Learning-based Approaches to Estimate Job Wait Time in HTC Datacenters

Luc Gombert and Frédéric Suter

IN2P3 Computing Center / CNRS
Villeurbanne, France

HEPiX Fall Workshop

October 13, 2020

Previously in HEPiX series . . .


How Fair is my Fair-Sharing?
Exposing Some Hidden Behavior Through Workload Analysis

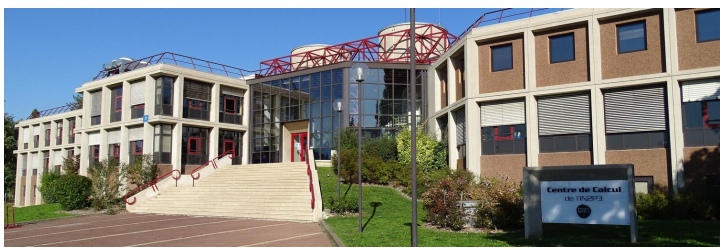
Frédéric Azevedo, Dalibor Klusáček, and Frédéric Suter

IN2P3 Computing Center / CNRS
Villeurbanne, France

HEPiX Spring Workshop
March 27, 2019

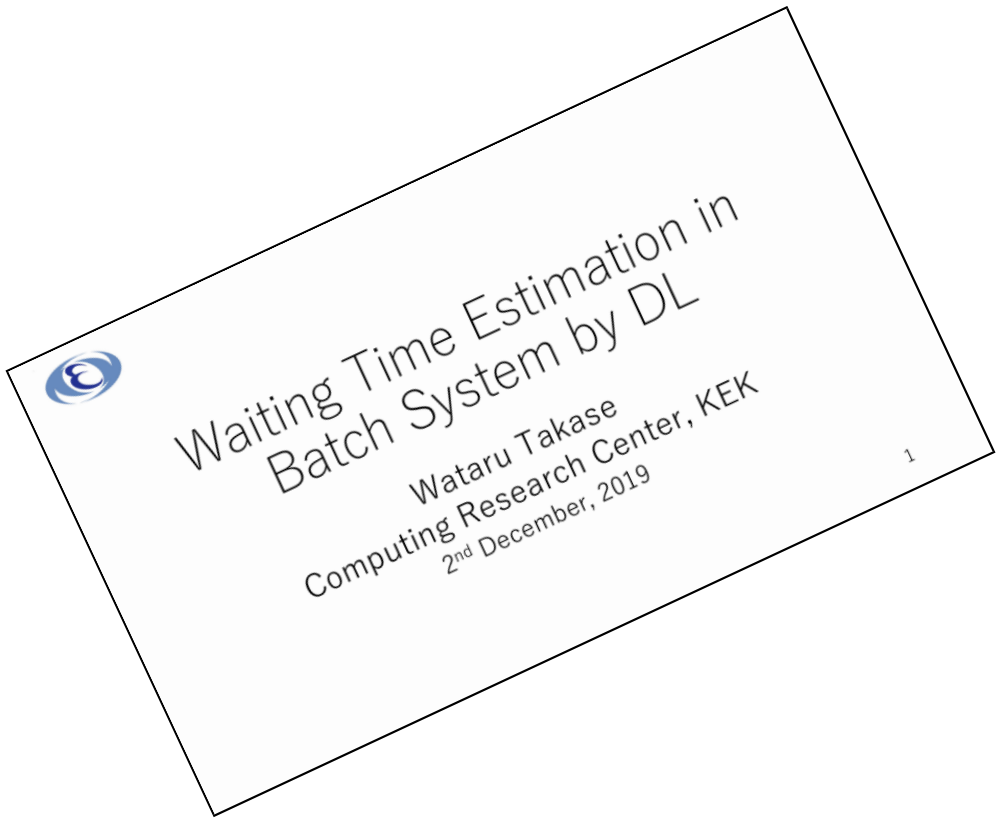
F. Suter – HEPiX Spring Workshop

- A first study of the workload processed at CC-IN2P3
- Focus on fairness for Local users
- Simulation of queue reconfiguration



