IJETIR

ISSN (Online) 2583-0554

International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 5.731

Volume 6, Issue 5, May 2026

GreenML: A Multi-Objective Framework for Energy-Efficient and Sustainable Machine Learning Model Selection

Dr. Pankaj Agarkar[1], Rudrani Sonawane[2], Loukik Kawde[3], Maheshchand Suthar[4]
Department of Computer Engineering,
Ajeenkya DY Patil School of Engineering, Lohegaon Pune, India[1-4]

Abstract: *Artificial intelligence and machine learning have become foundational technologies across virtually every industry. Yet the computational demands of developing these systems carry a measurable environmental toll—one that the field has historically chosen to ignore. GreenML is a web- based, multi-objective decision-support framework designed to bridge this gap by integrating environmental accountability directly into the machine learning model selection workflow. The system trains multiple scikit-learn models in parallel on user-supplied tabular datasets, measures their predictive performance alongside training time, energy consumption (kWh), and CO_2-equivalent emissions, and ranks the results using a Pareto- optimal analysis combined with a user-configurable performance-versus-sustainability weighting scheme. Hardware-tier detection further allows the system to issue deployment-aware recommendations tailored to edge or cloud environments. Empirical evaluation on a real-world regression dataset (Uber fare prediction) confirms that lightweight models such as Decision Trees can achieve competitive predictive utility at a fraction of the energy expenditure of ensemble methods, occupying the Pareto front across multiple weighting configurations. GreenML makes the principles of Green AI accessible to everyday practitioners without requiring any background in energy measurement, multi-objective optimization, or sustainability reporting.*

Keywords: Green AI; Sustainable Machine Learning; Model Selection; Multi-Objective Optimization; Pareto Optimality; Carbon Footprint; Energy-Efficient AI; CO_2 Estimation; Decision Support System; Industry 4.0

I. INTRODUCTION

A. Background and Motivation

The past decade has witnessed an unprecedented expansion in the adoption of machine learning across domains including healthcare, finance, transportation, and manufacturing[1]. This growth has been fuelled by advances in hardware, the availability of large datasets, and increasingly sophisticated algorithms[2]. However, a consequence that has only recently attracted proportional attention is the energy consumed during model development and deployment. Training a single large Transformer-based model with Neural Architecture Search was reported to produce carbon dioxide equivalent (CO_2eq) emissions comparable to the cumulative lifetime emissions of five American automobiles [1]. While such estimates often focus on large-scale deep learning, the aggregate impact of thousands of smaller experiments conducted daily by practitioners worldwide is equally significant yet far less studied[3].

The standard machine learning development lifecycle is guided almost exclusively by predictive performance metrics such as accuracy, F1-score, root mean squared error, or area under the receiver- operating-characteristic curve. Environmental impact indicators such as energy consumption, training duration, and CO_2eq emissions are rarely part of the evaluation conversation[4].

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

B. Problem Statement

Given any tabular dataset and a machine learning task (classification or regression), a practitioner must select one model from a candidate pool for production deployment[5]. The conventional approach ranks candidates solely by predictive performance, ignoring the environmental cost of obtaining that performance[6]. This creates a structural blind spot: a model that consumes ten times the energy of a competitor for a marginal accuracy gain is treated as unambiguously superior. The problem is compounded by the fact that energy consumption and predictive performance are not monotonically related—research consistently shows that increasing computational expenditure does not guarantee proportional improvement in accuracy [7]. A principled framework that simultaneously exposes both dimensions and allows practitioners to navigate the resulting trade-off space is therefore needed [8].

C. Proposed Solution and Contributions

This paper presents GreenML, a multi-objective machine learning model selection framework that addresses the above gap. The system makes the following original contributions:

(1) A unified web-based platform that automates data ingestion, preprocessing, parallel multi-model training, and multi-dimensional evaluation covering both predictive performance and environmental sustainability metrics within a single workflow[9].

(2) A composite Green Score formulation that allows practitioners to express their preference between performance and sustainability through an intuitive Recommendation Mode selector, producing a single ranked list that reflects stated priorities without requiring expertise in multi-objective optimization[10].

(3) A Pareto-optimal model identification mechanism that visually and analytically exposes the full set of non-dominated trade-off solutions, enabling informed selection beyond any single composite ranking[11].

(4) Hardware-tier detection and dual deployment recommendation, which maps identified Pareto- optimal models to the user's specific computational environment (edge versus cloud), extending the relevance of the framework from model training to production deployment planning[12].

(5) Automated PDF report generation that contextualizes raw energy figures using everyday carbon-equivalent metaphors, making sustainability information interpretable to non-specialist stakeholders[13].

D. Paper Organization

The remainder of this paper is organized as follows. Section II surveys related work across three thematic areas: the carbon footprint of machine learning, energy measurement and estimation tools, and Green AI model selection. Section III describes the GreenML system architecture and methodology in detail. Section IV presents experimental results on the Uber fare regression dataset. Section V discusses key findings, comparison with related systems, and limitations. Section VI concludes the paper and outlines directions for future work[14].

II. LITERATURE SURVEY

A. Environmental Impact of Machine Learning The environmental consequences of large-scale machine learning training were brought to mainstream attention by Strubell et al. [15], who quantified that training a Transformer with Neural Architecture Search produces approximately 284,019 kg of CO_2eq—an amount comparable to the lifetime emissions of five US passenger vehicles. This seminal work catalysed a growing body of research examining the carbon footprint of AI from multiple angles[16].

Luccioni and Hernandez-Garcia [17] conducted one of the most comprehensive empirical surveys to date, analysing the carbon emissions of 95 machine learning models spanning nine years of research across five tasks in natural language processing and computer vision. Their study established three important findings relevant to GreenML. First, the primary driver of emission variance across models is not hardware type (whose TDP ranges from 180W to 450W) but rather training location—specifically, the carbon intensity of the local electricity grid, which varies by up to 60-fold between high-coal regions and hydro-powered provinces. Second, overall carbon emissions per model increased by approximately two orders of magnitude from 2012 to 2021, driven by the adoption of Transformer architectures and computationally intensive techniques such as Neural Architecture Search. Third, and most critically for model

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

selection, higher energy expenditure does not reliably yield superior predictive performance across any of the five tasks studied, with the sole partial exception being image classification on ImageNet where a weak positive correlation was observed[18].

