Using Machine Learning in Trace-driven Energy-Aware Simulations of High-Throughput Computing Systems

A. Stephen McGough[1→2], Noura Al Moubayed[1→2], Matthew Forshaw[2]

1 Durham University, UK

2 Newcastle University, UK

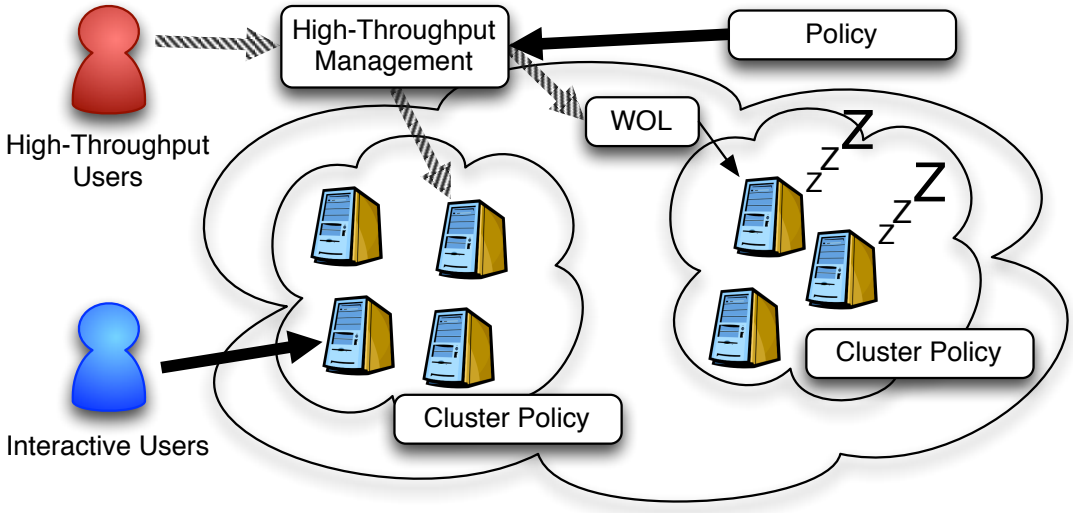
23/4/2017 | ENERGY-SIM 2017

Durham University

Newcastle University

Opportunistic High-throughput cluster

- Using collections of distributed workstations and/or dedicated clusters as a distributed high-throughput computing (HTC) facility
  - manages both resources (machines) and requests (jobs)
  - Often used to exploit existing computing facilities
  - Resilient architecture
    - If a job fails to complete on one resource it will be reallocated to a different resource