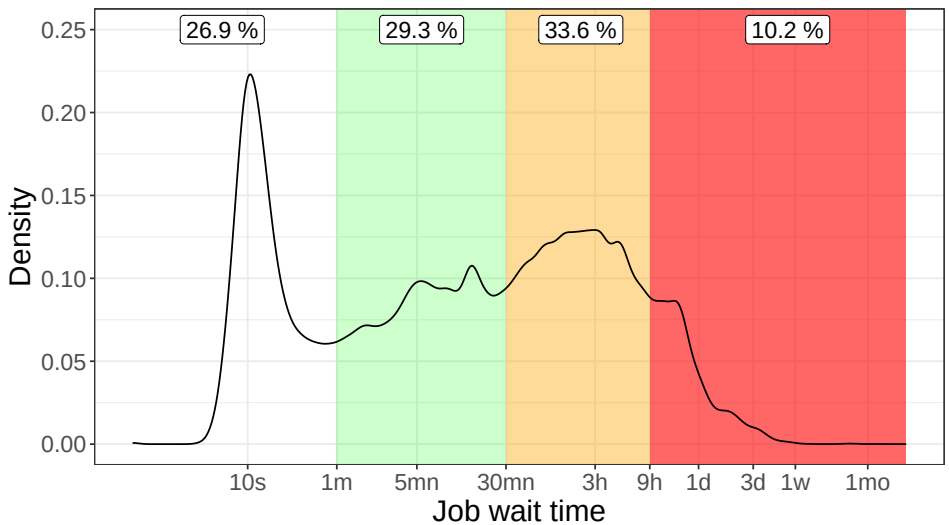
Acknowledgment

- Original motivation for this work came from a talk by **Wataru Takase** (KEK) at the FJPPL — Japan-France workshop on computing technologies



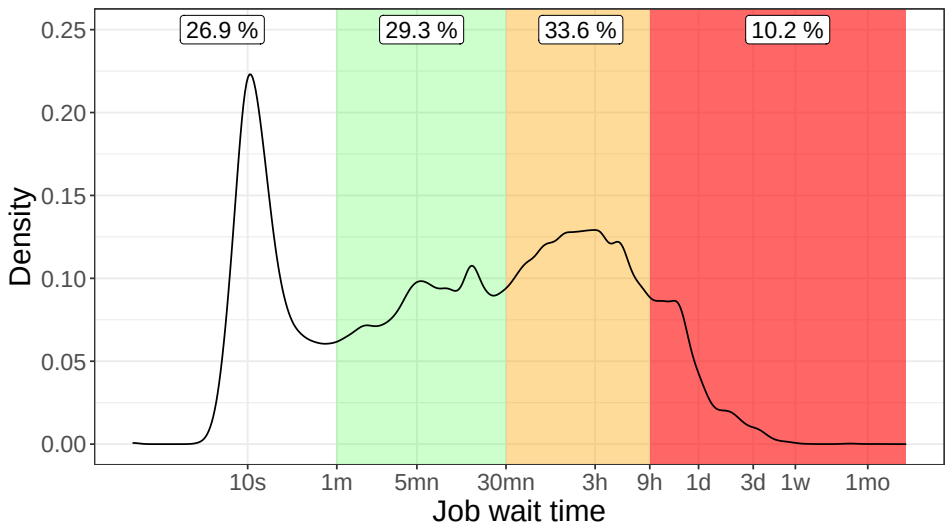
Motivations and Objectives

- Fair-share scheduling ⇒ no estimation of job start time returned to the user!
- Distribution of Local job wait time
  - Over 23 weeks from June 25, 2018 to December 2, 2018
  - 5,748,922 jobs on 35,000 cores



Motivations and Objectives

- Fair-share scheduling ⇒ no estimation of job start time returned to the user!
- Distribution of Local job wait time
  - Over 23 weeks from June 25, 2018 to December 2, 2018
  - 5,748,922 jobs on 35,000 cores



1. Can we explain why a job waits more than another?
2. Can we train some Machine Learning algorithms?
3. Can we get a good estimation of job wait time in the **orange** and **red** zones?

Outline

Who Submits the Job?

Job Features

- Owner: more than 2,500 individual accounts at CC-IN2P3
- Group: About 80 scientific collaborations

Resource Allocation Principle

1. Groups express pledges every year (as a computing power in HS06)
2. The sum of all pledges defines what CC-IN2P3 has to deliver
3. Each group gets a proportional share of this
   - Defines an consumption objective
   - Used by the job scheduler as a basis of its Fair-Share policy

Who Submits the Job?

