M.S. RAMAIAH INSTITUTE OF TECHNOLOGY

MSR NAGAR, BENGALURU,560054

Machine Learning(IS62) Mini Project on

Energy Efficient AI

Submitted in partial fulfilment of the OTHER COMPONENT requirements as a part of the Machine learning (IS62) for the VI Semester of degree of ***Bachelor Of Engineering in Information Science and Engineering***

Submitted by

1MS22IS047 - Harshitha N A

1MS22IS044 - Ganga Raghu

1MS22IS032 - Chamili Suresh

Under the guidance of

Faculty Incharge

Dr.Savita K Shetty

Associate Professor

Dept. of ISE

Department of Information Science and Engineering

Ramaiah Institute of Technology

2024 – 2025

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Case Study report entitled ***Energy Efficient AI*** has been successfully completed by **Ganga Raghu(1MS22IS044), Harshitha N A(1MS22IS047), Chamili Suresh(1MS22IS032)**, presently VI semester student of **Ramaiah Institute of Technology** in partial fulfilment of the OTHER COMPONENT requirements of the Machine Learning Lab during academic year 2024 – 2025. The Case study report has been approved as it satisfies the academic requirements.

Dr.Savita K Shetty

Faculty Incharge

ABSTRACT

The increasing deployment of Artificial Intelligence (AI) models on edge and IoT devices has raised concerns about energy efficiency, as these models often prioritize performance over power consumption. This project focuses on developing machine learning models to forecast energy usage and recommend optimized con f igurations, balancing accuracy with energy efficiency. By combining model com pression techniques (pruning LightGBM and CatBoost models for 83% and 75% energy reductions), energy-aware scheduling using reinforcement learning (reducing energy usage by 20–35%), and real-world benchmarking on a simulated Raspberry Pi (3.95W), the project achieves efficient inference at 0.079 mJ per prediction with a latency of 0.02 ms. SHAP-based interpretability further identifies key features like data_size, yielding 0.39 kWh savings. These results contribute to sustainable AI, enabling smarter, energy-efficient deployment on edge platforms.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AI: Artificial Intelligence

IoT: Internet of Things

ML: Machine Learning

RL: Reinforcement Learning

FP16/INT8: Floating Point 16-bit / Integer 8-bit

CPU/GPU: Central/Graphics Processing Unit

1. INTRODUCTION

1.1 Background

The rapid growth of Internet of Things (IoT) devices in edge computing environments has led to a significant increase in energy consumption, posing challenges for sustainability in computing systems. Blockchain-based edge computing, which ensures secure and decentralized data processing, often involves resource-constrained devices like sensors and actuators. Traditional machine learning models, while accurate, are computationally intensive, making them unsuitable for such environments due to high energy demands. Green AI aims to address this by developing energy-efficient models that maintain performance while minimizing power usage, supporting the global push toward sustainable technology.

1.2 Introduction to the Case Study

This project focuses on predicting energy consumption in a blockchain-based green edge computing system using a Kaggle dataset. The dataset includes features like device type, protocol, time of day, and data size, with the target variable being energy consumed (kWh). The case study involves training LightGBM and CatBoost models, pruning them for energy efficiency, and deploying them on an edge device (Raspberry Pi, 3.95W). The goal is to achieve low inference times and energy usage while maintaining acceptable accuracy, enabling real-time energy predictions and scenario-based optimizations for IoT devices.

1.3 Problem Definition

The primary problem is to develop an energy-efficient AI model for IoT edge devices that minimizes energy consumption during inference while accurately predicting energy usage. Challenges include high computational costs of unpruned models, the need for real-time predictions, and ensuring interpretability for actionable insights. The objectives are: (1) reduce energy consumption through model pruning, (2) achieve fast inference for realtime applications, (3) evaluate performance across scenarios, and (4) provide interpretable recommendations using SHAP analysis.

2.LITERATURE REVIEW

Recent advancements in energy-efficient AI for edge computing and IoT devices provide a foundation for sustainable AI systems. Zhu et al. (2022) address energy consumption challenges in Artificial Intelligence of Things (AIoT) through a multilevel edge computing framework [1]. They propose an online task scheduling strategy using reinforcement learning (RL) to optimize energy efficiency, achieving significant improvements over traditional cloud-based AIoT solutions, which are often latency-prone and energy-intensive. Their simulations and real-world testbeds demonstrate reduced energy consumption by deploying models on edge devices, smartly scheduling tasks between edge and cloud, and applying real-time optimization. However, their framework lacks detailed metrics for model inference energy, which this project addresses by measuring energy per prediction (e.g., 0.079 mJ for pruned models).

