

**ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA**

**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING**

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Intelligent Systems

**HARDWARE DIMENSIONING FOR
ENVIRONMENTAL SUSTAINABILITY:
BENCHMARK OF AI ALGORITHMS AND
ENVIRONMENTAL IMPACT**

CANDIDATE

Enrico Morselli

SUPERVISOR

Prof. Andrea Borghesi

CO-SUPERVISOR

Allegra De Filippo, PhD.

Academic year 2024-2025

Session 5th

dedicated(X) :- friend(X).

Contents

1	Introduction	1
1.1	Background and Rationale	1
2	Related Works	2
2.1	Carbon Footprint in AI	2
3	Metodology	4
3.1	Empirical Model Learning in HADA	4
3.2	Integration of CodeCarbon in HADA	5
4	Experimental Analysis	6
4.1	Benchmarking on Different Hardware Platforms	6
5	HADA-as-a-Service	7
5.1	HADA Web Application	7
6	Conclusions	9
	Bibliography	10
	Acknowledgements	11

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Rationale

Growing adoption of AI technologies

Growing energy requirements of AI models

What we will do in this work

”This thesis is structured as follows...”

Chapter 2

Related Works

2.1 Carbon Footprint in AI

Several studies have addressed the issue of AI's carbon footprint. Tools like **Green Algorithms** and **CodeCarbon** have been developed to estimate and monitor emissions.

CodeCarbon is an open-source tool designed to track the energy consumption of computational resources and estimate the corresponding carbon emissions. The formula used is:

$$CO2eq = C \times E \quad (2.1)$$

where:

- **C** represents the carbon intensity of electricity (kg CO₂e per kWh), varying by country and energy mix.
- **E** is the total electricity consumed during computation (kWh).

By monitoring CPU, GPU, and RAM consumption, CodeCarbon estimates the total emissions associated with a computation. It retrieves the carbon intensity based on the geographical location and logs results at user-defined intervals (default: 15 seconds).

Installation:

```
pip install codecarbon
```


Chapter 3

Metodology

3.1 Empirical Model Learning in HADA

HADA employs the **Empirical Model Learning (EML)** paradigm, which integrates **Machine Learning (ML)** models into an optimization framework.

EML involves:

1. **Data Collection:** Running target algorithms under various hyperparameter configurations and hardware settings to collect performance data.
2. **Surrogate Model Creation:** Training ML models (e.g., Decision Trees) to approximate the relationship between input configurations and performance metrics (e.g., runtime, memory, carbon emissions).
3. **Optimization:** Using the learned models within a combinatorial optimization framework to find the best hardware configuration.

HADA was originally applied to the **ANTICIPATE** and **CONTINGENCY** stochastic algorithms used in energy management. These algorithms compute energy production schedules while minimizing cost and considering uncertainties.

3.2 Integration of CodeCarbon in HADA

To extend HADA for sustainable AI, we integrate CodeCarbon to track emissions in:

- ANTICIPATE and CONTINGENCY algorithms.
- MaxFlow Algorithms:
 - Boykov-Kolmogorov (BK)
 - Excess Incremental Breadth First Search (EIBFS)
 - Hochbaum’s Pseudo Flow (HPF)

Chapter 4

Experimental Analysis

4.1 Benchmarking on Different Hardware Platforms

Experiments were conducted on:

- MacBook Pro (2019)
- Leonardo Supercomputer (CINECA HPC)

Each algorithm was executed on 30 instances with hyperparameter values ranging from 1 to 100, generating datasets with 6,000 records per algorithm per hardware platform.

Chapter 5

HADA-as-a-Service

5.1 HADA Web Application

Benchmark data was integrated into the HADA web application, requiring:

- Creation of JSON configuration files for each algorithm-hardware combination.
- Specification of hyperparameters and performance targets.

Example JSON structure:

```
{
  "name": "anticipate",
  "HW_ID": "macbook",
  "hyperparams": [
    {"ID": "num_scenarios", "type": "int", "LB": 1, "UB": 100}
  ],
  "targets": [
    {"ID": "time", "LB": null, "UB": null},
    {"ID": "memory", "LB": null, "UB": null},
    {"ID": "emissions", "LB": null, "UB": null}
  ]
}
```

}

Chapter 6

Conclusions

This work extends HADA by integrating carbon emission constraints, enhancing its applicability for sustainable AI hardware selection. Through experimental benchmarks on laptops and HPC systems, we validated the framework's ability to balance performance and environmental impact. The web-based prototype enables users to make informed decisions when configuring AI workloads under sustainability constraints.

Acknowledgements

I'm very grateful to the inventor of the Prolog language, without whom this thesis couldn't exist. I'd also like to acknowledge my advisor Prof. Mario Rossi by tail-recursively acknowledging my advisor.