Job Features

- Owner: more than 2,500 individual accounts at CC-IN2P3
- Group: About 80 scientific collaborations

Resource Allocation Principle

1. Groups express pledges every year (as a computing power in HS06)
2. The sum of all pledges defines what CC-IN2P3 has to deliver
3. Each group gets a proportional share of this
   - Defines an consumption objective
   - Used by the job scheduler as a basis of its Fair-Share policy

Intuitive Causes

1. Small groups get less resources $\rightsquigarrow$ wait more!
2. Overconsumption of share $\rightsquigarrow$ lower priority $\rightsquigarrow$ wait more!
3. Job owners can be manually blocked by operators $\rightsquigarrow$ wait more!

What is the Job Requesting?

Job Features

- Time: either Walltime or CPU time
  - hard or soft limits – default values if none provided
- Memory: either resident or virtual
  - hard or soft limits – default values if none provided
- Slots: almost always one for Local jobs
- Access to special resources: submitted to quotas

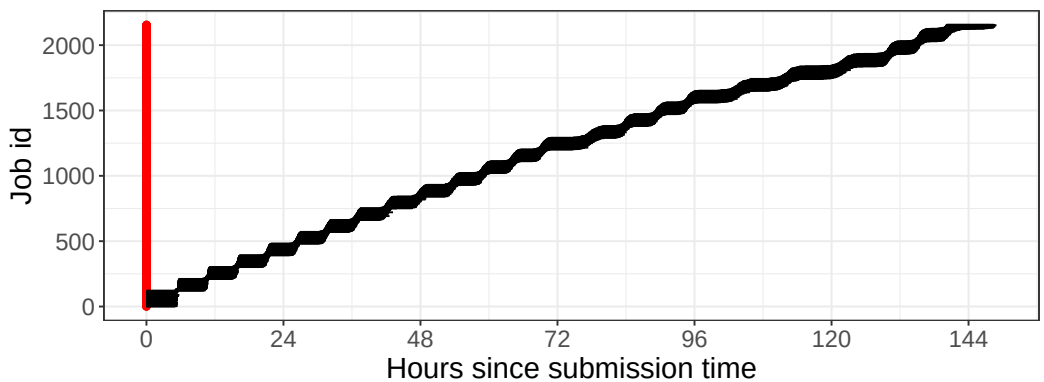
What is the Job Requesting?

Job Features

- Time: either Walltime or CPU time
  - hard or soft limits – default values if none provided
- Memory: either resident or virtual
  - hard or soft limits – default values if none provided
- Slots: almost always one for Local jobs
- Access to special resources: submitted to quotas

Intuituive Causes

1. HTC is not HPC! $\rightsquigarrow$ low impact of time, memory, and slot requests
2. Lots of (stringent) quotas $\rightsquigarrow$ wait more if reached!

