Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
FAIR Future Artificial Intelligence Research

2nd FAIR Workshop in Green-aware AI

Towards Interpretable Energy Estimation for Edge AI Applications

Riccardo Cantini, Alessio Orsino, Domenico Talia, Paolo Trunfio

University of Calabria

March 24, 2025

WP9.3 Task 1: *main concepts & goals*

- **Environmental sustainability**
  - Optimize energy use and resource management of AI applications.
  - Design energy-aware solutions to facilitate harnessing AI capabilities in low-resource settings.

- **Social sustainability**
  - Promote inclusivity and fairness of AI systems.
  - Uncover and mitigate biases in AI systems, to ensure ethical practices and support social equity.

WP9.3 Task 1: *main research activities*

- **Environmental sustainability**
  - **Interpretable energy estimation for edge AI applications**
  - Cross-architecture knowledge distillation of LLMs for efficient deployment at the edge
  - Efficient test-time adaptation on edge devices

- **Social sustainability**
  - Jailbreak-based adversarial analysis to uncover hidden biases and stereotypes in LLMs
  - Analysis of discriminatory tendencies in domain-specific LLMs

Introduction to Edge AI: *key advantages and challenges*

- **Edge AI** refers to AI and ML models executed directly on edge devices instead of centralized cloud servers.

- **Advantages**
  - Reduces latency, enabling (near)real-time decision-making.
  - Reduces reliance on cloud computing, improving privacy and security.
  - Optimizes bandwidth usage by processing data locally.

- **Challenges**
  - Limited computational resources (CPU, memory, storage).
  - Energy constraints, as edge devices are often battery-powered.
  - **Complexity of energy consumption estimation and lack of interpretability**.

Research objective

- Develop an **interpretable energy estimation model** that provides:
  - Accurate energy consumption predictions for edge AI/ML workloads.
  - Fine-grained, actionable insights of the consumption estimates.

- This will help fostering **sustainability** in edge AI settings:
  - It supports the adoption of green-aware practices such as intelligent scheduling strategies and green-aware NAS (neural architecture search).
  - Interpretability allows developers and engineers to gain actionable insights into optimizing algorithms and deployment configurations to minimize energy costs.

Problem formulation

- We define the energy consumption $\mathcal{E}(a, d)$ of an ML application, described by an algorithm a and a dataset d within a fixed distributed architecture, as the sum of three components:

$$\mathcal{E}(a, d) = \mathcal{E}_{comp}(a, d) + \mathcal{E}_{data}(a, d) + \mathcal{E}_{comm}(a, d)$$

  - $E_{comp}(a, d)$ is the fraction due to **computation**, $E_{data}(a, d)$ is the fraction due to **data access**, and $E_{comm}(a, d)$ is the fraction due to **communication** over the network.

- We frame **interpretable energy estimation** as a two-step process:
  1. Estimating energy components by leveraging supervised learning techniques, unveiling the patterns linking energy consumption to algorithm characteristics and input data.
  2. Use *eXplainable AI* (XAI) to provide insightful interpretations of the consumption estimates to the user.

Modeling energy via learnable proxies

- We represent a given pair (a, d) as a feature vector $\mathcal{F}$ describing algorithm and dataset characteristics.

- We rewrite the total energy consumption as the weighted sum of learnable proxies (i.e., measurable counters) for computation ($\mathcal{P}_{comp}$), data access ($\mathcal{P}_{data}$), and communication ($\mathcal{P}_{comm}$).

$$\mathcal{E}(F) = \beta_{comp} \cdot \mathcal{R}_{comp}(\mathcal{F}) + \beta_{data} \cdot \mathcal{R}_{data}(\mathcal{F}) + \beta_{comm} \cdot \mathcal{R}_{comm}(\mathcal{F})$$

- Where:
  - $\mathcal{F}$ represents the feature vector describing a and d.
  - $\mathcal{R}_{comp}, \mathcal{R}_{data}, \mathcal{R}_{comm}$ are regression models predicting energy proxies from $\mathcal{F}$, i.e., $\mathcal{P}_x = \mathcal{R}_x(\mathcal{F})$.
  - $\beta_{comp}, \beta_{data}, \beta_{comm}$ are experimentally determined scaling factors converting proxies into energy values.

Interpreting energy estimates with XAI

- **Interpretable energy estimation** implies understanding the relationship between input features $\mathcal{F}$ and the total energy consumption $\mathcal{E}(F)$.