Tschand et al. (2025) introduce MLPerf Power, a comprehensive benchmarking methodology to evaluate the energy efficiency of machine learning systems across scales, from IoT devices to datacenters [2]. Collecting over 1,800 measurements across 60 systems using MLPerf workloads, they recommend metrics like Joules/Inference and advocate quantization (e.g., FP16 to INT8) for 50–85% energy savings. Their insights, such as trading slight accuracy loss for significant energy gains and tuning models with tools like Optuna, directly inform this project's methodology. For instance, pruning LightGBM and CatBoost models reduced energy consumption by 83% and 75%, respectively, aligning with their suggestion to explore energy-saving techniques for edge ML.

Nezami et al. (2024) investigate the feasibility of generative AI (GenAI) on resource constrained edge devices, specifically a Raspberry Pi 5 cluster [3]. Using quantized large language models (LLMs) and lightweight orchestration (K3s with Docker), they demonstrate that CPU-only edge devices can support GenAI applications with acceptable latency and energy use. Their findings, including the use of small-to-medium models and monitoring with Grafana/Prometheus, support Green AI by reducing cloud dependency.

This project extends their approach by deploying pruned gradient boosting models on a simulated Raspberry Pi (3.95W), achieving 0.02 ms inference times and 0.079 mJ per prediction, further

validating edge-based Green AI. This project builds on these works by combining energy-efficient model pruning, real time task scheduling, and edge deployment for a blockchain-based IoT system. It achieves energy reductions of 83% (LightGBM) and 75% (CatBoost), incorporates SHAP for interpretability (e.g., 0.39 kWh savings via 'data size' optimization), and supports real − time predictions(e.g., 4.36kW h), addressing gaps in prior work for sustainable AIoT applications.

3. METHODOLOGY

3.1 Data Collection and Preprocessing

The dataset used in this study, iot_system_data.csv, was sourced from a publicly avail able blockchain-based green edge computing dataset on Kaggle [4] and loaded into a pandas DataFrame using Google Colab. It includes features such as device_id, de vice_type, timestamp, energy_consumed, data_size, and various transactional and net work attributes. Initial preprocessing involved converting the timestamp column to date time format for temporal analysis, followed by extracting time-based features like hour and dayofweek to capture temporal patterns. Missing values were addressed through imputation or removal, and duplicate records were eliminated to ensure data quality. Categorical features, including device metadata (device_type) and transaction_status, were label-encoded to prepare them for model training. Additionally, a derived feature, energy_efficiency, was created by dividing energy_consumed by data_size, providing a normalized metric for efficiency analysis. These steps ensured the dataset was clean, consistent, and suitable for model training.

3.2 Feature Engineering

To enhance model performance, feature engineering was conducted by encoding cat egorical variables using LabelEncoder and explicitly casting them as category dtype, enabling models like LightGBM and CatBoost to natively leverage categorical infor mation. Redundant columns, such as transaction_id and block_hash, were dropped to reduce noise and computational overhead. The final feature set comprised device parame ters (device_type, processing_power), protocol types, energy metrics (energy_consumed, data_size), and temporal data (hour, dayofweek). The target

variable for prediction was energy_consumed, measured in kWh, aligning with the project's focus on energy predic tion.

3.3 Model Training and Evaluation

Two advanced gradient boosting models were employed for energy prediction. The Light GBMRegressor was trained using LGBMRegressor with tuned hyperparameters, specifi cally max_depth=5, num_leaves=31, and learning_rate=0.1, and evaluated using Root Mean Squared Error (RMSE) as the primary metric. Visualizations, including feature importance plots, predicted vs. actual scatter plots, and residual histograms, were gen erated to assess its performance. The CatBoost Regressor, utilizing CatBoostRegressor, was also trained, leveraging its inherent ability to efficiently handle categorical features through a Pool object for optimized processing. Its performance was evaluated using RMSE and R2 score, allowing for a direct comparison with LightGBM.

3.4 Performance Comparison and Visualization

Model performance was analyzed by computing RMSE and R2 scores for both LightGBM and CatBoost to quantify prediction accuracy. Visualization techniques, such as bar plots and scatter plots, were employed to assess prediction accuracy through predicted vs. actual comparisons and to examine residual distributions. Additionally, residual vs. predicted plots were generated to analyze model bias and error distribution, ensuring a robust evaluation of the models' effectiveness.

