Predicting the Execution Time of Workflow Activities Based on Their Input Features

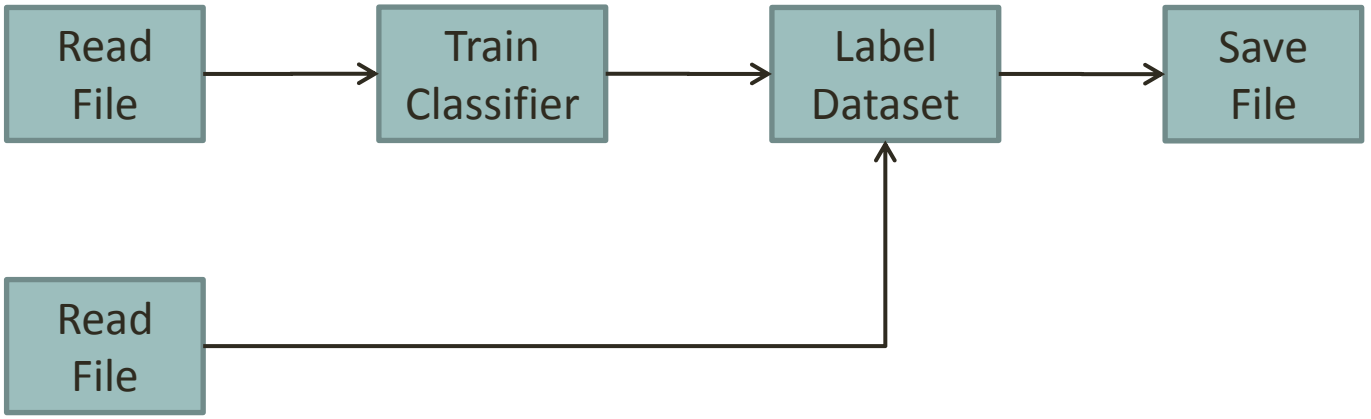
Tudor Miu and Paolo Missier
Newcastle University, UK

In this presentation

- Workflows
  - Execution time prediction
  - Rationale
  - Existing approaches
- Input-based execution time prediction
  - Rationale
  - Method
  - Experiment
  - Results
- Future work

Example of a workflow

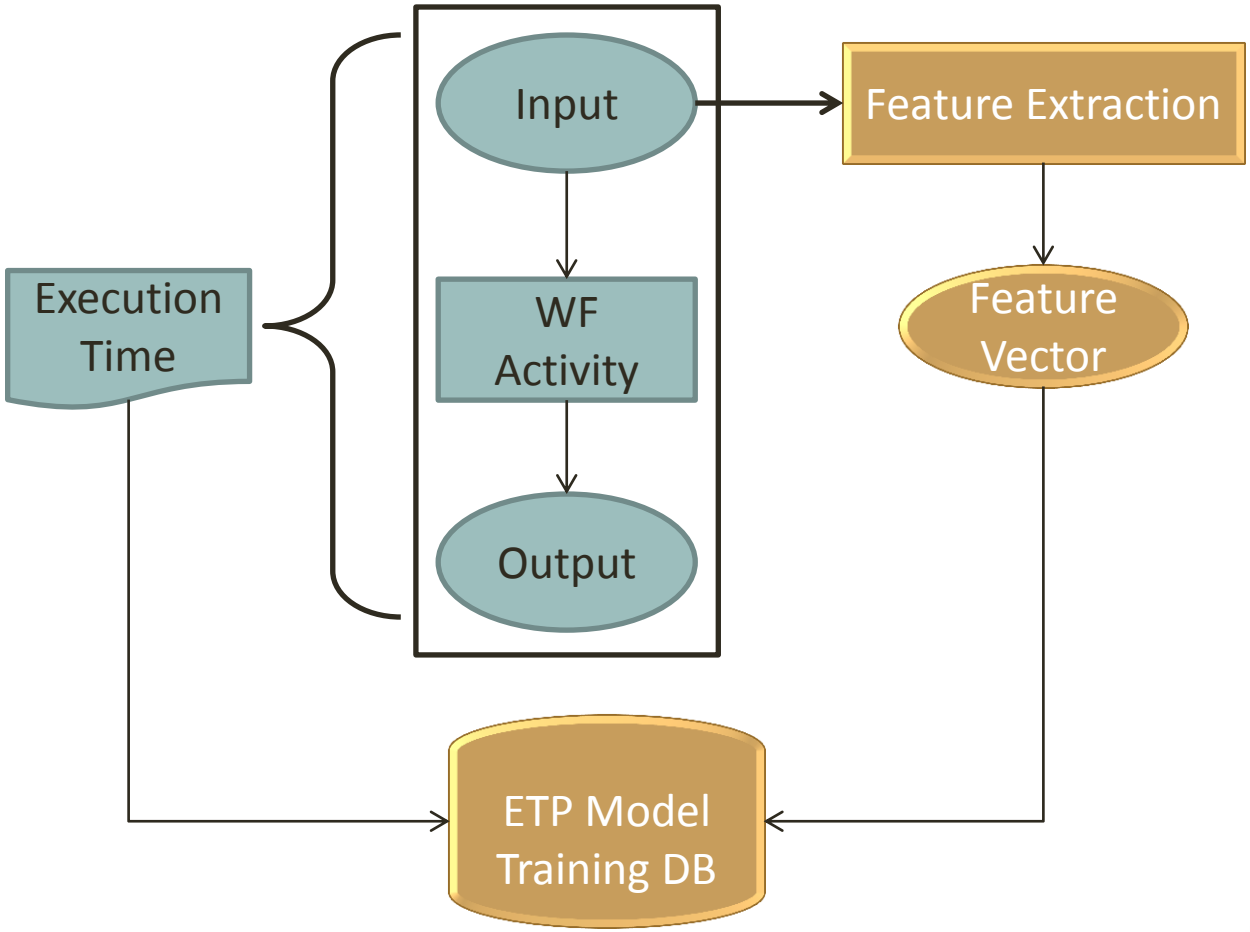
- Machine Learning – generic classification workflow