- Since overall consumption depends on the proxies $\mathcal{P}_x$, which are outputs of regression models $\mathcal{R}_x(\mathcal{F})$, we frame the problem of interpretability as explaining the predictions of these models.

- Such explanations, achieved by applying feature attribution techniques, enables to trace the total energy consumption back to the original input features.

- This approach not only estimates overall consumption but also delivers component-wise, feature-level explanations for why a specific algorithm-dataset combination incurs a particular energy cost.

Proposed methodology

- **Training of the ML Models**
  - Training data is collected during the execution of various ML algorithms across diverse datasets via application monitoring.
  - A set of platform-specific regression models, $\mathcal{R}_x$, is trained for each proxy $\mathcal{P}_x$ using collected data.
  - A classifier $\mathcal{C}$ is trained to filter out (a, d) pairs whose execution would violate a set of predefined constraints:
    - user-defined (e.g., max execution time)
    - platform-specific (e.g., max available memory)

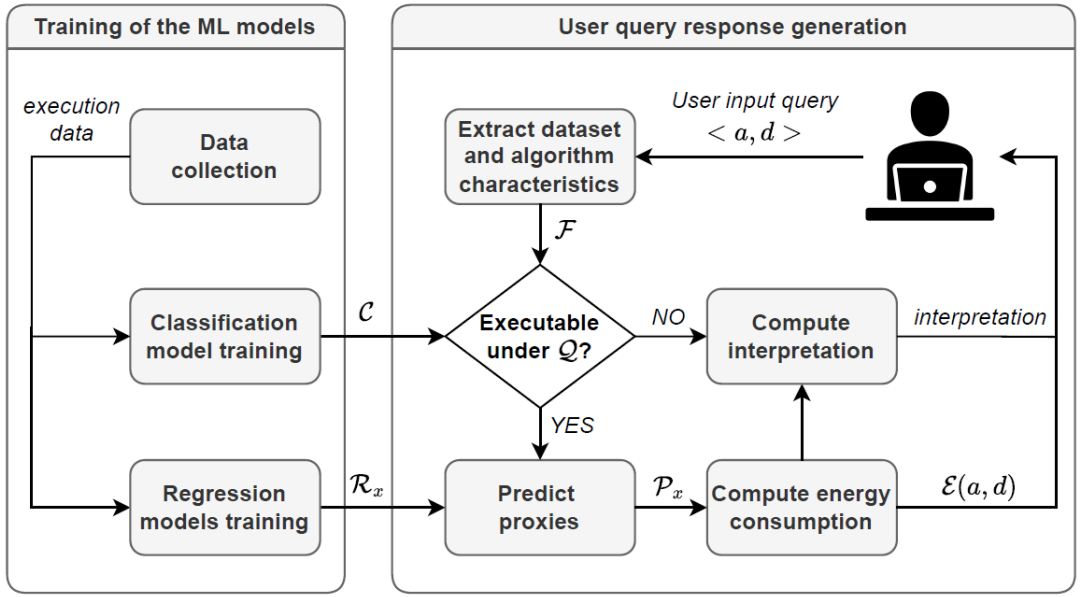


Fig. 1. **Execution flow of the proposed methodology**: (i) the left box shows the training of ML models using execution data; (ii) the right box shows how the energy estimate is computed and interpreted in response to a user query.

Proposed methodology

- **User Query Response Generation**
  - The classifier $\mathcal{C}$ first evaluates whether the application satisfies a set of predefined constraints $\mathcal{Q}$.
    - $\mathcal{Q} = 0 \rightarrow$ the application is rejected and a feature-level explanation is provided to the user.
    - $\mathcal{Q} = 1 \rightarrow$ the application is accepted and proceeds to regression and energy estimation.
  - Proxies $\mathcal{P}_x$ are estimated and aggregated using β_x factors to obtain the overall energy consumption.
  - Local explanations are provided for all regression estimates $\mathcal{P}_x = \mathcal{R}_x(\mathcal{F})$ explaining how each feature $f \in F$ affects the energy estimate.