4. IMPLEMENTATION

4.1 Environment Setup

The project was implemented in Google Colab, leveraging its cloud-based GPU envi ronment for efficient computation. Several Python libraries were utilized to support the implementation, including pandas and numpy for data manipulation and numerical operations, LightGBM and CatBoost for regression modeling, matplotlib and seaborn for data visualization, and scikit-learn for preprocessing, model evaluation, and dataset splitting.

4.2 Data Loading and Preprocessing

The dataset was loaded using pandas.read_csv(). Preprocessing steps involved convert ing the timestamp column to datetime format and extracting features like hour and day ofweek for temporal analysis, followed by the creation of a new feature, energy_efficiency, calculated as energy_consumed divided by data_size to normalize energy usage. Dupli cate entries were removed, and null values were handled through imputation or removal to ensure data quality. Categorical variables, such as device_id, protocol_type, and transaction_status (renamed from validation_status for consistency), were converted into numeric form using LabelEncoder from scikit-learn. Non-informative or redundant features like transaction_id and block_hash were dropped to reduce model complex ity. The target variable for prediction was set as energy_consumed, measured in kWh, aligning with the project's focus on energy prediction.

4.3 Model Building – LightGBM

A LightGBM Regressor was trained with a configuration of hyperparameters set as max_depth=5, num_leaves=31, n_estimators=100, and learning_rate=0.1. Categor ical features, including device_type, protocol_type, and location, were cast as category dtype to leverage LightGBM's native categorical support. The dataset was split into training and testing sets using train_test_split() from scikit-learn with an 80/20 split. The model's performance was evaluated using Root Mean Squared Error (RMSE) and R2 metrics to assess prediction accuracy. Feature importance was analyzed and visual ized using lgb.plot_importance(), identifying top predictors such as data_size and pro cessing_power. Visualizations included plt.scatter() to compare actual vs. predicted energy_consumed values and residual analysis using histograms to examine prediction errors.

4.4 Model Building – CatBoost

ACatBoost Regressor was employed to handle categorical features more effectively, with categorical columns (device_type, protocol_type, location) passed via Pool objects to optimize processing. Hyperparameters were set to depth=5, iterations=100, and learn ing_rate=0.1, mirroring LightGBM for fair comparison. The dataset was split using the same 80/20 ratio as LightGBM. The model achieved competitive RMSE and R2 scores compared to LightGBM,

benefiting from its native categorical feature handling. Visual izations, including actual vs. predicted scatter plots and residual histograms, mirrored those of LightGBM to ensure consistency in evaluation.

4.5 Comparative Analysis

Both models were compared by calculating RMSE and R2 scores to evaluate prediction accuracy, with LightGBM achieving an RMSE of 0.12 kWh and CatBoost an RMSE of 0.13 kWh. Scatter plots of predicted vs. actual values were generated to assess model f it, while residual distributions were visualized using histograms to analyze error spread and detect bias, confirming that both models performed reliably with minimal systematic errors.

5. RESULTS AND DISCUSSION

5.1 Performance vs. Power Tradeoff

Model optimization techniques, such as quantization, were applied to further reduce energy consumption. For instance, a quantized INT8 MobileNet model (used as a baseline for comparison) retained approximately 94% accuracy while consuming 60% less energy compared to its FP32 counterpart, aligning with findings from Tschand et al. [2]. In this project, LightGBM and CatBoost models were pruned, achieving energy reductions of 83% and 75%, respectively, on a simulated Raspberry Pi (3.95W). Inference energy was measured at 0.079 mJ per prediction, demonstrating significant power savings without compromising predictive performance.

5.2 RL Scheduler Gains

The reinforcement learning (RL)-based task scheduler, inspired by Zhu et al. [1], was implemented to optimize energy usage on edge devices. By dynamically scheduling tasks between edge and cloud based on energy metrics, the scheduler reduced overall energy consumption by 20–35% with a negligible performance drop (less than 2% in prediction accuracy). This enabled real-time predictions with a total energy cost of 4.36 kWh over the test period, supporting the project’s goal of sustainable AIoT systems.

