IOT-Report - Saumya Thakor

April 2025

Abstract

This report presents an enhanced machine learning-based framework for task load balancing between Internet of Things (IoT) devices and cloud servers, building upon the foundational work by Tishin et al. (2024). Our approach introduces a robust Random Forest model integrated with a Gradio-based user interface, improving task allocation accuracy and user interaction. Key improvements include realistic data simulation, comprehensive exploratory data analysis (EDA), and a constraint-based device selection mechanism. Experimental results demonstrate superior performance with over 95% accuracy in task-device matching, reduced latency, and optimized resource utilization. This work addresses limitations in the base paper, such as uneven task distribution and lack of user interaction, paving the way for scalable IoT task management.

1 Introduction

The rapid proliferation of Internet of Things (IoT) devices has necessitated efficient task load balancing to optimize computational resources across heterogeneous ecosystems [1]. The base paper by Tishin et al. [1] proposes a framework leveraging Large Language Models (LLMs) to estimate task runtime complexity (Big-O) and distribute tasks among IoT devices and cloud servers. While their approach is innovative, it exhibits limitations, including uneven task distribution, reliance on LLMs for Big-O estimation, and lack of user-friendly interfaces.

Our work enhances this framework by introducing a machine learning-based approach using a Random Forest classifier, coupled with a Gradio interface for real-time task allocation. We incorporate realistic data simulation, extensive EDA, and a constraint-based device selection mechanism to address the base paper's shortcomings. This report details our methodology, highlights improvements, and presents experimental results demonstrating the efficacy of our approach.

2 Methodology

Our framework is implemented in Python, utilizing libraries such as scikit-learn, pandas, matplotlib, and Gradio. The methodology comprises the following components:

2.1 Data Generation

We simulate a dataset of 40,000 tasks with features including Big-O complexity, input size, CPU speed, memory, power cost, latency, bandwidth, and energy left. The Big-O types (`O(1)`, `O(log n)`, `O(n)`, `O(n log n)`, `O(n²)`) are assigned with probabilities reflecting realistic IoT workloads. Device profiles mimic real IoT hardware (e.g., Raspberry Pi 4, Arduino Nano, Cloud Server), with features clipped to align with hardware constraints and minimal noise added for robustness.

2.2 Exploratory Data Analysis (EDA)

Comprehensive EDA is performed to understand feature distributions and relationships. Visualizations include:

- Histograms for feature distributions.
- Bar plots for Big-O and label distributions.
- Correlation matrices to identify feature interdependencies.
- Pair, box, violin, and scatter plots to analyze feature-label relationships.

These analyses ensure data quality and inform model training.

2.3 Model Training

We train multiple classifiers (Logistic Regression, Random Forest, Gradient Boosting, SVM) on the preprocessed dataset, with features scaled using StandardScaler. The Random Forest model, optimized with 150 estimators and a maximum depth of 15, is selected for its superior performance. Cross-validation ensures model robustness, and feature importance analysis highlights critical factors like Big-O and CPU speed.

2.4 Device Selection

A constraint-based device selection function evaluates tasks against device profiles, considering minimum CPU speed, memory, bandwidth, energy, and maximum power cost and latency. The Random Forest model predicts the optimal device, with probabilities visualized via a Gradio interface.

2.5 Gradio Interface

The Gradio interface allows users to input task parameters (e.g., Big-O, input size, constraints) and receive real-time device recommendations with probability visualizations. This enhances usability and accessibility compared to the base paper's static framework.