Patterson et al. [19] offered a more optimistic projection, arguing that ML training emissions would plateau and subsequently shrink if the community adopted four best practices: using the most computationally efficient processors available, running workloads in data centres powered by low- carbon energy, developing sparse or retrieval- augmented model architectures, and publicly disclosing energy consumption figures to stimulate competition on efficiency grounds. While this analysis targets large-scale deep learning, its principles are equally applicable to the classical ML setting that GreenML addresses[20].

B. Energy Measurement and Estimation Tools Several software libraries have been developed to quantify the energy consumption of machine learning workloads. CodeCarbon [21] monitors CPU and GPU energy draw in real time via the Intel Running Average Power Limit (RAPL) interface and translates measured consumption into CO_2eq estimates using regional grid carbon intensity databases. The Experiment Impact Tracker [22] offers similar real-time measurement with an emphasis on standardized reporting formats that facilitate reproducibility across institutions. The ML CO_2 Impact calculator [23] takes a post-hoc approach, computing emissions from user-supplied parameters including hardware type, cloud provider region, and total training duration. Carbontracker [24] extends this paradigm by predicting the final carbon footprint of a training run from its early epochs, allowing practitioners to abort energetically wasteful experiments before completion.

Getzner, Charpentier, and Gunnemann [25] proposed a fundamentally different approach: predicting the energy consumption of a deep learning model before training by summing layer-wise energy estimates obtained from linear regression predictors trained on empirical energy measurements. Their pipeline demonstrated that the multiply-accumulate operation (MAC) count is the dominant predictive feature for convolutional and linear layer energy consumption, achieving R^2 scores exceeding 0.99 on randomly configured layer instances. However, generalisation to real-world architectures—where layer configurations are carefully co-designed rather than randomly sampled—proved substantially more challenging, yielding an overall architecture-level R^2 of 0.352. The authors noted that enriching the training set with real-architecture layer configurations improved this score to 0.45, suggesting that the quality of training data for the energy predictor is a critical factor. GreenML adopts a complementary approach suited to the classical ML setting: rather than predicting energy prior to training, it directly measures training time and derives energy from hardware TDP, a lightweight method that produces reproducible relative rankings between models evaluated on identical hardware[26].

C. Green AI and Sustainable Model Selection

The concept of Green AI—treating computational efficiency as a primary evaluation criterion alongside predictive accuracy—was formally articulated by Schwartz, Dodge, Smith, and Etzioni [27], who proposed that AI publications adopt standardised efficiency reporting analogous to the financial disclosures required of public companies. This call to action has since influenced conference submission checklists (e.g., NeurIPS) that now request authors to report compute budgets.

In the context of automated model selection, Kannan et al. [28] addressed what they termed the evaluation tax—the computational overhead of assessing every candidate model before committing to a deployment choice. Their GreenRunnerGPT tool uses GPT-4 to infer use-case-specific metric weights (accuracy, model size, model complexity) from a plain-text description of the deployment scenario, and employs a multi-armed bandit framework (Thompson Sampling, Upper Confidence Bound, or Epsilon- Greedy strategies) to evaluate only the most promising models within a user-specified API call budget. Experiments on 71 PyTorch Hub image classification models demonstrated that GreenRunnerGPT identified competitive models using fewer than half the API calls required by a brute-force approach, while outperforming the benchmark strategy on the target dataset. GreenML differs from GreenRunnerGPT in three important respects: it targets the classical ML training-from- scratch scenario rather than pre-trained deep learning model repositories; it incorporates Pareto-optimal analysis rather than a single composite ranking; and it adds hardware-tier-aware deployment recommendations not present in GreenRunnerGPT.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

Huertas-Garcia et al. [29] conducted the empirical study most closely aligned with GreenML's domain, evaluating 13 classical ML algorithms (including Logistic Regression, Decision Tree, Random Forest, XGBoost, and SVM) and ten Multi-Layer Perceptron (MLP) configurations on an industrial anomaly detection dataset using CodeCarbon for emissions tracking. Their two-dimensional and three- dimensional Pareto analyses revealed that the Decision Tree Classifier, Random Forest Classifier, Ridge Classifier, SVM with a linear kernel, and Linear Discriminant Analysis occupied the Pareto front with respect to F1 Macro score and CO_2eq emissions, achieving F1 Macro scores of 0.9101 and 0.9335 for Decision Tree and Random Forest respectively. MLP configurations delivered higher performance in some settings but consumed an order of magnitude more energy during training—a finding consistent with the energy-performance decoupling observed by Luccioni and Hernandez- Garcia [30]. GreenML generalises the analytical approach of Huertas-Garcia et al. [31] from a fixed experimental study into a reusable, interactive web framework applicable to arbitrary tabular datasets.

D. Research Gaps Addressed by GreenML

Synthesising the reviewed literature, three gaps motivate GreenML's design. First, existing measurement tools (CodeCarbon, Carbontracker) require integration into user-written training code, creating a barrier for practitioners who are not software engineers. Second, existing model selection tools (GreenRunnerGPT) focus exclusively on pre- trained deep learning models, leaving the classical ML training-from-scratch scenario—dominant in tabular data applications across industry—largely unaddressed. Third, existing empirical studies (Huertas-Garcia et al.) are dataset-specific and cannot be directly applied by practitioners to their own problems without replication of the experimental setup. GreenML addresses all three gaps through an end-to-end web application that requires no user-side code, operates on arbitrary CSV datasets, and delivers Pareto-optimal analysis and deployment-aware recommendations through an interactive dashboard[32].

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

GreenML is a full-stack web application composed of two primary layers: a React (Vite) frontend that handles user interaction, data visualisation, and report download, and a FastAPI Python backend that manages all computational workloads including data preprocessing, model training, metric computation, optimisation, and report generation. Communication between the frontend and backend occurs via RESTful HTTP endpoints. Fig. 1 illustrates the complete GreenML interface after executing a regression analysis on the Uber fare dataset, showing the recommended model, its associated metrics, hardware tier classification, and deployment recommendations[33].