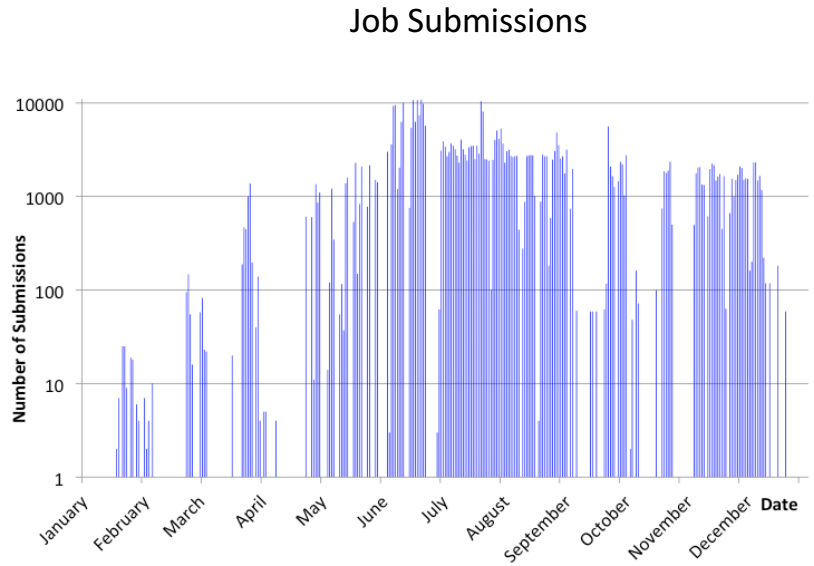

Dedicated


Cycle stealing

HTC

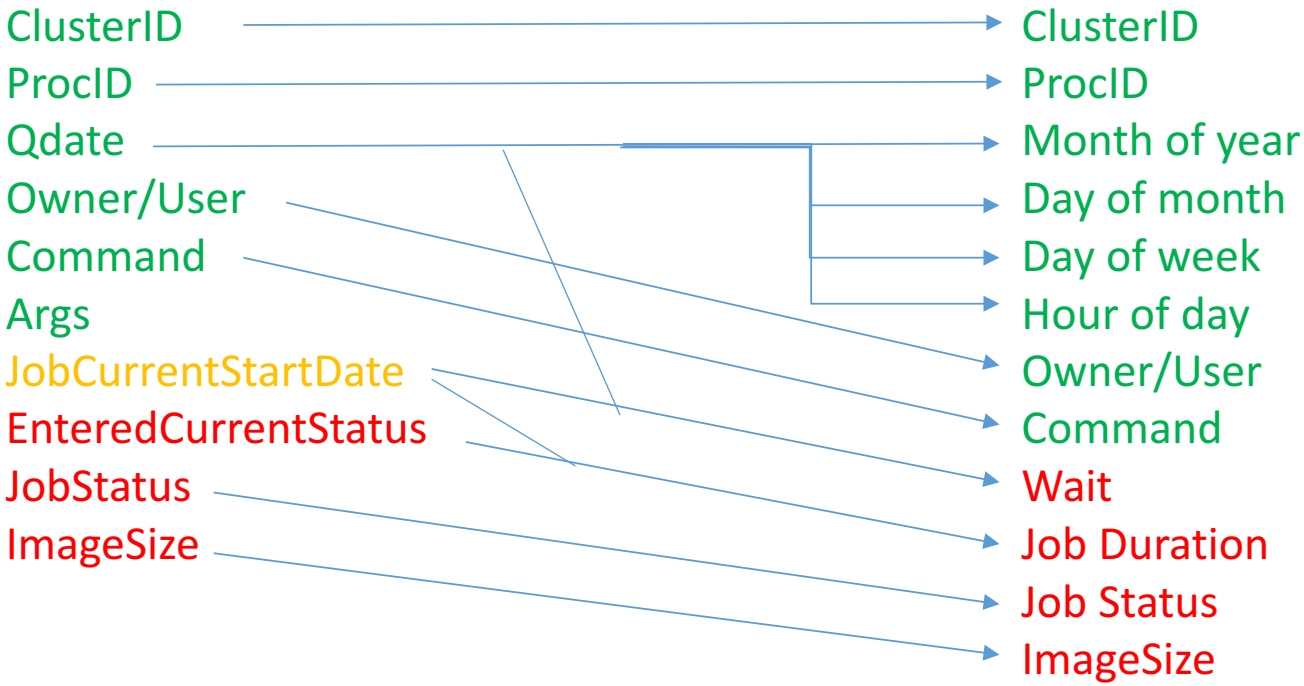
Motivation

- The trace logs contain lots of data → knowledge
  - Can we extract this to do better prediction of system
  - Without looking at data in the future
- Which are ’good’ jobs and which are ‘bad’
- How long will the job take to run?
- How much memory will it need?
- Which jobs are not running as expected?