Execution time prediction (ETP)

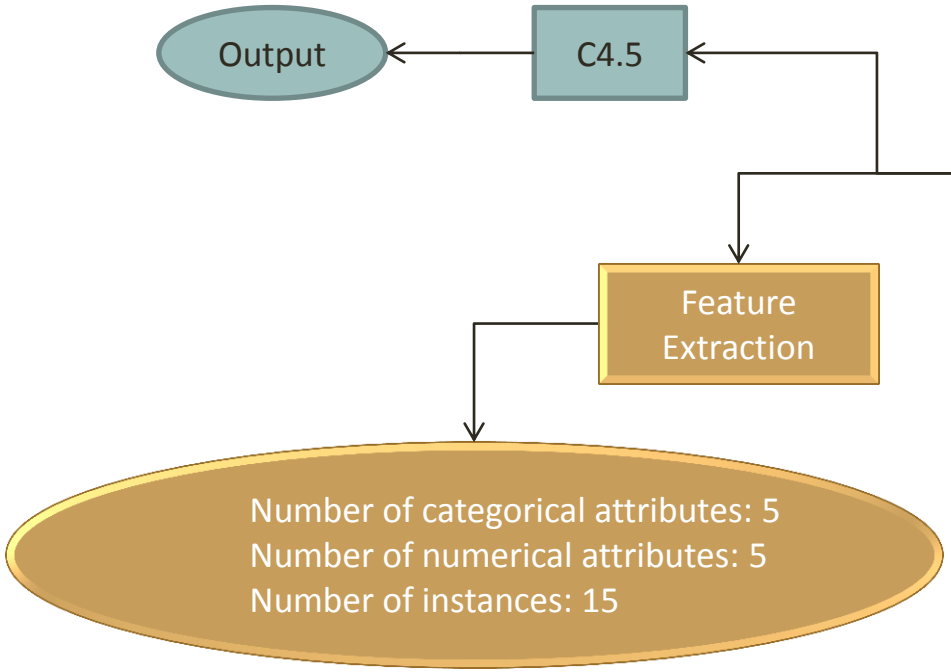
- Why?
  - Scheduling in large computational infrastructures
  - Cost estimation on ‘rented’ platforms, e.g. cloud
- How it is usually done
  - Time-series analysis
  - Incidental features – group jobs with similar attributes
- We propose to predict execution times for jobs whose execution time is determined:
  - by the input
  - in less incidental ways