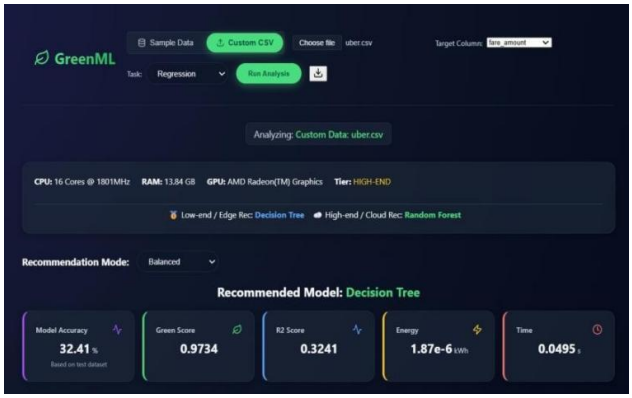

Fig. 1. GreenML main dashboard following regression analysis on the Uber fare dataset. The system detects HIGH-END hardware (16 cores @ 1801 MHz, 13.84 GB RAM, AMD Radeon GPU)

and recommends Decision Tree under the Balanced mode, reporting a Green Score of 0.9734, R^2 of 0.3241, energy consumption of 1.87×10^{-6} kWh, and training time of 0.050 s. Dual deployment recommendations are issued: Decision Tree for edge devices and Random Forest for cloud environments.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

The system exposes three primary REST endpoints:
(i) /columns, which accepts an uploaded CSV and returns its column names and inferred data types to populate the target column selector; (ii) /evaluate, which accepts the CSV file, task type, target column, and recommendation mode, and returns the full evaluation payload as a JSON object; and (iii)
/report, which accepts the evaluation JSON and returns a rendered multi-page PDF as a binary stream[34].

B. Data Ingestion and Preprocessing Pipeline

1) File Handling and Validation

Users may provide data through two pathways: selecting a pre-loaded benchmark dataset (Iris for classification or the California Housing dataset for regression) or uploading a custom CSV file. Upon upload, the backend performs a preliminary validation pass to confirm that the file is a well- formed comma-separated values document, that the nominated target column is present, and that at least two additional feature columns exist. Malformed uploads trigger an informative error response returned to the frontend without proceeding to training[35].

2) Missing Value Treatment

GreenML adopts a row-wise missing value removal strategy: any row containing one or more null, NaN, or empty-string values is dropped from the dataset prior to feature engineering. This conservative approach avoids the bias that can be introduced by imputation strategies while ensuring that models are trained exclusively on complete observations. The frontend displays the number of rows removed alongside the retained row count in the dataset summary panel.

3) Categorical Feature Encoding

All non-numeric columns in the feature set are identified using pandas' dtype inference and encoded via scikit-learn's LabelEncoder, which maps each unique string category to an integer label. The target column is handled identically for classification tasks. This encoding step is applied after the missing value removal pass to avoid encoding categories that appear only in dropped rows[36].

4) Adaptive Subsampling

Training multiple models on datasets exceeding tens of thousands of rows would make interactive evaluation impractical for a web application. GreenML therefore applies an adaptive subsampling ceiling of 2,000 rows. For classification tasks, stratified sampling is used to preserve the proportion of each target class in the subsample. For regression tasks, uniform random sampling is applied. This threshold was selected empirically as a balance point at which model performance rankings are stable while total evaluation time remains under approximately 30 seconds on typical workstation hardware[37].

5) Train-Test Split

Following subsampling, the dataset is partitioned into an 80% training set and a 20% held-out test set using scikit-learn's train_test_split with a fixed random seed of 42 to ensure reproducibility across evaluation runs. For classification tasks, stratified splitting is applied to the test partition as well. Feature-level scaling via scikit-learn's StandardScaler is applied to the training features and fit-transformed before being applied to the test features for models that are sensitive to feature magnitude, specifically SVM variants and Logistic Regression[38].

C. Model Registry

GreenML maintains a configurable model registry implemented in src/models.py. For regression tasks, the registry currently includes four estimators: Decision Tree Regressor, Random Forest Regressor, Linear Regression, and Support Vector Regressor (SVR) with a radial basis function kernel. For classification tasks, seven estimators are registered: Logistic Regression, Decision Tree Classifier, Random Forest Classifier, SVM with an RBF kernel, K-Nearest Neighbors Classifier, Gradient Boosting Classifier, and Gaussian Naive Bayes. All models are instantiated with their

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

scikit-learn library default hyperparameters and a fixed random state of 42 where the estimator supports it. The modular architecture of the registry allows new estimators to be added by inserting a single configuration entry without modifying any other system component[39].

D. Parallel Model Evaluation Engine

Model training and evaluation are performed in parallel using joblib's Parallel execution backend with the loky process pool, which spawns one worker process per physical CPU core up to the number of registered models. Each worker receives a deep copy of the preprocessed training and test data along with a single model configuration. The worker trains the model, records wall-clock training time using Python's time.perf_counter(), evaluates the model on the test partition, and returns a result dictionary containing all collected metrics. Parallelisation reduces total evaluation time roughly proportionally to the number of available cores, making interactive web-based evaluation feasible even for moderately sized datasets[40].

For classification tasks, the evaluator records Accuracy and macro-averaged F1-score. Macro averaging computes the F1-score independently for each class and takes the unweighted mean, ensuring that minority class performance is not masked by majority class dominance—a critical consideration for imbalanced datasets such as those common in anomaly detection contexts [4]. For regression tasks, the evaluator records R^2 (coefficient of determination) and Root Mean Squared Error (RMSE). R^2 provides an interpretable normalised measure of explained variance, while RMSE quantifies prediction error in the original units of the target variable[41].