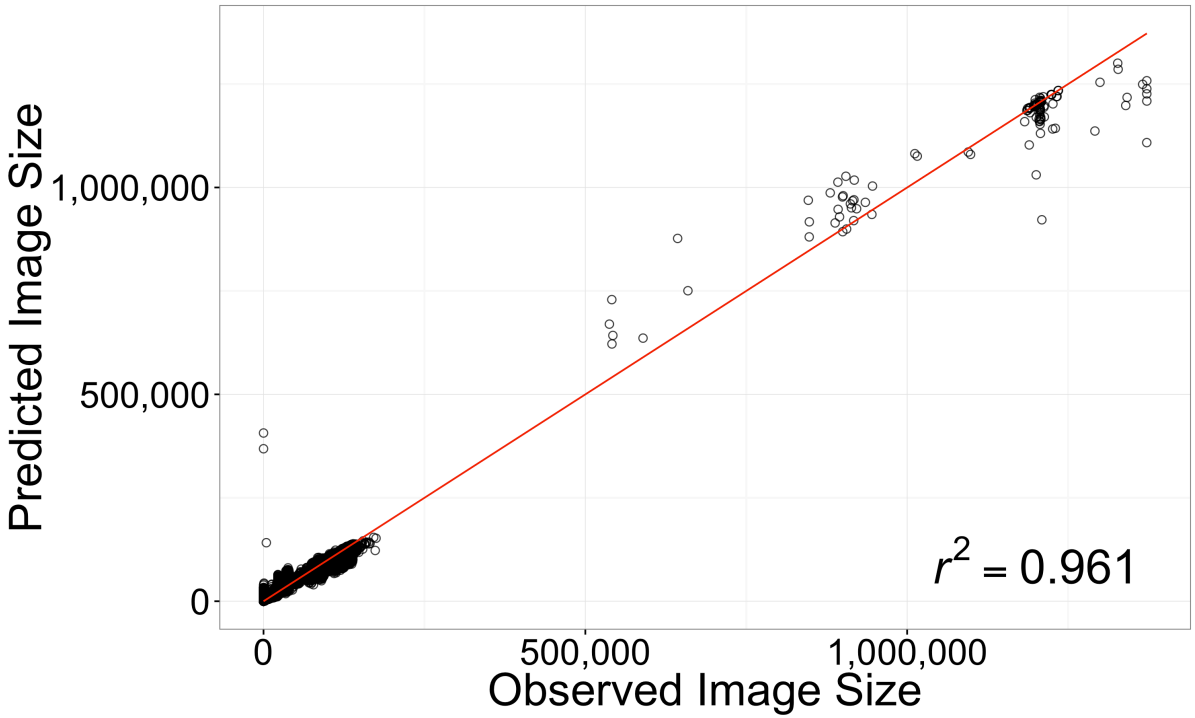
Trace Data

- Trace log contains:
  - Data we know at time of submission
  - Data we know while the job is running
  - Data we only know when job has finished
- Seek to predict job characteristics before known



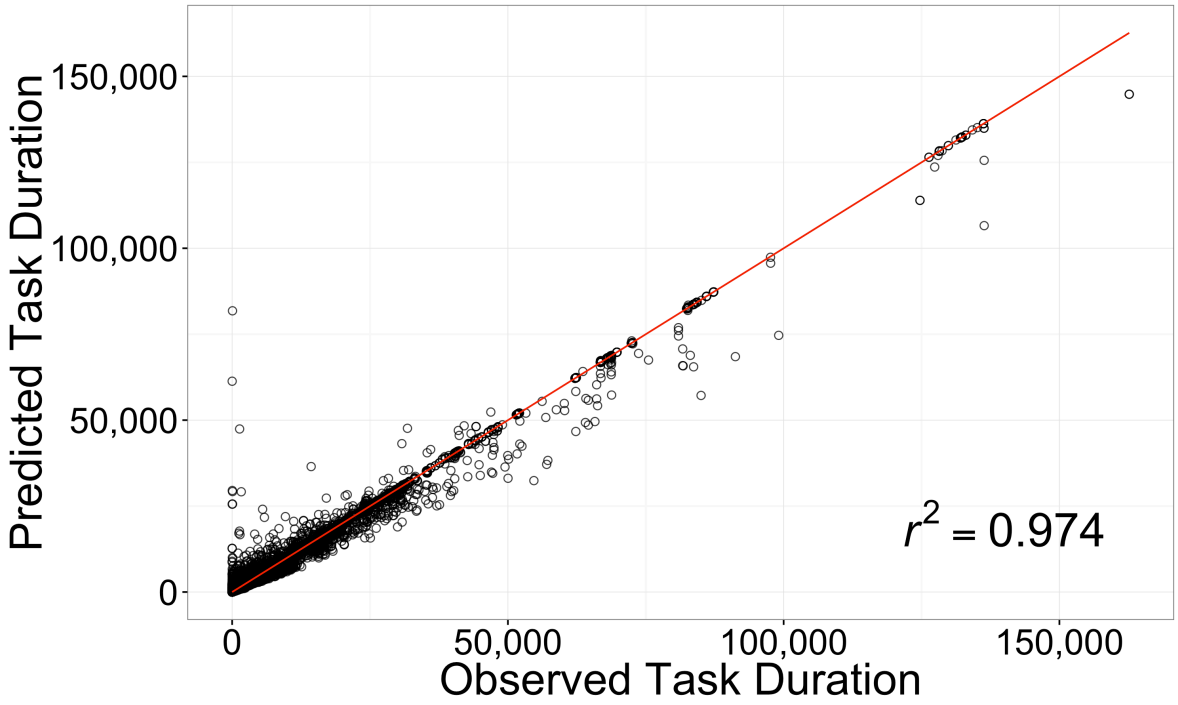
Prediction of Image size

- Using a Random Forrest Regression approach
- Used to better select the resource to deploy to
- High confidence in prediction