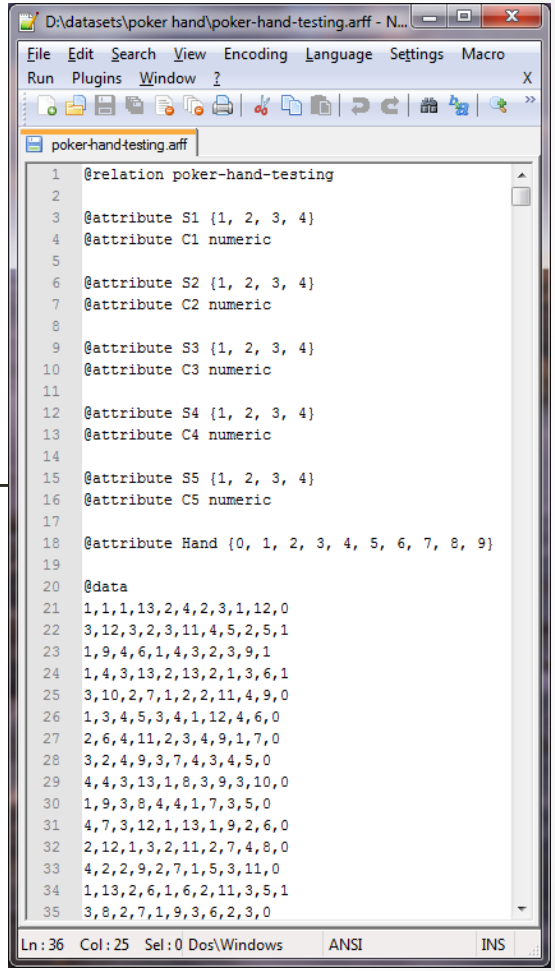
Input-based execution time prediction (ETP) – creating an execution history



Feature extraction example: C4.5

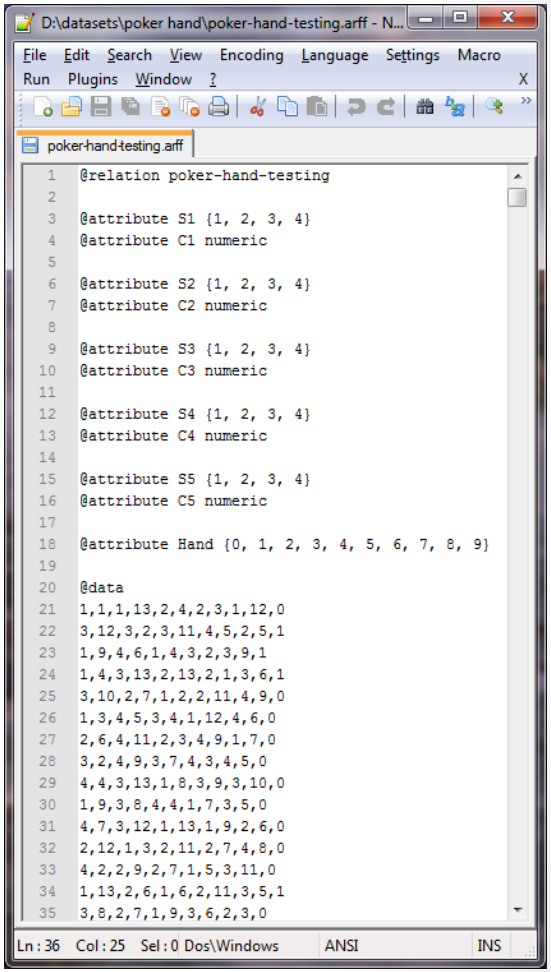
- Measurable quantity
- Summary





Features for C4.5: generic input features

- Either *count* variables
  - Number of categorical attributes
  - Number of numerical attributes
- Or *indicator* variables
  - Attribute included in the input dataset?
  - Input schema description
- Plus *number of instances*