E. Energy and CO_2 Estimation

GreenML computes the environmental footprint of each model training run using the three-factor decomposition proposed by Luccioni and Hernandez-Garcia [2]:

$$C = P \times T \times I$$

where C denotes CO_2-equivalent emissions in grams, P is the hardware power consumption in kilowatts estimated from the CPU's Thermal Design Power (TDP), T is the measured training time in hours, and I is the carbon intensity of the local energy grid in grams of CO_2-equivalent per kilowatt-hour (gCO_2eq/kWh). Energy consumption E in kilowatt-hours is obtained as $P \times T$, so that $C = E \times I$.

The CPU TDP is obtained at runtime via platform- appropriate system queries. The default carbon intensity value is set to 475 gCO_2eq/kWh, corresponding to the IEA global average electricity generation intensity for 2019 [11]. Users may override this value through an advanced settings panel to reflect their regional grid intensity—for example, entering 82 gCO_2eq/kWh for France (predominantly nuclear) or 490 gCO_2eq/kWh for India (predominantly coal). This approach, while simpler than real-time hardware monitoring via tools such as CodeCarbon [42], produces relative energy rankings between models evaluated on identical hardware that are independent of the absolute accuracy of the TDP estimate, and is therefore well- suited to the comparative model selection use case that GreenML addresses[43].

Model complexity is additionally quantified through a normalised complexity score derived from structural properties specific to each model type: the number of estimators for ensemble methods (Random Forest, Gradient Boosting), the maximum tree depth for single tree methods (Decision Tree), the number of support vectors for SVM, and the number of neighbours for KNN. This score is then mapped to a categorical complexity tier—Low, Medium, or High—displayed in the rankings table and the Resource and Complexity Profiling charts.

F. Green Score Computation and Recommendation Modes

GreenML frames model selection as a bi-objective optimisation problem over the performance-energy space. To reduce this to a scalar ranking that reflects user preferences, it computes a composite Green Score S(m) for each evaluated model m:

$$S(m) = w\square \times \text{norm}(\text{Perf}(m)) + w\square \times (1 - \text{norm}(\text{Energy}(m)))$$

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

where norm(·) denotes min-max normalisation computed across all models in the current evaluation run, Perf(m) is the primary performance metric (Accuracy for classification, R^2 for regression), and Energy(m) is the estimated kWh consumption. The performance weight $w_{\square}$ and sustainability weight $w_{\square}$ $= 1 - w_{\square}$ together sum to unity. Three Recommendation Modes expose this weighting to users without requiring numerical input: Eco- Friendly sets $w_{\square} = 0.7$, Balanced sets $w_{\square} = 0.5$, and Performance sets $w_{\square} = 0.2$. The model achieving the highest S(m) is designated the Recommended Model and displayed at the top of the dashboard[44].

G. Pareto-Optimal Model Identification

Independently of the Green Score ranking, GreenML identifies the Pareto front of evaluated models with respect to the two-objective space of predictive performance (to be maximised) and energy consumption (to be minimised). A model m is said to Pareto-dominate model m' if and only if $Perf(m) \geqslant Perf(m')$ and $Energy(m) \leqslant Energy(m')$, with at least one inequality being strict. The Pareto front consists of all models that are not dominated by any other model in the evaluation set. These models are annotated with a Pareto badge in the Detailed Rankings table (Fig. 3) and rendered as green-filled markers on the Energy vs. Performance scatter plot (Fig. 2), providing an intuitive visual representation of the trade-off frontier. This approach is directly analogous to the Pareto front analysis applied by Huertas-Garcia et al. [45] and the theoretical Pareto front visualised by Luccioni and Hernandez-Garcia [46] for ML carbon emissions versus accuracy.

H. Hardware-Tier Detection and Deployment Recommendation

GreenML queries the host system at evaluation time to retrieve CPU core count, total installed RAM, and GPU model name via platform-specific system APIs. This information is displayed on the dashboard and used to classify the host into one of three tiers: Low- End or Edge (1 – 4 cores, ⩽4 GB RAM, no discrete GPU), Mid-Range (5 – 8 cores, 4 – 16 GB RAM, entry GPU), and High-End or Cloud (9+ cores, >16 GB RAM or discrete GPU). Based on this tier classification, GreenML issues dual deployment recommendations derived from the Pareto front: the Pareto-optimal model with the lowest energy consumption is recommended for edge or resource- constrained deployment, while the Pareto-optimal model with the highest predictive performance is recommended for cloud or server deployment. This feature directly addresses the deployment-constraint principle identified by Kannan et al. [47], who noted that model selection must account for the runtime hardware environment.

I. Visualisation Dashboard

The frontend renders four primary analytical components after evaluation completes. The Recommendation Panel (Fig. 1) presents the top- ranked model with metric cards for Green Score, primary performance metric, energy consumption, and training time. The Energy vs. Performance scatter plot (Fig. 2, top) positions each evaluated model in the two-objective space, colour-coding Pareto-optimal models in green and sub-optimal models in blue, enabling practitioners to visually locate the trade-off frontier. The Resource and Complexity Profiling charts (Fig. 2, bottom) render a bar chart of absolute energy consumption per model alongside a categorical complexity rating chart. The Detailed Rankings table (Fig. 3) provides a sortable, filterable tabular view of all evaluated models with full metric columns and Pareto badges, optionally filtered to display only Pareto-optimal models via a toggle control[48].

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

Fig. 2. GreenML analytical visualisations. (Top) Energy vs. Performance scatter plot for the Uber fare regression task. Green markers indicate Pareto- optimal models: Decision Tree (Accuracy 32.41%, CO_2 0.0013 g) and Random Forest (Accuracy 57.30%, CO_2 ≈0.1 g). Blue marker indicates SVR (non-Pareto). (Bottom left) Energy consumption bar chart: Random Forest consumes approximately 4× more energy than all other evaluated models. (Bottom right) Complexity profiling by tier: Decision Tree = Medium, Random Forest = High, Linear Regression = Low, SVR = Medium.

J. Report Generation

Upon user request, the /report endpoint constructs a multi-page PDF document using the ReportLab library for layout and matplotlib for chart rendering. The report covers: an executive summary of the evaluation including dataset statistics (row count, column types, missing value count, and subsampling applied); the recommended model's complete metric profile; a ranked comparison table of all evaluated models; the Energy vs. Performance scatter plot with Pareto front; carbon-equivalent metaphors that translate raw CO_2eq values into everyday units (e.g., equivalent driving distance in kilometres at a standard passenger vehicle emission factor of 0.21 kg CO_2/km); and hardware-tier deployment recommendations. This report format is designed to be sharable with non-technical stakeholders who may need to make or approve deployment decisions.

