for Resource-Constrained Edge Devices. *Information* **2026**, *17*, 380. <https://doi.org/10.3390/info17040380>.
12. Ferro, M.; Silva, G.D.; De Paula, F.B.; Vieira, V.; Schulze, B. Towards a sustainable artificial intelligence: A case study of energy efficiency in decision tree algorithms. *Concurr. Comput. Pract. Exp.* **2021**, *35*, e6815. <https://doi.org/10.1002/cpe.6815>.
13. Strubell, E.; Ganesh, A.; McCallum, A. Energy and policy considerations for modern deep learning research. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 13693–13696. <https://doi.org/10.1609/aaai.v34i09.7123>.
14. Liu, V.; Yin, Y. Green AI: Exploring carbon footprints, mitigation strategies, and trade offs in large language model training. *Discov. Artif. Intell.* **2024**, *4*, 49. <https://doi.org/10.1007/s44163-024-00149-w>.
15. Yokoyama, A.M.; Ferro, M.; De Paula, F.B.; Vieira, V.G.; Schulze, B. Investigating hardware and software aspects in the energy consumption of machine learning: A green AI-centric analysis. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7825. <https://doi.org/10.1002/cpe.7825>.
16. Yu, J.-R.; Chen, C.-H.; Huang, T.-W.; Lu, J.-J.; Chung, C.-R.; Lin, T.-W.; Wu, M.-H.; Tseng, Y.-J.; Wang, H.-Y. Energy Efficiency of inference algorithms for clinical laboratory data sets: Green Artificial Intelligence Study. *J. Med. Internet Res.* **2021**, *24*, e28036. <https://doi.org/10.2196/28036>.
17. Santos, S.O.S.; Skierski, A.; García-Núñez, D.; Lazzarini, V.; De Andrade Moral, R.; Galvan, E.; Ottoni, A.L.C.; Nepomuceno, E. Green Machine Learning: Analysing the Energy Efficiency of Machine Learning Models. In *2024 35th Irish Signals and Systems Conference (ISSC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. <https://doi.org/10.1109/ISSC61953.2024.10603302>.
18. Islam, M.S.; Zisad, S.N.; Kor, A.-L.; Hasan, M.H. Sustainability of Machine Learning Models: An Energy Consumption Centric Evaluation. In *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. <https://doi.org/10.1109/ECCE57851.2023.10101532>.
19. García-Martín, E.; Rodrigues, C.F.; Riley, G.; Grah, H. Estimation of energy consumption in machine learning. *J. Parallel Distrib. Comput.* **2019**, *134*, 75–88. <https://doi.org/10.1016/j.jpdc.2019.07.007>.
20. Fahad, M.; Shahid, A.; Manumachu, R.R.; Lastovetsky, A. A comparative study of methods for measurement of energy of computing. *Energies* **2019**, *12*, 2204. <https://doi.org/10.3390/en12112204>.

21. Raffin, G.; Trystram, D. Dissecting the Software-Based Measurement of CPU Energy Consumption: A Comparative analysis. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *36*, 96–107. <https://doi.org/10.1109/tpds.2024.3492336>.
22. Trihinas, D.; Thamsen, L.; Beilharz, J.; Symeonides, M. Towards Energy Consumption and Carbon Footprint Testing for AI-Driven IoT Services. In *2022 IEEE International Conference on Cloud Engineering (IC2E)*; IEEE: Piscataway, NJ, USA, 2022; pp. 29–35. <https://doi.org/10.1109/IC2E55432.2022.00011>.
23. Tschand, A.; Rajan, A.T.R.; Idgunji, S.; Ghosh, A.; Holleman, J.; Kiraly, C.; Ambalkar, P.; Borkar, R.; Chukka, R.; Cockrell, T.; et al. MLPerf Power: Benchmarking the Energy Efficiency of Machine Learning Systems from μ Watts to MWatts for Sustainable AI. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1201–1216. <https://doi.org/10.1109/HPCA61900.2025.00092>.
24. Budenny, S.A.; Lazarev, V.D.; Zakharenko, N.N.; Korovin, A.N.; Plosskaya, O.A.; Dimitrov, D.V.; Akhripkin, V.S.; Pavlov, I.V.; Oseledets, I.V.; Barsola, I.S.; et al. ECO2AI: Carbon Emissions Tracking of Machine Learning models as the first step towards Sustainable AI. *Dokl. Math.* **2022**, *106*, S118–S128. <https://doi.org/10.1134/s1064562422060230>.