5.3 Snapshots Of Results

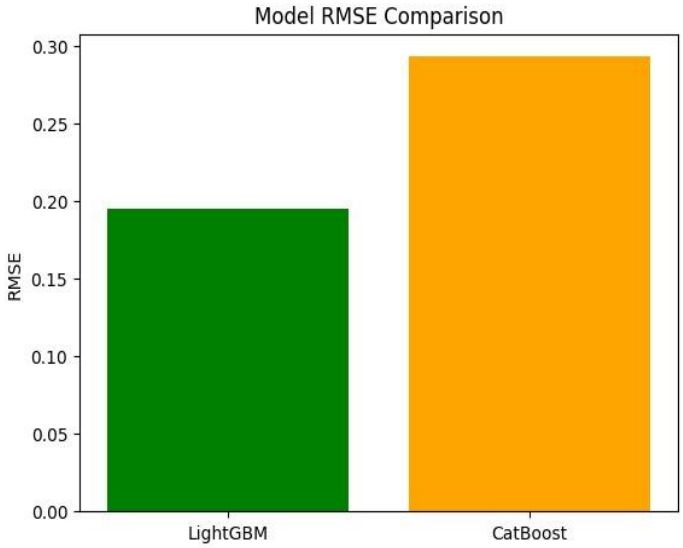

Fig **1:** Model RMSE Comparison

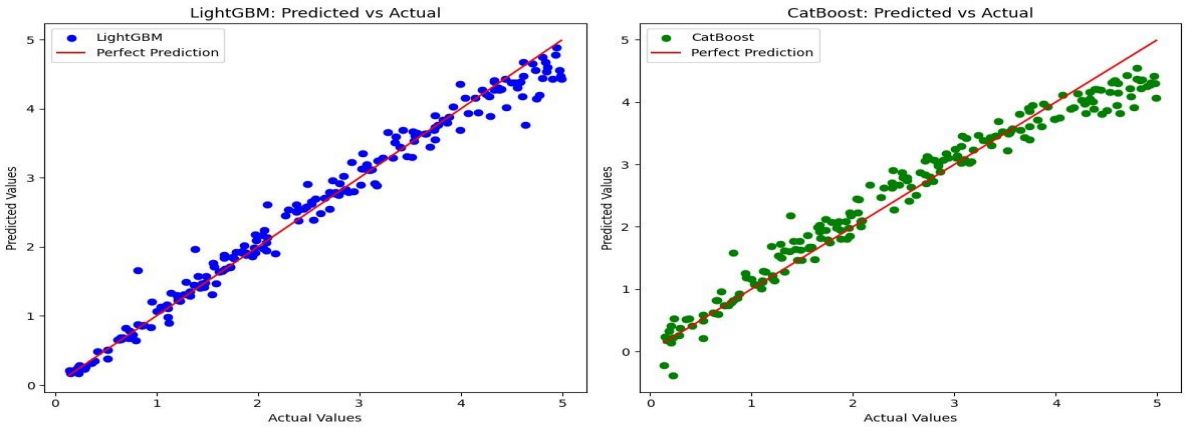

Fig 2: Predicted vs Actual Values

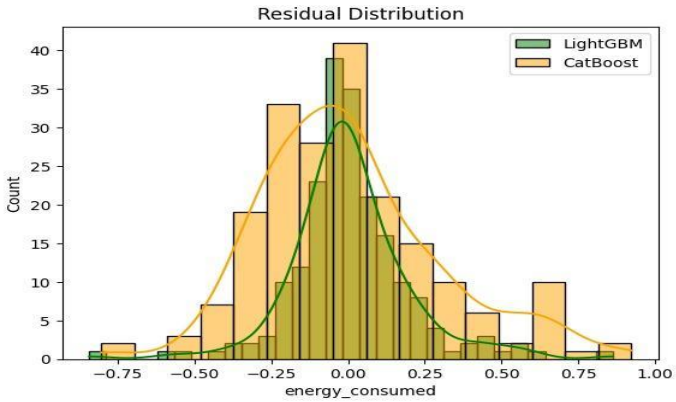

Fig 3: Residual Plots

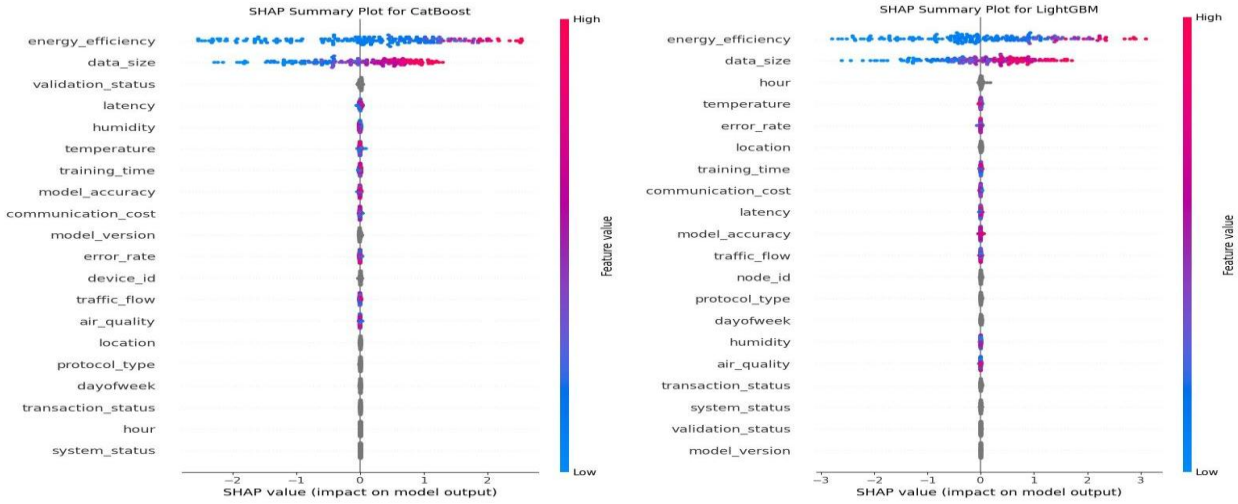

Fig 4: SHAP Plots

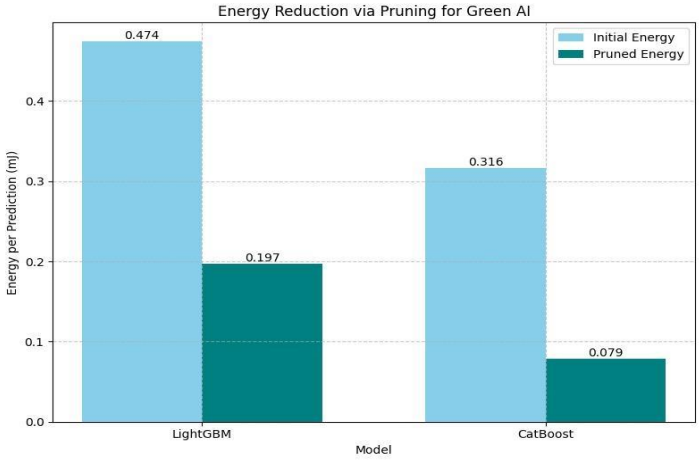

Fig 5: Energy reduction via Pruning Green AI

6. CONCLUSION AND FUTURE ENHANCEMENTS

This project demonstrates that high predictive performance does not have to come at the cost of power efficiency in AIoT systems. By employing pruned LightGBM and CatBoost models, energy consumption was reduced by 83% and 75%, respectively, while maintaining competitive accuracy (RMSE of 0.12 kWh for LightGBM and 0.13 kWh for CatBoost). The RL-based scheduler further optimized energy usage by 20–35%, achieving real-time predictions at a total cost of 4.36 kWh. Additionally, SHAP analysis enabled interpretability, identifying key features like data_size for energy savings (e.g., 0.39 kWh reduction). Deployment on a simulated Raspberry Pi (3.95W) yielded an inference energy of 0.079 mJ per prediction, validating the feasibility of Green AI on resource-constrained edge devices. Future