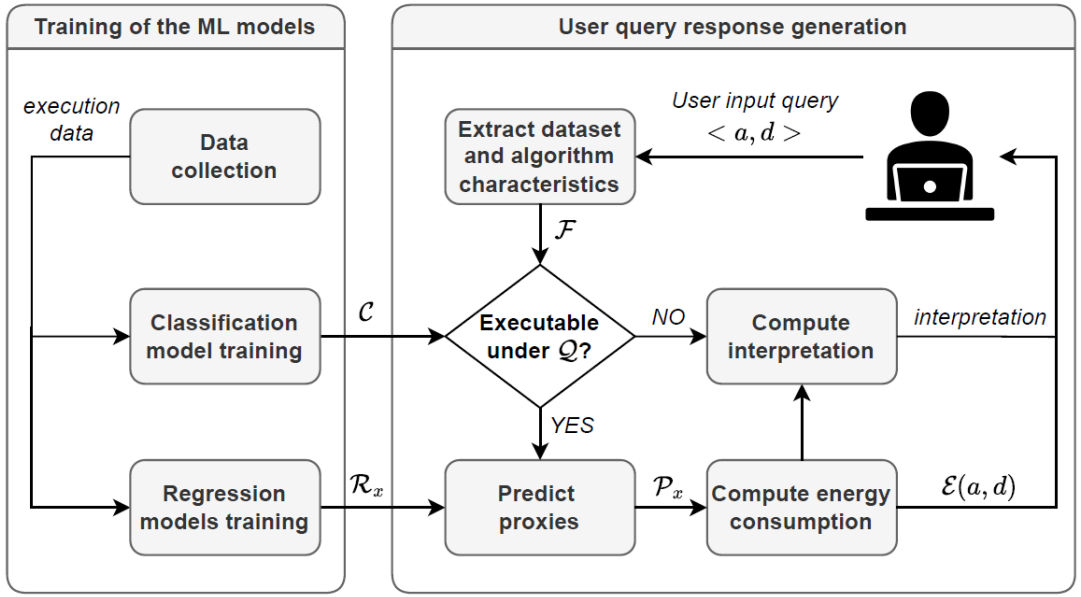


Fig. 1. **Execution flow of the proposed methodology**: (i) the left box shows the training of ML models using execution data; (ii) the right box shows how the energy estimate is computed and interpreted in response to a user query.

Experimental evaluation

- **Experimental setting**
  - The experiments were executed on a **Raspberry Pi 4 Model B** featuring a 64-bit quad-core Cortex A72 processor and equipped with 4 GB of LPDDR4 RAM.
  - We utilized scikit-learn implementation of the following algorithms: *Decision Tree*, *K-Means*, *Logistic Regression*, *K-Nearest Neighbors*, *Principal Component Analysis*, *Linear Regression*, and *Random Forest*.
  - Experiments were run using a single-core setting, using *perf* as a profiling tool for measuring proxy variables and a *RS PRO RS-9519BT* digital multimeter for the actual measurement of energy consumption.

➢ *Preliminary experiments*: a single-device setting is employed to provide an initial assessment of the effectiveness of the proposed methodology. Therefore, communication costs are excluded in this case, since no data exchange between devices occurs.

Experimental evaluation

- **Experimental design and data collection**
  - A set of synthetic datasets with different shapes (rows, columns) is generated with the *Orthogonal Latin Hypercube Sampling* (OLHS), a statistical sampling method designed to create evenly distributed samples across multiple dimensions.
  - The final training dataset $\mathcal{D}$ is obtained by monitoring the execution of the selected set of ML algorithms on such datasets, measuring the value of proxy variables.
  - The proxy variables selected to model energy consumption are as follows:
    - **Execution time** for computation, reflecting how long the application runs. The scaling factor is the device's average power consumption at full CPU load (β_{comp} = 3.2W).
    - **Cache misses** (CM) for data access, indicating instances where data must be fetched from slower memory. The scaling factor is the average energy per cache miss, emphasizing the impact of poor data locality (β_{data} = 1 nJ).

Classification and Regression Results

- We used two distinct set of constraints for the classification model, defined as follows:

$$\mathcal{Q}_1 = \{t_{\text{exec}} \leq 200\,\text{s},\ \text{mem} \leq 4\,\text{GB}\},$$
$$\mathcal{Q}_2 = \{t_{\text{exec}} \leq 400\,\text{s},\ \text{mem} \leq 4\,\text{GB}\}.$$

- Across all tested configurations and tasks, *AutoTabPFN* a pre-trained Transformer model designed for effective *in-context learning* (ICL) with small tabular datasets, consistently outperformed competing models.

COMPARISON OF CLASSIFICATION AND REGRESSION MODELS. RESULTS ARE AVERAGED OVER 10 RUNS, WITH BEST VALUES MARKED IN BOLD.

