

University of Pisa

Department of Computer Science

**Parallel And Distributed Systems:
Paradigms And Models**

Course Project 2 - A.A. 2024/25

June 2025

**Distributed Parallel
Random Forest**

A Scalable Implementation for Classification Tasks

Implementation Variants:

- Single-node Shared-memory (FastFlow & OpenMP)
- Multi-node Hybrid (MPI + FastFlow)
- Performance Evaluation & Scalability Analysis

Author:

Oleksiy Nedobiychuk

Student ID: 597455

Email: o.nedobiychuk@studenti.unipi.it

Supervisor:

Massimo Torquati

Department of Computer Science

Email: massimo.torquati@unipi.it

Abstract

This report presents the design and implementation of a distributed parallel Random Forest algorithm for large-scale classification tasks. The project delivers three distinct implementations: single-node shared-memory versions using FastFlow and OpenMP frameworks, and a multi-node hybrid version combining MPI with either FastFlow or OpenMP. The implementations are validated against scikit-learn on datasets from the UCI Machine Learning Repository, including SUSY, Cover-type, and Iris datasets. Performance evaluation encompasses systematic analysis of scalability across varying numbers of decision trees, dataset characteristics, and computational resources. The study demonstrates efficient parallel algorithms achieving significant speedup while maintaining classification accuracy, with comprehensive analysis of strong and weak scalability up to 8 compute nodes. Key contributions include optimized load balancing strategies, minimized communication overhead, and detailed cost model development for distributed Random Forest implementations.

Contents

1	Introduction	2
2	Validation	2

1 Introduction

Authors:

- Alice Rossi -
- Bob Bianchi -
- Carla Verdi -
- David Neri -
- Eva Gallo -

2 Validation

I decided to use [this](#) Random Forest implementation because it is a pure implementation of Random Forests as described in: Leo Breiman. Random Forests. Machine Learning 45(1):5-32, 2001. The implementation is written in C++ and can be found inside the `include` folder of the project. I had to make some changes to the code because during the validation process, I found that it had some bugs.

List of Figures

List of Tables