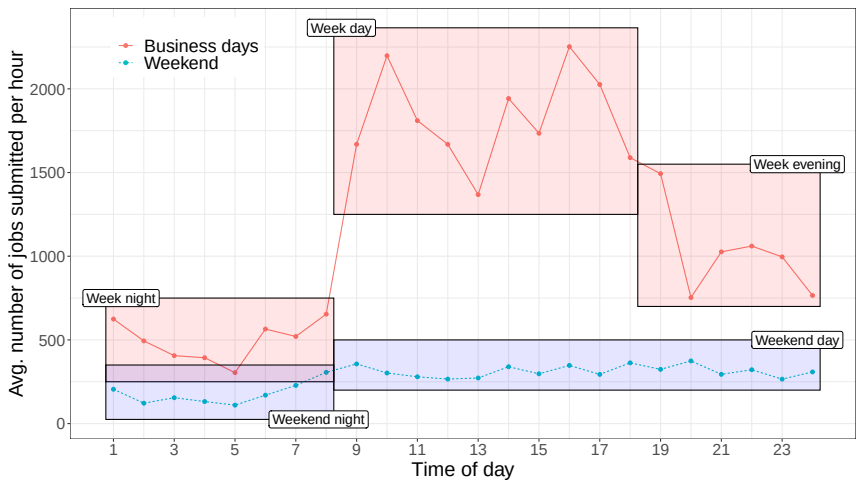


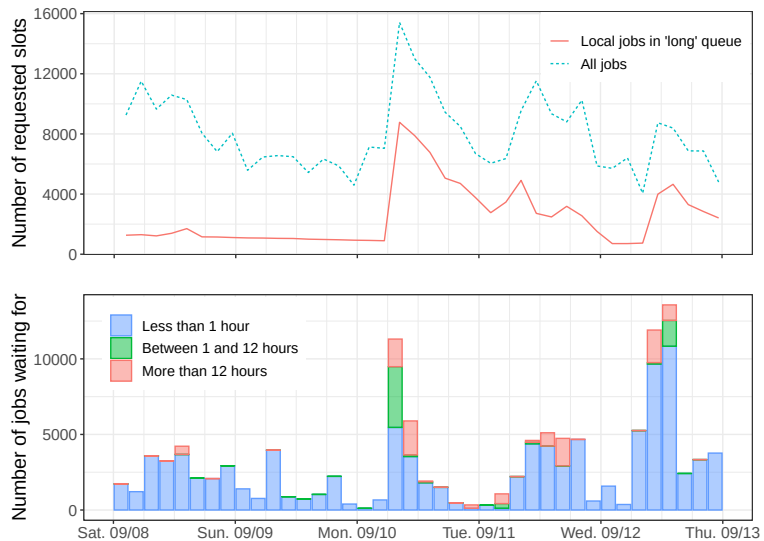
When and Where is the Job Submitted?

Job and System features

- Submission time
- Current queue status: number of pending jobs
- Current platform status: number of running jobs

Intuitive Causes





Outline

Objectives and Performance Metrics

Objectives

- Regression problem: Estimate the time a job will wait when submitted
  - Users may not really need that level of precision
- Classification problem: Determine in which time range a job will fall

Class	Wait Time Range
1	Less than 30 minutes
2	30 minutes to 2 hours
3	2 hours to 4 hours
4	4 hours to 6 hours
5	6 hours to 9 hours
6	9 hours to 12 hours
7	12 hours to 24 hours
8	more than 24 hours

Performance metrics

- Learning and Prediction times: Has to be usable in production!
- Wait time estimation: Error distribution
- Wait time range classification: Confusion matrix

ML Algorithm Selection

Common Properties

- Rely on ScikitLearn implementations
- Favor fast algorithms

Regression

- Linear Regression
- Decision Tree Regressor
- Ensemble Methods
  - AdaBoost and Bagging
  - Depth-9 DT as weak learner
  - 50 subsets

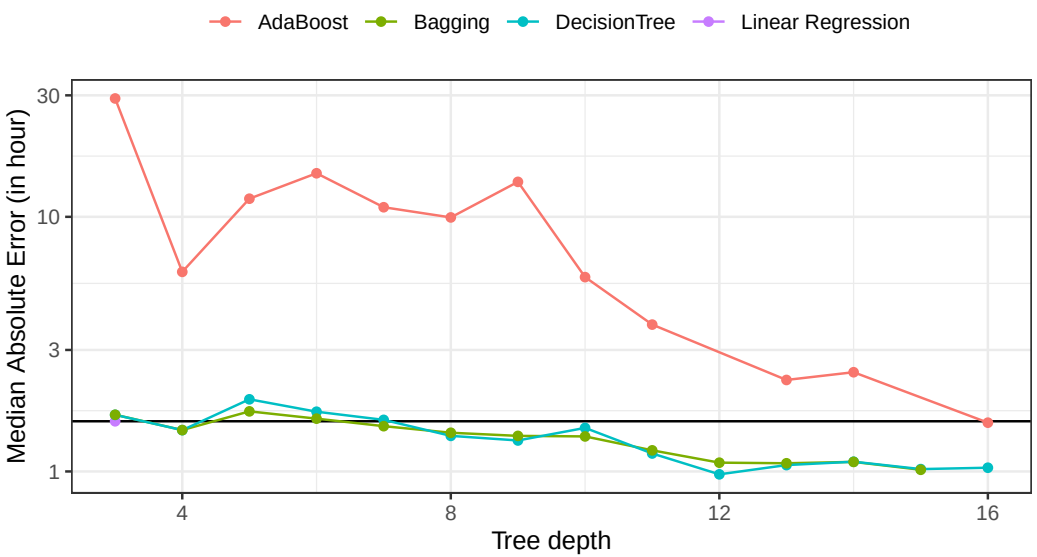
Classification

- Naive Bayes
- Decision Tree Classifier
- Ensemble Methods
  - AdaBoost and Bagging
  - Depth-1 DT as weak classifier
  - 50 subsets