Features for C4.5: static analysis

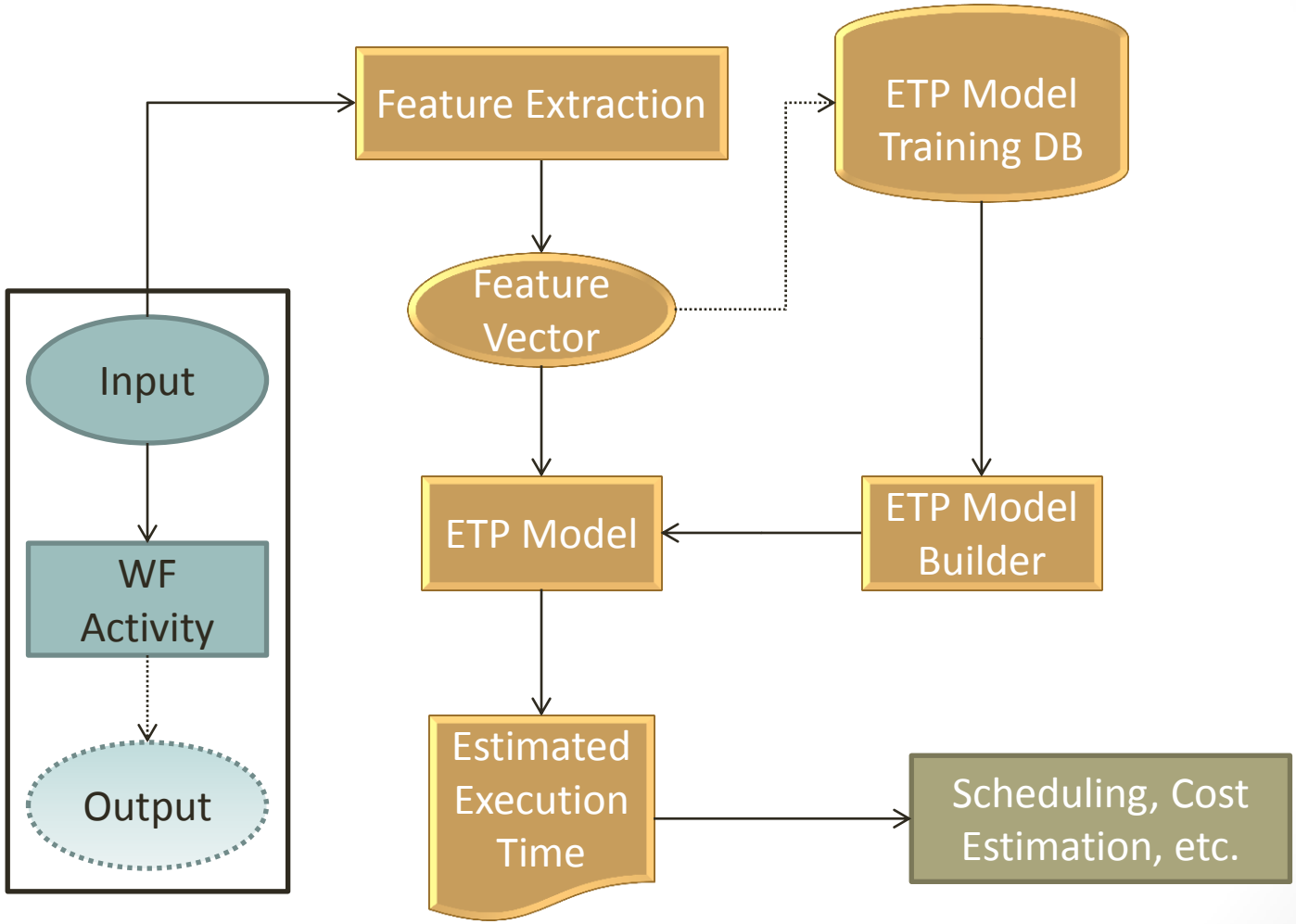
- Analysis of the mechanics of the target algorithm
  - Features
    - N – number of instances
    - p – number of attributes
    - $O(pNlogN)..O(pN^2)$
  - Predictions by linear regression

$$\hat{t} = \beta_1 p_{num} NlogN + \beta_2 p_{cat} NlogN + \beta_3 p_{num} N^2 + \beta_4 p_{cat} N^2$$

(plus lower order terms)