Prediction of task Duration

- Using a Random Forrest Regression approach
- Used to better select the resource to deploy to
- High confidence in prediction
- Use to help understand if job is running as expected

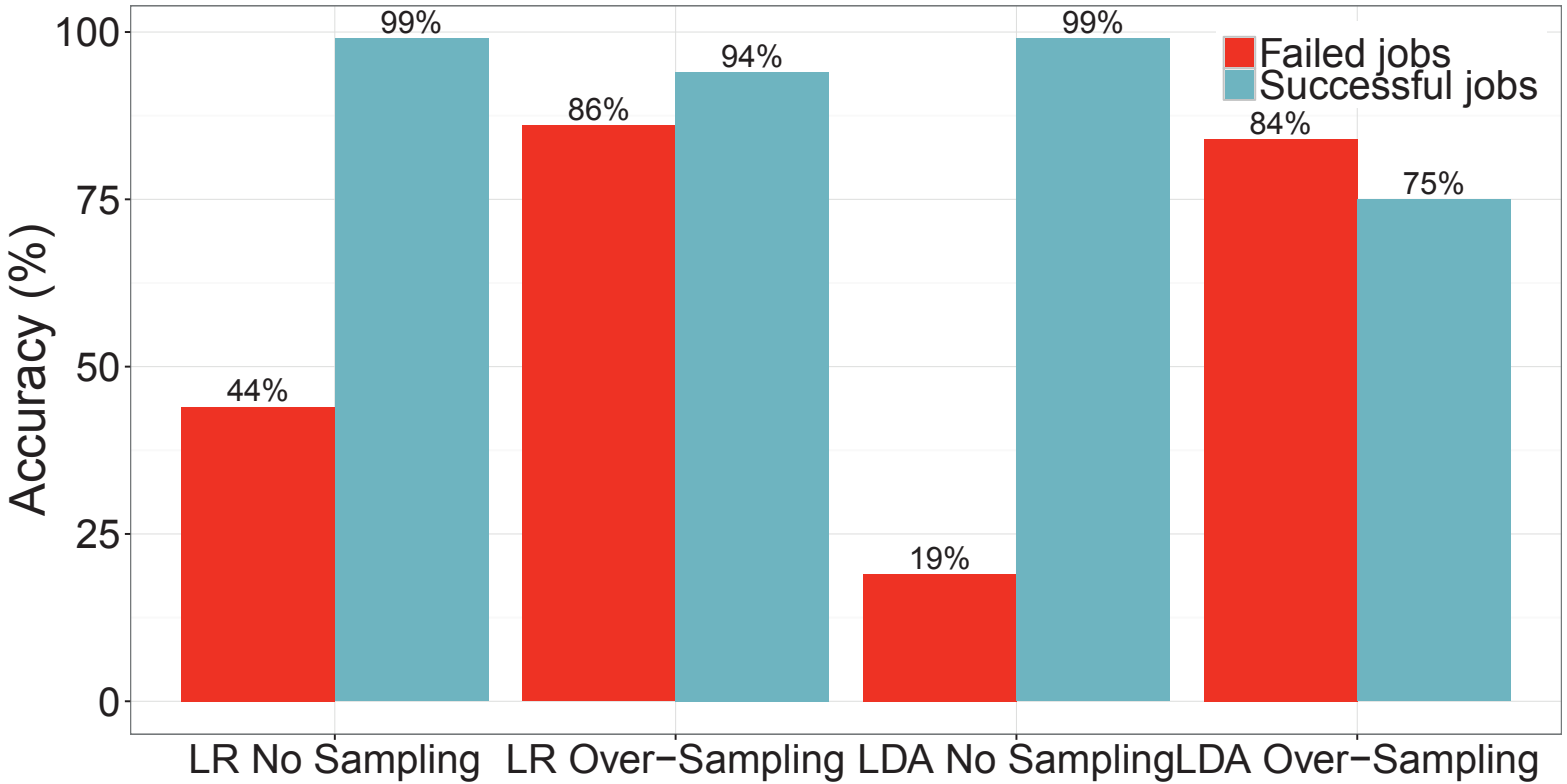


Prediction – is job ‘good’