25. Yatagha, R.; Mejri, O.; Waedt, K.; Ruland, C. Assessing the Complexity and Real-Time Performance of Anomaly Detection Algorithms in Resource-Constrained Environments. In *2024 IEEE 20th International Conference on Intelligent Computer Communication and Processing (ICCP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8. <https://doi.org/10.1109/ICCP63557.2024.10793006>.
26. Verdecchia, R.; Cruz, L.; Sallou, J.; Lin, M.; Wickenden, J.; Hotellier, E. Data-Centric Green AI An Exploratory Empirical Study. In *2022 International Conference on ICT for Sustainability (ICT4S)*; IEEE: Piscataway, NJ, USA, 2022; pp. 35–45. <https://doi.org/10.1109/ICT4S55073.2022.00015>.
27. Zemanek, V.; Hu, Y.; De Reus, P.; Oprescu, A.; Malavolta, I. Exploring the Impact of K-Anonymisation on the Energy Efficiency of Machine Learning Algorithms. In *2024 10th International Conference on ICT for Sustainability (ICT4S)*; IEEE: Piscataway, NJ, USA, 2024; pp. 128–137. <https://doi.org/10.1109/ICT4S64576.2024.00022>.
28. Zhang, J.; Shen, J.; Wang, Z.; Guo, Q.; Yan, R.; Pan, G.; Tang, H. SpikingMiniLM: Energy-efficient spiking transformer for natural language understanding. *Sci. China Inf. Sci.* **2024**, *67*, 200406. <https://doi.org/10.1007/s11432-024-4101-6>.
29. Patterson, D.; Gonzalez, J.; Holze, U.; Le, Q.; Liang, C.; Munguia, L.-M.; Rothchild, D.; So, D.R.; Texier, M.; Dean, J. The carbon footprint of machine learning training will plateau, then shrink. *Computer* **2022**, *55*, 18–28. <https://doi.org/10.1109/mc.2022.3148714>.
30. Bouza, L.; Bugeau, A.; Lannelongue, L. How to estimate carbon footprint when training deep learning models? A guide and review. *Environ. Res. Commun.* **2023**, *5*, 115014. <https://doi.org/10.1088/2515-7620/acf81b>.
31. Gupta, U.; Kim, Y.G.; Lee, S.; Tse, J.; Lee, H.-H.S.; Wei, G.-Y.; Brooks, D.; Wu, C.-J. Chasing Carbon: The Elusive Environmental Footprint of Computing. *IEEE Micro* **2022**, *42*, 37–47. <https://doi.org/10.1109/mm.2022.3163226>.
32. Wu, C.-J.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Chang, G.; Behram, F.A.; Huang, J.W.-C.; Bai, C.; et al. Sustainable AI: Environmental Implications, Challenges and Opportunities. *arXiv* **2021**, <https://doi.org/10.48550/arXiv.2111.00364>.
33. Gowda, S.N.; Hao, X.; Li, G.; Gowda, S.N.; Jin, X.; Sevilla-Lara, L. Watt for what: Rethinking deep Learning’s Energy-Performance relationship. *arXiv* **2024**, <https://doi.org/10.48550/arXiv.2310.06522>.
34. Knipper, R.A.; Mishty, K.; Sadi, M.; Santu, S.K.K. SNNLP: Energy-Efficient Natural Language Processing using spiking neural networks. *arXiv* **2024**, <https://doi.org/10.48550/arXiv.2401.17911>.
35. Narduzzi, S.; Zenke, F.; Liu, S.-C.; Dunbar, L.A. EFLOP: A Sparsity-Aware metric for evaluating computational cost in Spiking and Non-Spiking neural networks. *Neuromorphic Comput. Eng.* **2025**, *5*, 034011. <https://doi.org/10.1088/2634-4386/addee8>.
36. Rozam, N.F.; Jati, H.; Marpanaji, E. Comparative Study of Machine Learning Algorithms for DDoS Attack Detection in SDN Networks: A Carbon Emission Analysis with Hyperparameter Optimization Using Bayesian Optimization. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2025**, *15*, 530–537. <https://doi.org/10.18517/ijaseit.15.2.20767>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.