Model	Classification (F1 macro)		Regression (MAE)	
	$\mathcal{Q}_1$	$\mathcal{Q}_2$	Time (s)	CM (10^9)
AutoTabPFN	**0.93 ± 0.02**	**0.92 ± 0.03**	**39.11 ± 10.89**	**0.20 ± 0.07**
Random Forest	0.93 ± 0.03	0.90 ± 0.03	61.84 ± 16.51	0.39 ± 0.09
Gradient Boosting	0.93 ± 0.03	0.90 ± 0.03	64.93 ± 18.64	0.45 ± 0.12
KNeighbors	0.87 ± 0.03	0.80 ± 0.04	70.00 ± 18.18	0.47 ± 0.14
SVM	0.52 ± 0.08	0.51 ± 0.06	76.86 ± 11.84	0.57 ± 0.10

Example test cases

- We tested the system by submitting 3 different instances, obtaining the following results.

- **Test Case 1**
  - Algorithm: Logistic Regression, size: 501 MB, rows: 750 000, cols: 692
  - Classifier prediction → **Out-of-Memory Error**
  - Non-executability explanation (*SHAP abs.*):
    - **Dataset size** (0.35) ⬆
    - Number of rows (0.25) ⬆
    - Algorithm (0.21) ⬇
    - Number of columns (0.19) ⬇

- **Interpretation**: *The high dataset size led to excessive memory use, causing an OOM error*.

Example test cases

- **Test Case 2**
  - Algorithm: SVM, size: 244.62 MB, rows: 3 375 000, cols: 19
  - Classifier prediction → **Timeout Error**
  - Non-executability explanation (*SHAP abs.*):
    - **Number of rows** (0.52) ⬆
    - Algorithm (0.42) ⬆
    - Number of columns (0.06) ⬇
    - Dataset size (*near zero*) ⬆
- **Interpretation**: *The SVM model struggled with large row sizes, leading to an execution timeout*.

Example test cases

- **Test Case 3**
  - Algorithm: Random Forest, size: 18.45 MB, rows: 293 139, cols: 9
  - Classifier prediction → **Executable**
  - Regressors prediction:
    - **Execution time**: 398.6 s
      - **Algorithm** (0.50) ⬆
      - Number of rows (0.31) ⬇
      - Dataset size (0.12) ⬆
      - Number of columns (0.07) ⬇
    - **Cache miss**: $6.06 \cdot 10^9$
      - **Algorithm** (0.66) ⬆
      - Number of rows (0.33) ⬆
      - Number of columns (0.01) ⬆
      - Dataset size (*near zero*) ⬆
- **Interpretation**: *While the execution was successful, cache efficiency optimizations could reduce energy usage.*

Key findings and conclusions

- **Main contributions**
  - A novel methodology for **interpretable energy estimation** in Edge AI.
  - First approach combining **XAI** with **ICL-based energy estimation** in Edge AI settings.
  - Supports **green-aware AI** practices by providing fine-grained interpretable energy estimates to the user.