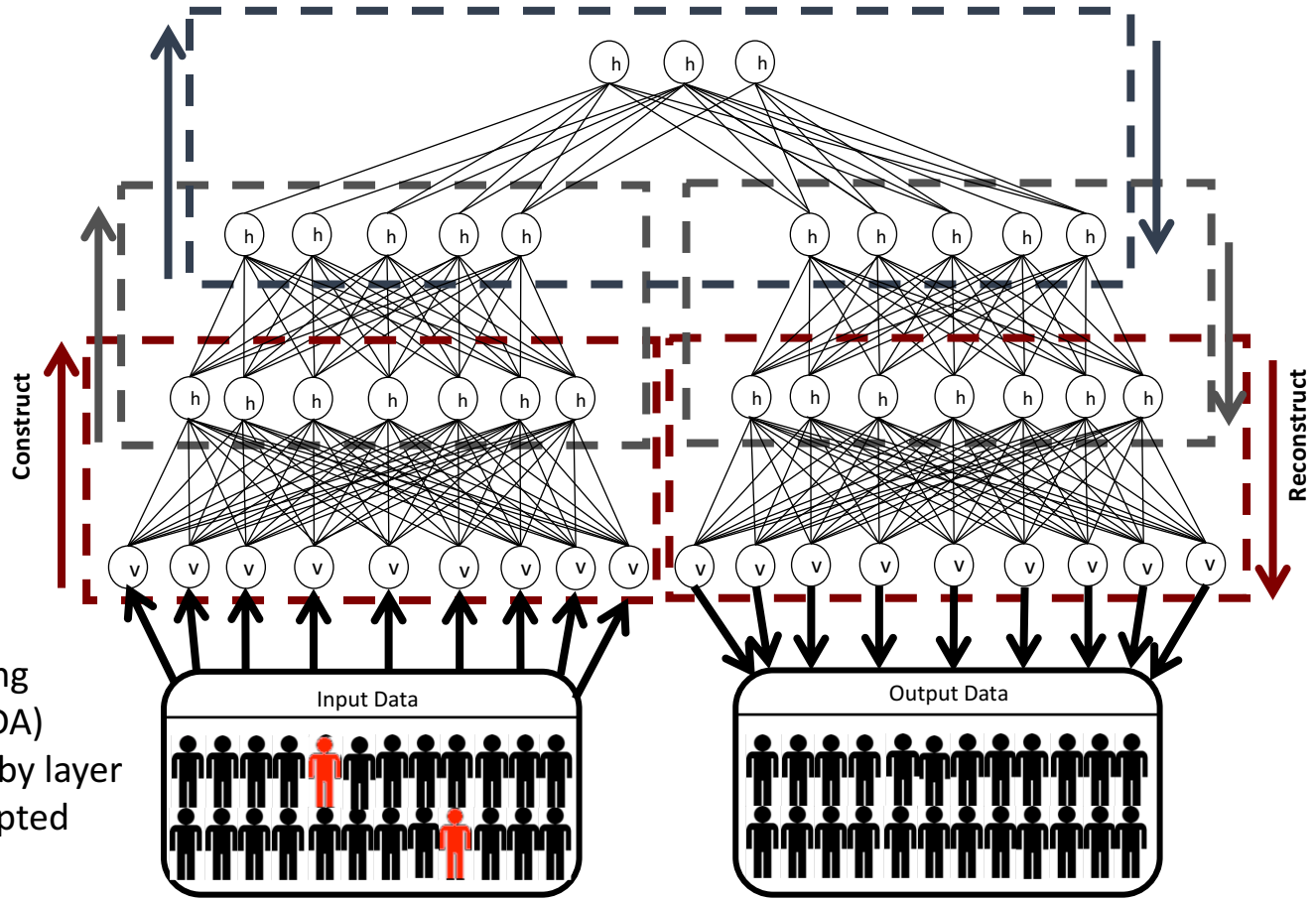
- Problem here is ratio of good jobs to bad
  - 429,593 good compared to 4189 bad
- Most AI algorithms would just say all jobs good
- Need to oversample using SMOTE
- Then compared two classifiers:
  - Logistic Regression (LR)
  - Linear Discriminant Analysis (LDA)