Additionnal Approach

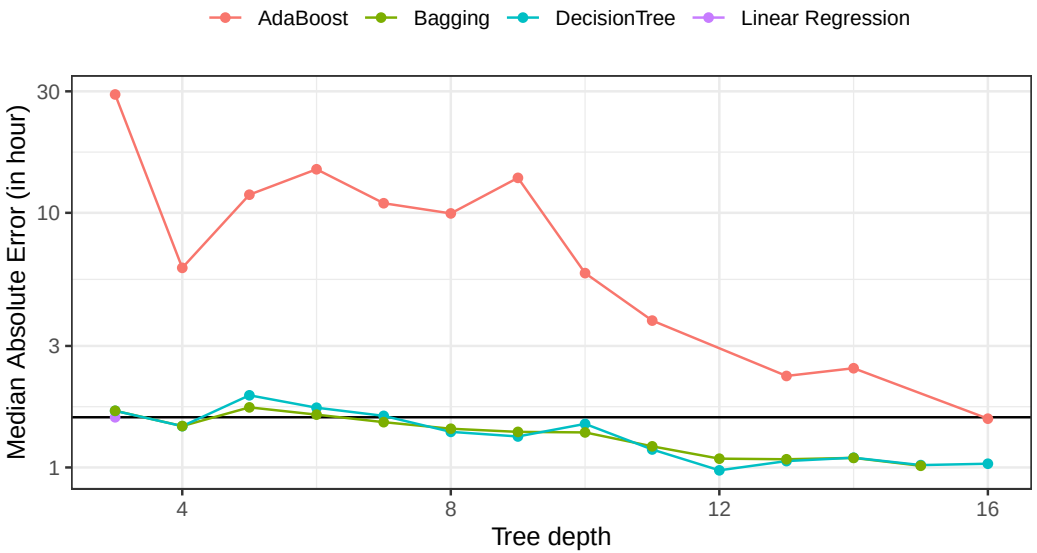
- Two-step Classification: solve regression and then classify

Accuracy of the Job Wait Time Estimation



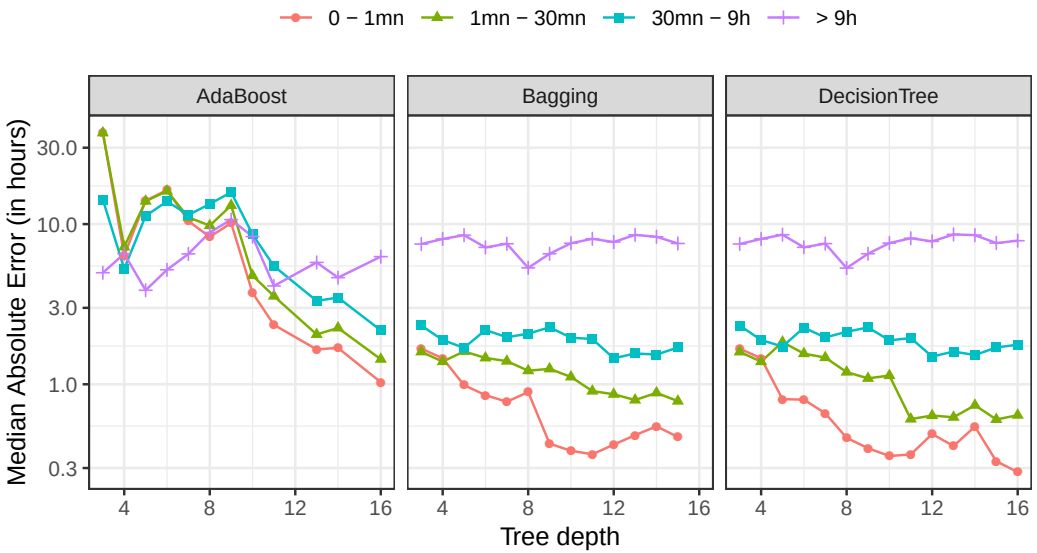
- AdaBoost is bad
- Bagging $\approx$ DT
- Less than 1h error for 50% of the jobs
- Satisfying!

Accuracy of the Job Wait Time Estimation



- AdaBoost is bad
- Bagging ≈ DT
- Less than 1h error for 50% of the jobs
- Satisfying!