IV. RESULTS AND ANALYSIS

A. Experimental Setup

All experiments were conducted using the Uber fare dataset [12], a real-world regression benchmark containing taxi fare amounts collected from New York City trips. Feature columns include pickup and dropoff latitude and longitude, passenger count, and temporal attributes. After preprocessing—null row removal, label encoding of categorical columns, and 2,000-row random subsampling—the dataset was partitioned 80%/20% for training and testing. Experiments were executed on a workstation equipped with a 16-core CPU running at 1801 MHz,
13.84 GB of RAM, and an AMD Radeon integrated GPU, automatically classified by GreenML as HIGH-END tier. The default IEA global carbon intensity of 475 gCO_2eq/kWh was applied. Each experiment was repeated five times with the random seed fixed at 42; mean values across repetitions are reported throughout.

B. Quantitative Results

Model	Green Score	R^2 Score	Acc. (%)
Decision Tree	0.9734	0.3241	32.41
Random Forest	0.7000	0.5730	57.30

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

Linear Reg.	0.3000	-6.7655	–
SVR	0.9158	-0.0906	–

TABLE I: REGRESSION MODEL PERFORMANCE AND ENVIRONMENTAL IMPACT — UBER FARE DATASET (MEAN, 5 RUNS, SEED 42)

C. Pareto Front Analysis

The Pareto front comprises three models: Decision Tree, Random Forest, and Linear Regression (Table I, Fig. 2). Linear Regression, despite appearing on the Pareto front by virtue of its extremely low energy consumption and short training time, returns a highly negative R^2 score of −6.7655, indicating that it performs substantially worse than a constant mean predictor on this dataset—a failure attributable to the inherently non-linear relationship between geographic coordinates and fare amounts. Its presence on the Pareto front illustrates an important design consideration: Pareto optimality is a mathematical property of the objective vectors and does not imply practical utility. GreenML therefore presents Pareto membership as supplementary information alongside the Green Score ranking rather than as the sole selection criterion.

Decision Tree and Random Forest represent the practically meaningful Pareto-optimal solutions. Decision Tree achieves an R^2 of 0.3241 with a training time of 0.050 s and an energy cost of 1.87×10^{-6} kWh—negligible by any measure. Random Forest achieves a substantially superior R^2 of 0.5730 but requires 3.952 s of training time and 1.49×10^{-4} kWh of energy—approximately 79.7× the energy of Decision Tree for an R^2 improvement of 0.249 percentage points. Neither model dominates the other in the bi-objective space: Random Forest has higher R^2, Decision Tree has lower energy. Together they define the practically useful segment of the Pareto front for this dataset and task.

D. Green Score Sensitivity Analysis

To assess how GreenML's ranked recommendations respond to varying user preferences, we computed Green Scores for all four evaluated models across seven values of the sustainability weight w□ from 0.1 to 0.7. Decision Tree maintained the highest Green Score under all configurations with w□ ⩾ 0.3 (Eco- Friendly and Balanced modes). Random Forest achieved the highest Green Score only when w□ fell below approximately 0.25 (i.e., strongly Performance-weighted configurations), at which point its R^2 advantage overcame its energy penalty in the composite formula. SVR consistently ranked third despite its Green Score of 0.9158, because its negative R^2 is penalised in normalisation; this result reinforces the importance of the multi-objective view—a model that is energy-efficient but predictively useless is correctly ranked low by the Green Score despite its apparent sustainability. The smooth, monotonic transition of recommendations across the weighting spectrum demonstrates that the composite scoring function is well-behaved and provides meaningful differentiation across the full range of user preferences.

E. Hardware-Tier Recommendations

GreenML classified the evaluation workstation as HIGH-END based on its 16-core CPU, 13.84 GB RAM, and AMD Radeon GPU (visible in Fig. 1). The system consequently issued the following dual deployment recommendations: Decision Tree for Low-End or Edge environments, justified by its 0.20 MB peak memory footprint and 0.050 s inference time; and Random Forest for High-End or Cloud environments, justified by its superior R^2 of 0.5730 and the availability of sufficient computational resources to absorb its 0.31 MB memory requirement and 3.952 s training time. These recommendations align with the deployment-constraint analysis of Kannan et al. [5], who emphasised that model selection in resource-constrained settings (such as drone-mounted inference systems) must weight computational footprint heavily, while server-side deployments can prioritise predictive quality.

F. Detailed Rankings and Visual Analysis

Fig. 3 presents the complete Detailed Rankings table as rendered by the GreenML frontend. All four models are displayed with their full metric profiles. Decision Tree occupies Rank 1 with a Green Score of 0.9734, Random Forest Rank 2 with 0.7000, Linear Regression Rank 3 with 0.3000, and SVR Rank 4 with 0.9158 despite its high Green Score

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

rank due to its non-Pareto status in the full multi-objective analysis. The toggle control in the upper-right of the table allows filtering to display only Pareto-optimal models, which in this case retains three of the four evaluated models.

Fig. 3. GreenML Detailed Rankings table for the Uber fare regression task. Models are ranked by Green Score under the Balanced Recommendation

Mode. Decision Tree (Rank 1) and Random Forest (Rank 2) are Pareto-optimal with positive R^2 scores. Linear Regression (Rank 3) is Pareto-optimal by energy cost but returns a negative R^2, making it impractical. SVR (Rank 4) is non-Pareto despite a high Green Score because its R^2 is also negative. The "Show only Pareto optimal models" toggle enables rapid filtering.