Prediction – is job ‘good’

- Then compared two classifiers:
  - Logistic Regression (LR)
  - Linear Discriminant Analysis (LDA)



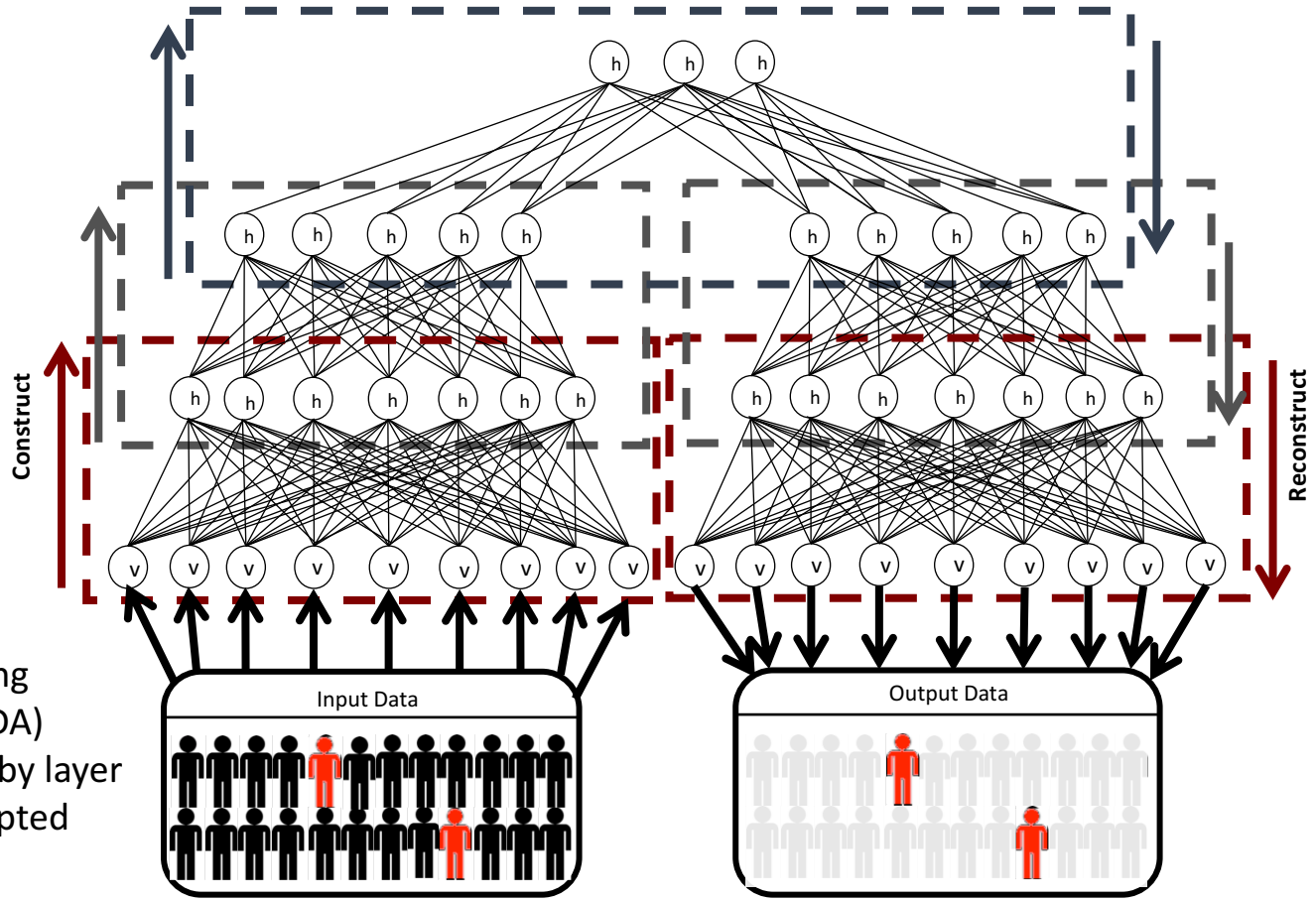
Anomaly Detection: Unsupervised Deep Learning