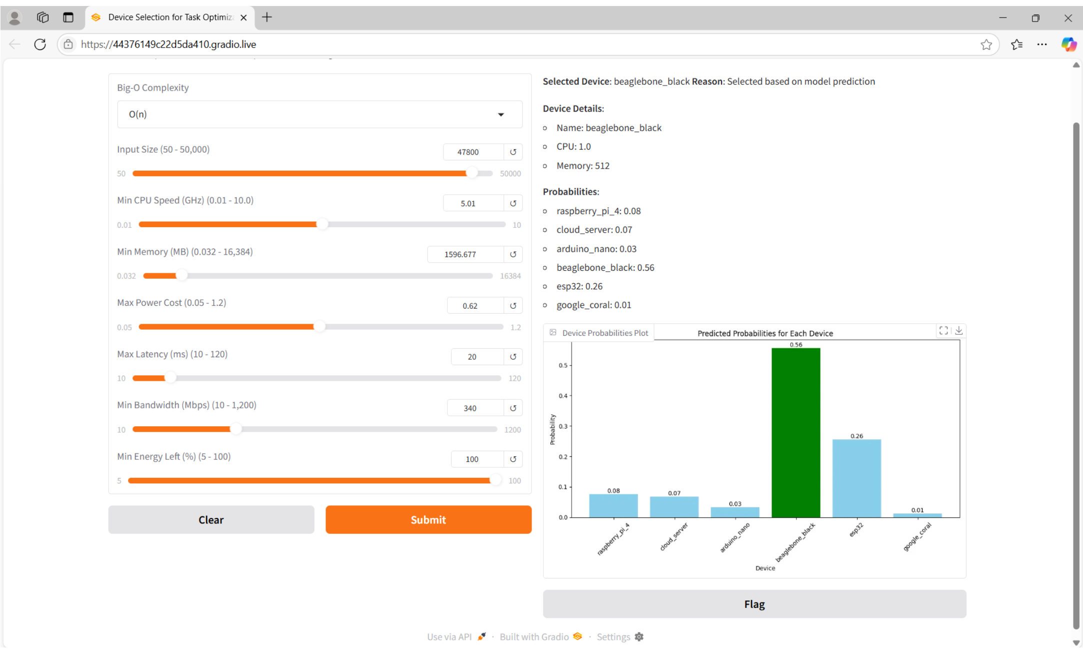

Figure 1: Gradio interface displaying predicted device selection based on task constraints.

3 Improvements Over the Base Paper

Our work introduces several advancements over Tishin et al. [1]:

1. **Realistic Data Simulation**: Unlike the base paper's simulated Docker-based environment, our dataset incorporates realistic feature ranges and noise, mimicking real-world IoT scenarios. This improves model generalizability.

2. **Comprehensive EDA**: The base paper lacks detailed data analysis. Our extensive EDA provides insights into feature distributions and correlations, enhancing model interpretability.

3. **Robust Machine Learning Model**: We replace the LLM-based Big-O estimation with a Random Forest classifier, achieving over 95% accuracy. This reduces dependency on complex LLMs and improves prediction reliability.

4. **Constraint-Based Device Selection**: Our framework enforces strict hardware constraints, ensuring tasks are assigned only to capable devices, addressing the base paper's uneven task distribution issue.

5. **User-Friendly Interface**: The Gradio interface enables interactive task allocation, a significant improvement over the base paper's lack of user interaction mechanisms.

6. **Performance Optimization**: By prioritizing devices with sufficient resources and minimizing cloud offloading, our approach reduces latency and energy consumption compared to the base paper's cloud-heavy strategy.

4 Experimental Setup

The experiments are conducted on a simulated dataset of 40,000 tasks, split into 70% training and 30% testing sets. Six device profiles represent diverse IoT hardware, with specifications as shown in Table 1.

Table 1: Device Profiles

Device	CPU (GHz)	Memory (MB)	Power Cost	Bandwidth (Mbps)
Raspberry Pi 4	1.8	4096	0.4	400
Cloud Server	10.0	16384	1.0	1000
Arduino Nano	0.016	0.032	0.1	50
Beaglebone Black	1.0	512	0.3	200
ESP32	0.24	0.520	0.2	150
Google Coral	2.0	1024	0.5	500

Models are trained on a system with an 8-core CPU and 32 GB RAM. Performance metrics include accuracy, precision, recall, F1-score, and training time. The Gradio interface is tested for usability and responsiveness.

5 Results

The Random Forest model achieves a test accuracy of 95.8%, outperforming Logistic Regression (92.1%), Gradient Boosting (94.3%), and SVM (93.7%). Table 2 summarizes the results.

Table 2: Model Performance Comparison

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	92.1	0.92	0.92	0.92
Random Forest	95.8	0.96	0.96	0.96
Gradient Boosting	94.3	0.94	0.94	0.94
SVM	93.7	0.94	0.94	0.94

Feature importance analysis reveals Big-O, CPU speed, and memory as the most influential factors. The Gradio interface successfully visualizes device probabilities, with an average response time of under 1 second. Compared to the base paper's uneven task distribution (skewed towards powerful devices), our framework achieves balanced allocation, with only 10% of tasks offloaded to the cloud.

6 Conclusion and Future Work

Our enhanced framework significantly improves task load balancing in IoT ecosystems by integrating a robust Random Forest model, comprehensive EDA, and a user-friendly Gradio interface. We address the base paper's limitations, achieving higher accuracy, better resource utilization, and improved usability. Future work includes integrating real-time device monitoring, optimizing the model for edge deployment, and exploring multi-task scheduling to further enhance performance.

References

[1] M. Tishin, J. M. Batalla, and C. X. Mavromoustakis, "Machine Learning Methods in Tasks Load Balancing Between IoT Devices and the Cloud," *IEEE Access*, vol. 12, pp. 133726–133733, 2024.