- Split by "zone"
- Better for early starters
- Degradation for others
- Not satisfying :-/

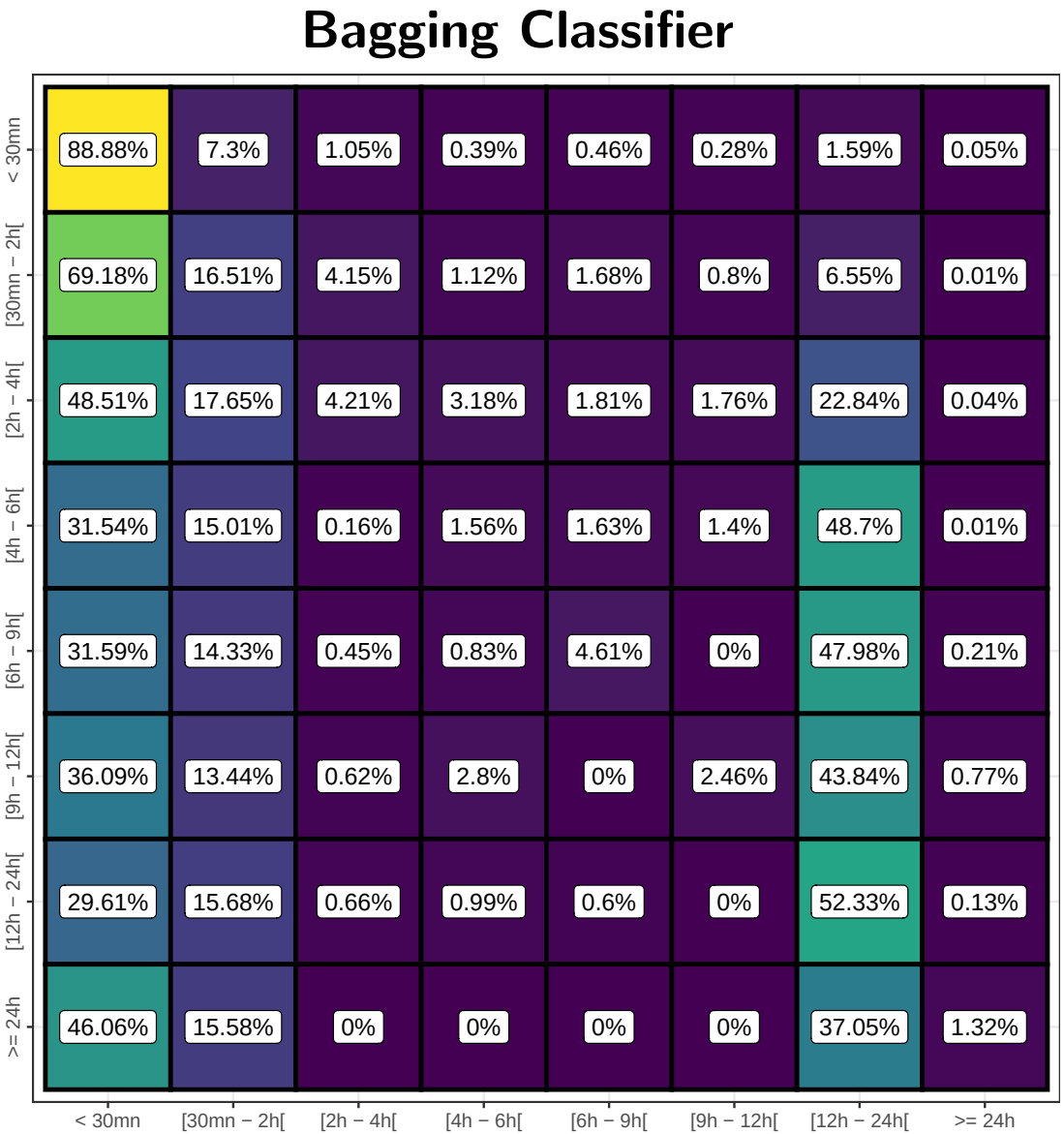


Accuracy of the Time Range Classification

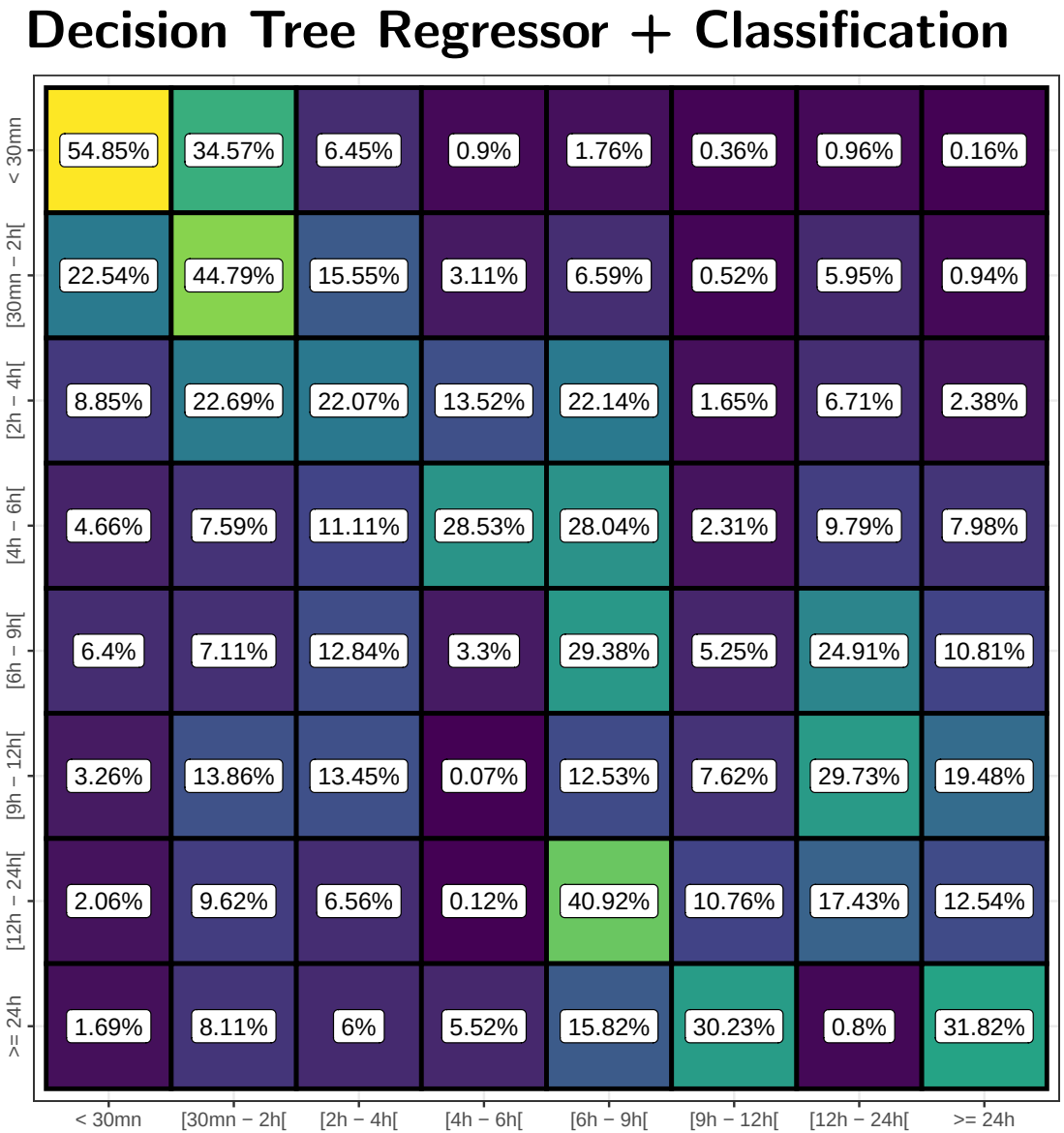
$\sim$ 43 % of jobs in the right class

$\sim$ 73% of jobs in right or adjacent class

Accuracy of the Time Range Classification



Accuracy of the Time Range Classification



Conclusion and Future Work

Conclusion

- Analyzed 23 weeks of job submissions to a HTC center
- Identified some intuitive causes of job wait time
- Learn on 15 job and system features to predict job wait time
- Early results for Regression and Classification problems
  - Assessing the performance of multiple ML algorithms
  - Some biases have to be solved

Future Work

- Improve our predictions
  - Take early starter jobs into account
- Investigate the use of Deep Learning algorithms
- Automate and transfer procedure to User Support team at CC-IN2P3
- Integrate this work to the newly deployed CC-IN2P3 user portal

Learning-based Approaches to Estimate Job Wait Time in HTC Datacenters

QUESTIONS?

Luc Gombert and Frédéric Suter

IN2P3 Computing Center / CNRS