- Not always possible/feasible
- Not always best even if available

Input-based ETP – delivering predictions



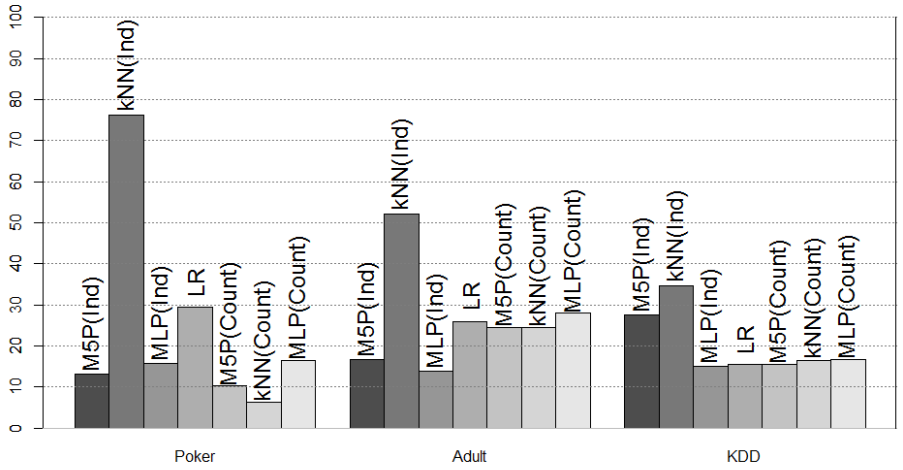
Input-based ETP experiment

- Target algorithm: training phase of C4.5
  - Fixed parameterization
  - Invariable platform (idle computer)
- Three common ML datasets from the UCI ML repository *
- Three common regression models (Weka) used for ETP
  - K Nearest Neighbours (kNN)
  - Multilayer Perceptron (MLP)
  - M5P decision tree (M5P)
- Simulated input variability
  - Random vertical and horizontal sampling
- Measured prediction error
  - For each model/dataset combination
  - As a function of size of execution history

* http://archive.ics.uci.edu/ml/

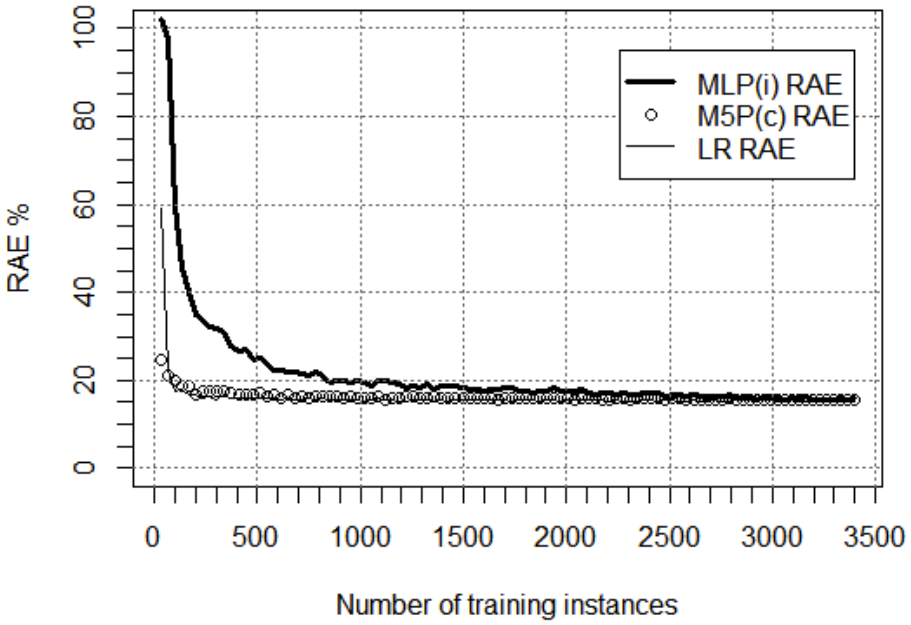
Results

- Relative Absolute Error: $RAE = \frac{\sum_{i=1}^{N}|a_i - p_i|}{\sum_{i=1}^{N}|a_i - \bar{a}|}$
- Based on 3400 samples of training sets for C4.5 for each major dataset



* Obtained from 10-fold cross validation

Results



* 10-fold cross validation on 10 samples for each size of the execution time history

- Train model on one major dataset and make predictions on another?
  - Sometimes RAE > 100%

Future work

- Input features complementary to activity parameters
- Input-based ETP may explain some variability
  - Input-based ETP alone is not a panacea
  - Combining input-based ETP with other established methods
- Latent features
  - Detecting change
  - What else?