Stacked Denoising Autoencoder (SDA)

- Trained layer by layer
- Input is corrupted with noise

Anomaly Detection: Unsupervised Deep Learning

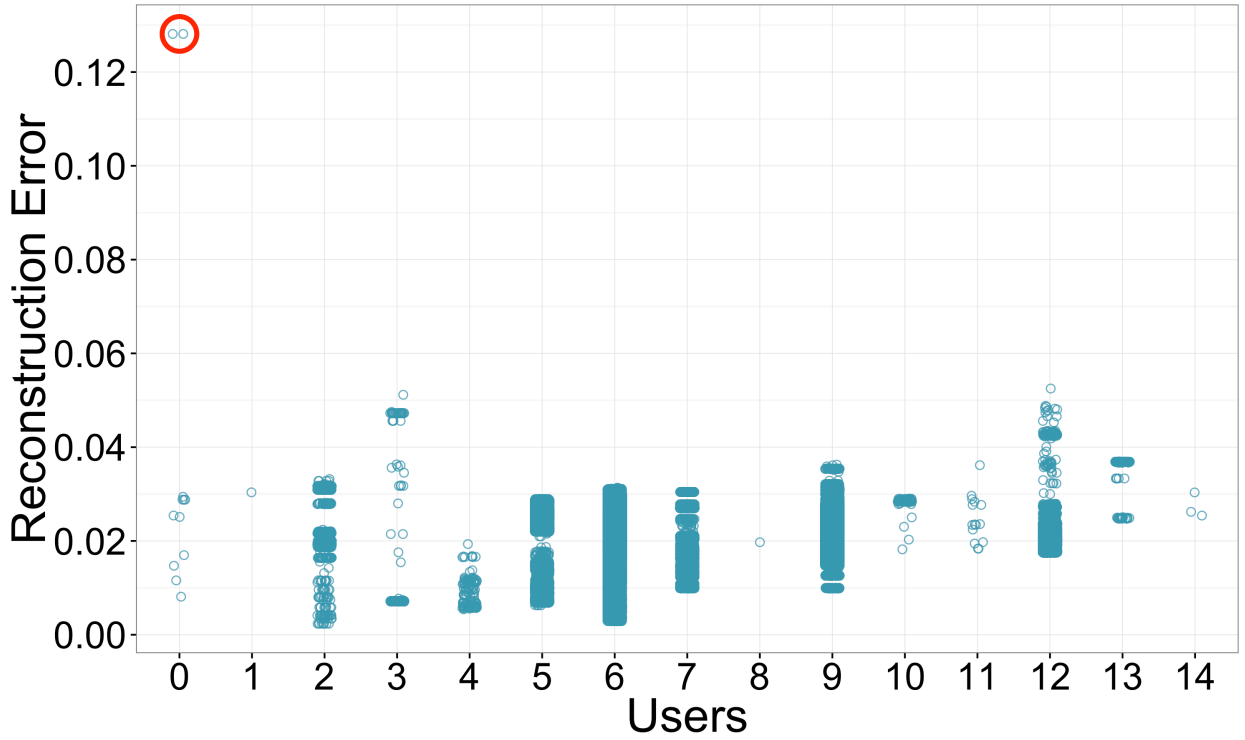


Stacked Denoising Autoencoder (SDA)

- Trained layer by layer
- Input is corrupted with noise

Finding the ’odd’ jobs

- Using Deep learning (Auto Encoder)
- Train system and use reconstruction error
- Where reconstruction error is high – this is an odd job



Conclusions and future Directions

- Can identify ‘good’ and ’bad’ jobs – good accuracy
- Can determine execution time – good accuracy
- Can determine image size – good accuracy
- Future:
  - Use this to build better scheduling algorithms for simulations and eventually deploy to real system
  - Can we create better synthetic logs
    - Arbitrary length
    - Arbitrary density

stephen.mcgough@durham.ac.uk stephen.mcgough@newcastle.ac.uk
noura.al-moubayed@durham.ac.uk
matthew.forshaw@newcastle.ac.uk