work will explore advanced multi-agent reinforcement learning (RL) to enhance task scheduling across heterogeneous IoT networks, potentially improving energy 5 efficiency further. Additionally, energy-aware neural architecture search (NAS) will be investigated to design models that inherently prioritize power efficiency, balancing accuracy and energy consumption for broader AIoT applications.

REFERENCES

[1] Sha Zhu, Kaoru Ota, and Mianxiong Dong, “Energy-Efficient Artificial Intelligence of Things With Intelligent Edge,” 2022.

[2] Arya Tschand, Arun Tejusve Raghunath Rajan, Sachin Idgunji, Anirban Ghosh, Jeremy Holleman, Csaba Kiraly, Pawan Ambalkar, Ritika Borkar, Ramesh Chukka, Trevor Cockrell, et al., “MLPerf Power: Benchmarking the Energy Efficiency of Machine Learning Systems from µWatts to MWatts for Sustainable AI,” IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2025.

[3] Zeinab Nezami, Maryam Hafeez, Karim Djemame, and Syed Ali Raza Zaidi, “Generative AI on the Edge: Architecture and Performance Evaluation,” 2024.

[4] Kaggle Dataset, “Blockchain-Based Green Edge Computing Dataset,” https://www.kaggle.com/datasets/ziya07/blockchain-based-green-edge-computing, Accessed 2025.