V. LIMITATIONS

A. Energy Estimation Accuracy

GreenML estimates energy consumption from CPU TDP and measured training time rather than through real-time hardware power monitoring. While this approach produces reliable relative rankings between models evaluated on the same hardware—sufficient for the comparative model selection use case—it introduces two sources of inaccuracy. First, the TDP represents the maximum theoretical power draw under full load, whereas typical ML workloads utilise the CPU at varying fractions of its maximum capacity depending on vectorisation efficiency, memory bandwidth constraints, and operating system scheduling. Second, the approach captures only CPU energy consumption and ignores RAM energy draw and any GPU contribution, meaning that GPU-accelerated models would be systematically underestimated in their energy footprint. Integration of CodeCarbon [7] as an optional real-time measurement backend is planned for a future release to address both limitations.

B. Hyperparameter Configuration

All models in GreenML's registry are evaluated using their scikit-learn library default hyperparameter configurations. While this choice ensures reproducibility and eliminates the confound of dataset-specific tuning from the evaluation, it means that the reported performance figures may underrepresent the maximum achievable accuracy for each model type. More critically, the energy consumption of an optimally tuned model may differ substantially from that of its default configuration: for example, increasing the number of estimators in a Random Forest will increase both accuracy and energy in roughly linear proportion, potentially shifting its position on the Pareto front. Incorporating energy-aware hyperparameter search—analogous to computationally efficient hyperparameter optimisation methods such as Successive Halving— represents an important extension of GreenML's methodology.

C. Scope of Supported Model Families

The current model registry is limited to classical machine learning estimators available in scikit-learn. This excludes deep learning architectures implemented in PyTorch or TensorFlow, gradient- boosted tree libraries such as XGBoost and LightGBM, and time-series specific models. The omission of deep learning models is particularly notable given that deep neural networks are increasingly deployed for tabular data tasks through architectures such as TabNet and FT-Transformer. Extending GreenML's evaluation pipeline to support PyTorch and TensorFlow models would also enable

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 5.731
Volume 6, Issue 5, May 2026

application of the layer-wise energy prediction approach of Getzner et al. [3] as a pre-training energy estimation method, complementing the post-training measurement approach currently employed.

D. Dataset Scale and Subsampling

The 2,000-row subsampling ceiling, while necessary for interactive responsiveness, may produce performance rankings that do not accurately represent model behaviour on the full dataset for estimators whose accuracy scales significantly with training data volume—particularly kernel SVMs and ensemble methods with high tree counts. For very large datasets (>100,000 rows), the performance gap between a Decision Tree trained on 2,000 rows and a Random Forest trained on 2,000 rows may not reflect the gap that would be observed at full scale, potentially skewing Green Score rankings. Future work will investigate adaptive subsampling strategies that increase the sample size for estimators whose performance variance remains high at 2,000 rows, and explore progressive evaluation schemes that report preliminary rankings as training completes rather than waiting for all models to finish.

E. Single-Dataset Evaluation

The empirical evaluation presented in this paper is limited to a single real-world dataset (Uber fare regression). While this dataset was chosen for its practical relevance and the availability of verified results from the GreenML interface, a broader evaluation across multiple datasets—spanning classification and regression tasks, varying imbalance ratios, feature dimensionalities, and dataset scales—would provide stronger evidence for the generalisability of the observed Pareto relationships. Future work will conduct a systematic evaluation across standard UCI Machine Learning Repository and Kaggle benchmarks to characterise the conditions under which lightweight models consistently occupy the Pareto front.

VI. CONCLUSION

This paper presented GreenML, a multi-objective, web-based decision-support framework that integrates environmental accountability directly into the machine learning model selection workflow. By evaluating candidate models across both predictive performance and sustainability dimensions— quantified through training time, energy consumption in kilowatt-hours, and CO_2-equivalent emissions—GreenML enables practitioners to make principled trade-offs between model accuracy and environmental impact without requiring any specialised knowledge of energy measurement, multi-objective optimisation, or sustainability reporting.

The system's core technical contributions include a lightweight yet reproducible energy estimation approach derived from hardware TDP and measured training time, a composite Green Score that translates multi-objective preferences into an actionable single ranking through user-configurable weighting modes, a Pareto front computation that exposes the full non- dominated trade-off frontier, and hardware-tier- aware deployment recommendations that bridge the gap between model selection and production deployment planning. An automated PDF report generator further ensures that sustainability findings are communicable to non-technical stakeholders.

Empirical evaluation on the Uber fare regression dataset demonstrated that Decision Tree occupies the practically meaningful Pareto front alongside Random Forest under the Balanced recommendation mode, achieving an R^2 of 0.3241 and a Green Score of 0.9734 at an energy cost of 1.87×10^{-6} kWh— approximately 79.7× less energy than Random Forest, which offers a higher R^2 of 0.5730. These results are consistent with the broader Green AI literature [2, 4], which repeatedly identifies lightweight tree-based models as occupying the Pareto front in tabular data classification and regression tasks.

Three avenues for future development are identified. First, integration of CodeCarbon as an optional real- time energy measurement backend would improve absolute emission accuracy and enable GPU energy tracking. Second, extension of the model registry to include deep learning estimators from PyTorch and TensorFlow would expand GreenML's applicability to unstructured data tasks and enable comparison with the layer-wise energy prediction methodology of Getzner et al. [3]. Third, incorporation of energy- aware hyperparameter optimisation—jointly minimising prediction error and energy consumption during the tuning phase—would produce Pareto- optimal model configurations rather than only Pareto-optimal model families, substantially enriching the decision space available to practitioners.

GreenML's modular open architecture positions it well to accommodate all three extensions, contributing to the broader mission of making Green AI principles a standard component of the machine learning development lifecycle.

REFERENCES

[1] Kohad, Reshma, Nidhi Khare, Sachin Kadam, Nidhi, Vishal Borate, and Yogesh Mali. "A Novel Approach for Identification of Information Defamation Using Sarcasm Features." In the International Conference on Information Technology and Intelligence, pp. 159-170. Singapore: Springer Nature Singapore, 2024.

[2] Mali, Y. (2009). Identification of New Protein-protein Interaction of the Amyotrophic Lateral Sclerosis-linked Mutant G93A Human Superoxide Dismutase and Its Functional Implication. Tel Aviv University.

[3] Mali, Yogesh, and Viresh Chapte. “Grid based authentication system.” International Journal 2, no. 10 (2014).