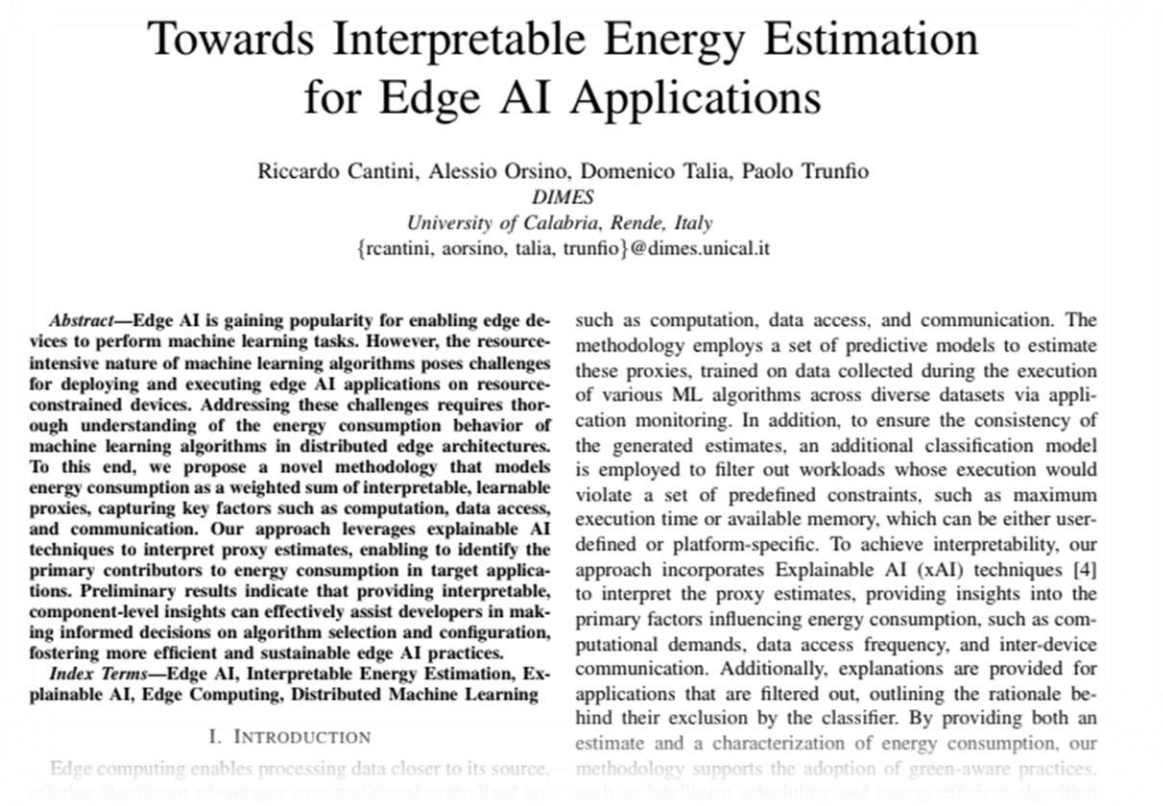
- **Impact & future applications**
  - Optimizing AI deployment in low-power edge environments (IoT, mobile AI, embedded systems).
  - Enabling real-time energy-aware decision-making in edge settings.
  - Supporting green-aware AI applications, such as green-aware NAS and ML workflow scheduling.

Towards Interpretable Energy Estimation for Edge AI Applications

- This work was accepted for publication at the **3rd International Workshop on Intelligent and Adaptive Edge-Cloud Operations and Services (Intel4EC**) workshop.
- The workshop will be held in conjunction with the **39th IEEE International Parallel and Distributed Processing Symposium (IPDPS'25)**.

INTEL4EC

IPDPS 2025 • Milano, Italy



Towards Interpretable Energy Estimation for Edge AI Applications

Riccardo Cantini, Alessio Orsino, Domenico Talia, Paolo Trunfio
DIMES
University of Calabria, Rende, Italy
{rcantini, aorsino, talia, trunfio}@dimes.unical.it

***Abstract*—Edge AI is gaining popularity for enabling edge devices to perform machine learning tasks. However, the resource-intensive nature of machine learning algorithms poses challenges for deploying and executing edge AI applications on resource-constrained devices. Addressing these challenges requires thorough understanding of the energy consumption behavior of machine learning algorithms in distributed edge architectures. To this end, we propose a novel methodology that models energy consumption as a weighted sum of interpretable, learnable proxies, capturing key factors such as computation, data access, and communication. Our approach leverages explainable AI techniques to interpret proxy estimates, enabling to identify the primary contributors to energy consumption in target applications. Preliminary results indicate that providing interpretable, component-level insights can effectively assist developers in making informed decisions on algorithm selection and configuration, fostering more efficient and sustainable edge AI practices.**

***Index Terms*—Edge AI, Interpretable Energy Estimation, Explainable AI, Edge Computing, Distributed Machine Learning**

I. Introduction

Edge computing enables processing data closer to its source, offering significant advantages over traditional centralized ap-

such as computation, data access, and communication. The methodology employs a set of predictive models to estimate these proxies, trained on data collected during the execution of various ML algorithms across diverse datasets via application monitoring. In addition, to ensure the consistency of the generated estimates, an additional classification model is employed to filter out workloads whose execution would violate a set of predefined constraints, such as maximum execution time or available memory, which can be either user-defined or platform-specific. To achieve interpretability, our approach incorporates Explainable AI (xAI) techniques [4] to interpret the proxy estimates, providing insights into the primary factors influencing energy consumption, such as computational demands, data access frequency, and inter-device communication. Additionally, explanations are provided for applications that are filtered out, outlining the rationale behind their exclusion by the classifier. By providing both an estimate and a characterization of energy consumption, our methodology supports the adoption of green-aware practices,