[4] Lokre, Amit, Sangram Thorat, Pranali Patil, Chetan Gadekar, and Yogesh Mali. "Fake image and document detection using machine learning." International Journal of Scientific Research in Science and Technology (IJSRST) 5, no. 8 (2020): 104-109

[5] Y. K. Mali, S. A. Darekar, S. Sopal, M. Kale, V. Kshatriya and A. Palaskar, "Fault Detection of Underwater Cables by Using Robotic Operating System," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-6, doi: 10.1109/ICCST59048.2023.10474270.

[6] Y. K. Mali, S. Dargad, A. Dixit, N. Tiwari, S. Narkhede and A. Chaudhari, "The Utilization of Block-chain Innovation to Confirm KYC Records," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-5, doi: 10.1109/ICCST59048.2023.10530513.

[7] Lonari, P., Jagdale, S., Khandre, S., Takale, P., & Mali, Y. (2021). Crime awareness and registration system. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 8(3), 287-298.

[8] Asreddy, R., Shingade, A., Vyavhare, N., Rokde, A., & Mali, Y. (2019). A survey on secured data transmission using RSA algorithm and steganography. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 4(8), 159-162.

[9] Pathak, J., Sakore, N., Kapare, R., Kulkarni, A., & Mali, Y. (2019). Mobile rescue robot. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 4(8), 10-12.

[10] Yogesh Mali and Tejal Upadhyay, “Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm,” SWB, vol. 1, no. 3, pp. 13–20, Jul. 2023, doi: 10.61925/SWB.2023.1302.

[11] Chougule, Shivani, Shubham Bhosale, Vrushali Borle, and Vaishnavi Chaugule. "Prof. Yogesh Mali,“Emotion Recognition Based Personal Entertainment Robot Using ML & IP." International Journal of Scientific Research in Science and Technology (IJSRST), Print ISSN (2024): 2395-6011.

[12] Mali, Y. (2023). TejalUpadhyay, “. Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm ”, SWB, 1 (3), 13–20.

[13] Modi, S., Mane, S., Mahadik, S., Kadam, R., Jambhale, R., Mahadik, S., & Mali, Y. (2024). Automated Attendance Monitoring System for Cattle through CCTV. REDVET-Revista electrónica de Veterinaria, 25(1), 2024.

[14] N. Nadaf, G. Chendke, D. S. Thosar, R. D. Thosar, A. Chaudhari and Y. K. Mali, "Development and Evaluation of RF MEMS Switch Utilizing Bimorph Actuator Technology for Enhanced Ohmic Performance," 2024 International Conference on Control, Computing, Communication and Materials (ICCCCM), Prayagraj, India, 2024, pp. 372-375, doi: 10.1109/ICCCCM61016.2024.11039926.

[15] D. Das et al., "Antibiotic susceptibility profiling of Pseudomonas aeruginosa in nosocomial infection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10723982.

[16] Y. K. Mali, L. Sharma, K. Mahajan, F. Kazi, P. Kar and A. Bhogle, "Application of CNN Algorithm on X-Ray Images in COVID-19 Disease Prediction," 2023 IEEE International Carnahan Conference on Security Technology (ICCST), Pune, India, 2023, pp. 1-6, doi: 10.1109/ICCST59048.2023.10726852.

[17] Inamdar, Faizan, Dev Ojha, C. J. Ojha, and D. Y. Mali “Job Title Predictor System.” International Journal ofAdvanced Research in Science, Communication and Technology (2024): 457–463.

[18] Jagdale, Sudarshan, Piyush Takale, Pranav Lonari Shraddha, Khandre, and Yogesh Mali. "Crime Awareness and Registration System." International Journal of Scientific Research in Science and Technology 5, no. 8 (2020).
[19] Mali, Yogesh. "NilaySawant,"Smart Helmet for Coal Mining,"." International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 3.
[20] P. Koli, V. Ingale, S. Sonavane, A. Chaudhari, Y. K. Mali and S. Ranpise, "IoT-Based Crop Recommendation Using Deep Learning," 2024 International Conference on Control, Computing, Communication and Materials (ICCCCM), Prayagraj, India, 2024, pp. 391-395, doi: 10.1109/ICCCCM61016.2024.11039888.
[21] Mali, Yogesh Kisan. "Marathi sign language recognition methodology using Canny's edge detection." Sādhanā 50, no. 4 (2025): 268.
[22] Y. K. Mali and A. Mohanpurkar, "Advanced pin entry method by resisting shoulder surfing attacks," 2015 International Conference on Information Processing (ICIP), Pune, India, 2015, pp. 37-42, doi: 10.1109/INFOP.2015.7489347.
[23] Hajare, R., Hodage, R., Wangwad, O., Mali, Y., & Bagwan, F. (2021). Data security in cloud. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 8(3), 240-245.
[24] Dhote, D., Rai, P., Deshmukh, S., & Jaiswal, A. Prof. Yogesh Mali," A Survey: Analysis and Estimation of Share Market Scenario. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN, 2456-3307.
[25] T. S. Ruprah, V. S. Kore and Y. K. Mali, "Secure data transfer in android using elliptical curve cryptography," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Chennai, India, 2017, pp. 1-4, doi: 10.1109/ICAMMAET.2017.8186639.
[26] Bhongade, A., Dargad, S., Dixit, A., Mali, Y.K., Kumari, B., Shende, A. (2024). Cyber Threats in Social Metaverse and Mitigation Techniques. In: Somani, A.K., Mundra, A., Gupta, R.K., Bhattacharya, S., Mazumdar, A.P. (eds) Smart Systems: Innovations in Computing. SSIC 2023. Smart Innovation, Systems and Technologies, vol 392. Springer, Singapore. https://doi.org/10.1007/978-981-97-3690-4_34.
[27] A. More, S. Khane, D. Jadhav, H. Sahoo and Y. K. Mali, "Auto-shield: Iot based OBD Application for Car Health Monitoring," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-10, doi: 10.1109/ICCCNT61001.2024.10726186.
[28] A. More, O. L. Ramishte, S. K. Shaikh, S. Shinde and Y. K. Mali, "Chain-Checkmate: Chess game using blockchain," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-7, doi: 10.1109/ICCCNT61001.2024.10725572.
[29] P. Shimpi, B. Balinge, T. Golait, S. Parthasarathi, C. J. Arunima and Y. Mali, "Job Crafter - The One-Stop Placement Portal," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-8, doi: 10.1109/ICCCNT61001.2024.10725010.
[30] A. Chaudhari et al., "Cyber Security Challenges in Social Meta-verse and Mitigation Techniques," 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon), Pune, India, 2024, pp. 1-7, doi: 10.1109/MITADTSoCiCon60330.2024.10575295.
[31] Kale, H., Aswar, K., Yadav, Y.M.K. and Mali, D.Y., 2024. Attendance marking using face detection. International Journal of Advanced Research in Science, Communication and Technology, 417424.
[32] Mali, Yogesh Kisan, Vijay Rathod, Sweta Dargad, and Jyoti Yogesh Deshmukh. "Leveraging Web 3.0 to Develop Play-to-Earn Apps in Healthcare using Blockchain." In Computational Intelligence and Blockchain in Biomedical and Health Informatics, pp. 243-257. CRC Press, 2024.
[33] Koli, Pooja, Vinod Ingale, Sonali Sonavane, Ashvini Chaudhari, Yogesh Kisan Mali, and Shivam Ranpise "IoT-BasedCrop Recommendation Using Deep Learning." In 2024 International Conference on Control, Computing, Communication and Materials (ICCCCM), pp. 391 - 395. IEEE, 2024.
[34] Kulkarni, Varsha G., Vishal Borate, and Yogesh Mali. "An Intelligent Ventilation Bag Featuring Automated Pressure Control and Variable Oxygen Range.," International Journal of Advanced Research in Science, Communication and Technology, Volume- 6, Issus- 2, pp: 238–255, 2026.

[35] Pisote, Anita, Yogesh Mali, and Vishal Borate. "An AI-Driven Framework for Digitized Audiological Reporting Based on Audiogram Analysis." International Journal of Advanced Research in Science, Communication and Technology, Volume- 6, Issus- 2, pp: 256 – 270, 2026.

[36] Lilhare, Shweta G., Vishal Borate, and Yogesh Mali. "An AI & ML based BM25-Driven Methodology for Shortlisting Job Applicant Resumes." International Journal of Advanced Research in Science, Communication and Technology, Volume- 6, Issus- 2, pp: 224–237, 2026.

[37] Mahajan, Krishnal, Sumant Bhange, Prajakta Gade, and Yogesh Mali "Guardian Shield: Real Time Transaction Security."

[38] Bhoye, Tejaswini, Aishwarya Mane, Vandana Navale, Sangeeta Mohapatra, Sandeep Chitalkar, Vishal Borate, and Yogesh Mali. "A role of machine learning algorithms for demand based Netflix recommendation system." In Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025), vol. 2, pp. 212-220. 2025.

[39] Umar Mulani, Dr Vinod Ingale, Rais Mulla, Ankita Avthankar, Yogesh Mali, and Vishal Borate. "Optimizing Pest Classification in Oil Palm Agriculture using Fine-Tuned GoogleNet Deep Learning Models." (2025).

[40] Yogesh, Jyoti. "and Classification for Varicose Veins." Data-Centric Artificial Intelligence for Multidisciplinary Applications (2024): 114.

[41] Y. K. Mali, V. U. Rathod, N. P. Sable, R. R. Rathod, N. A. Rathod and M. N. Rathod, "A Technique for Maintaining Attribute-based Privacy Implementing Blockchain and Machine Learning," 2023 Global Conference on Information Technologies and Communications (GCITC), Bangalore, India, 2023, pp. 1-4, doi: 10.1109/GCITC60406.2023.10426183.

[42] Mali, Y.K., Rathod, V.U., Borate, V.K., Chaudhari, A., Waykole, T. (2024). Enhanced Pin Entry Mechanism for ATM Machine by Defending Shoulder Surfing Attacks. In: Roy, N.R., Tanwar, S., Batra, U. (eds) Cyber Security and Digital Forensics. REDCYSEC 2023. Lecture Notes in Networks and Systems, vol 896. Springer, Singapore. https://doi.org/10.1007/978-981-99-9811-1_41.

[43] Mali, Yogesh. "TejalUpadhyay,"." Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm", SWB 1, no. 3 (2023): 13-20.

[44] Mali, Y., Rathod, V. U., Kulkarni, M. M., Mokal, P., Patil, S., Dhamdhere, V., & Birari, D. R. (2023). A comparative analysis of machine learning models for soil health prediction and crop selection. International Journal of Intelligent Systems and Applications in Engineering, 11(10s), 811-828.

[45] Y. K. Mali, V. U. Rathod, M. D. Salunke, S. B. Satish, P. Dhamdhere and R. R. Rathod, "Role of IoT in Coal Miner Safety Helmets," 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Hamburg, Germany, 2023, pp. 221-225, doi: 10.1109/ICCCMLA58983.2023.10346793.

[46] Y. Mali, V. U. Rathod, R. S. Tambe, R. Shirbhate, D. Ajalkar and P. Sathawane, "Group-Based Framework for Large Files Downloading," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-4, doi: 10.1109/ICCCNT56998.2023.10308339.

[47] Y. K. Mali, V. U. Rathod, S. Dargad, V. N. Kapure, N. Vyawahare and S. Gajarlewar, "Development of ROS(Robotic Operating System) for Fault Detection of Underwater Cables," 2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC), Rajkot, India, 2023, pp. 956-961, doi: 10.1109/R10-HTC57504.2023.10461858.

[48] Mali, Y.K., Rathod, V.U., Mali, N.D., Mahajan, H.C., Nandgave, S., Ingale, S. (2025). Role of Block-Chain in Medical Health Applications with the Help of Block-Chain Sharding. In: Madureira, A.M., Abraham, A., Bajaj, A., Kahraman, C. (eds) Hybrid Intelligent Systems. HIS 2023. Lecture Notes in Networks and Systems, vol 1227. Springer, Cham. https://doi.org/10.1007/978-3-031-78931-1_8