Anomaly detection: ATLAS and RAMP report

James Catmore (University of Oslo)
David Rousseau (LAL, Paris)

Including contributions from
Imad Chaabane (LRI/UPSud), Victor Estrade (LRI/UPSud), Julio De Castro Vargas Fernandes (Rio de Janeiro), Sergei Gleyzer (UFlorida), Cécile Germain (LRI/UPSud), Isabelle Guyon (LRI/UPSud), Balazs Kegl (LAL/CNRS), Alexei Klimentov (BNL), Edouard Leicht (LAL/CNRS), Gilles Louppe (NYU), David Rousseau (LAL/CNRS), Jose Manoel Seixas (Rio de Janeiro), Jean-Roch Vlimant (CalTech)

What is anomaly detection?

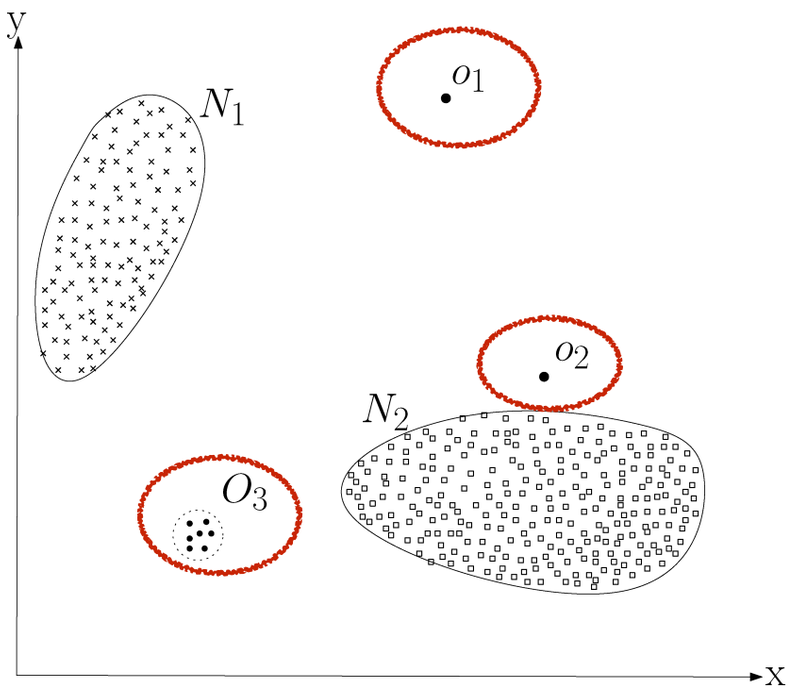
I think

is

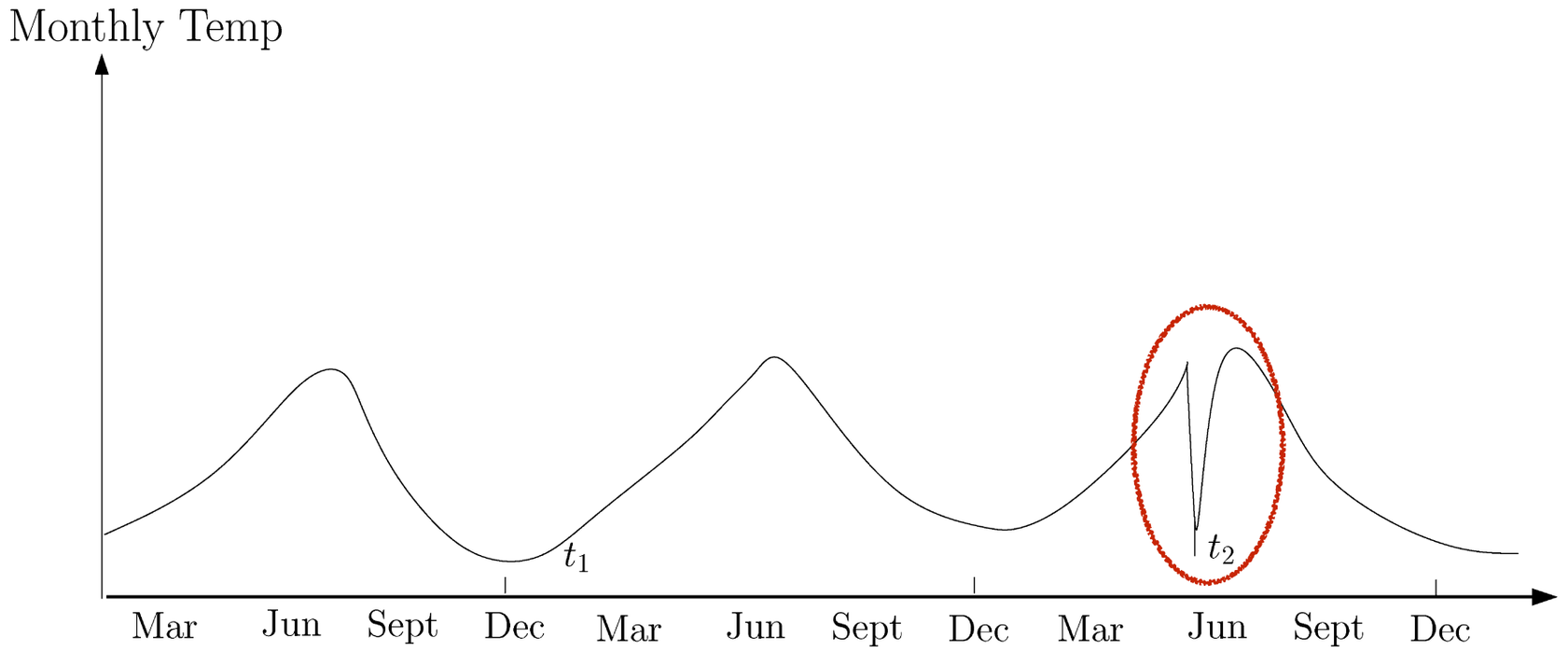
- Automatic identification of data instances (events) that are in some way different from the bulk of the data and which need detailed scrutiny by experts. Usually implied:
  - produced by a different mechanism than the bulk of the events
  - small number of anomalies w.r.t. the main part of the data
- Can be
  - supervised: train to recognise specific anomalous cases
  - semi-supervised: train only on the bulk of the data without anomalies → strong relation to one-class classification
  - unsupervised: algorithm automatically identifies the bulk by some means and thence the anomalies
- Difficult problem because in general we don't know what the anomalies look like, and there may be very few of them
  - Testing is particularly challenging: how do we evaluate the performance of an algorithm on a type of event that we have never seen before?

Types of anomaly

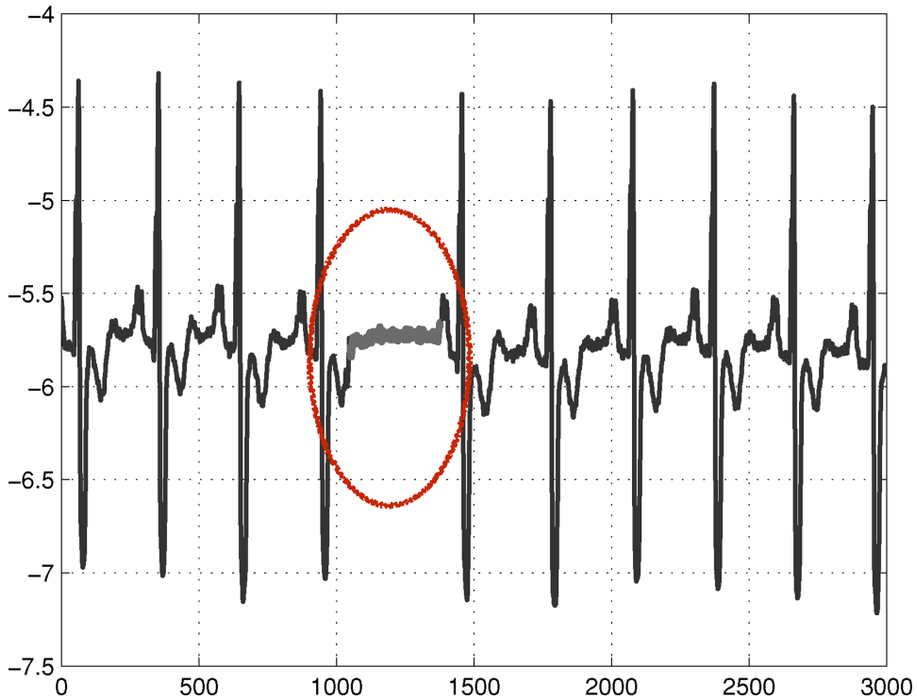
POINT



CONTEXTUAL

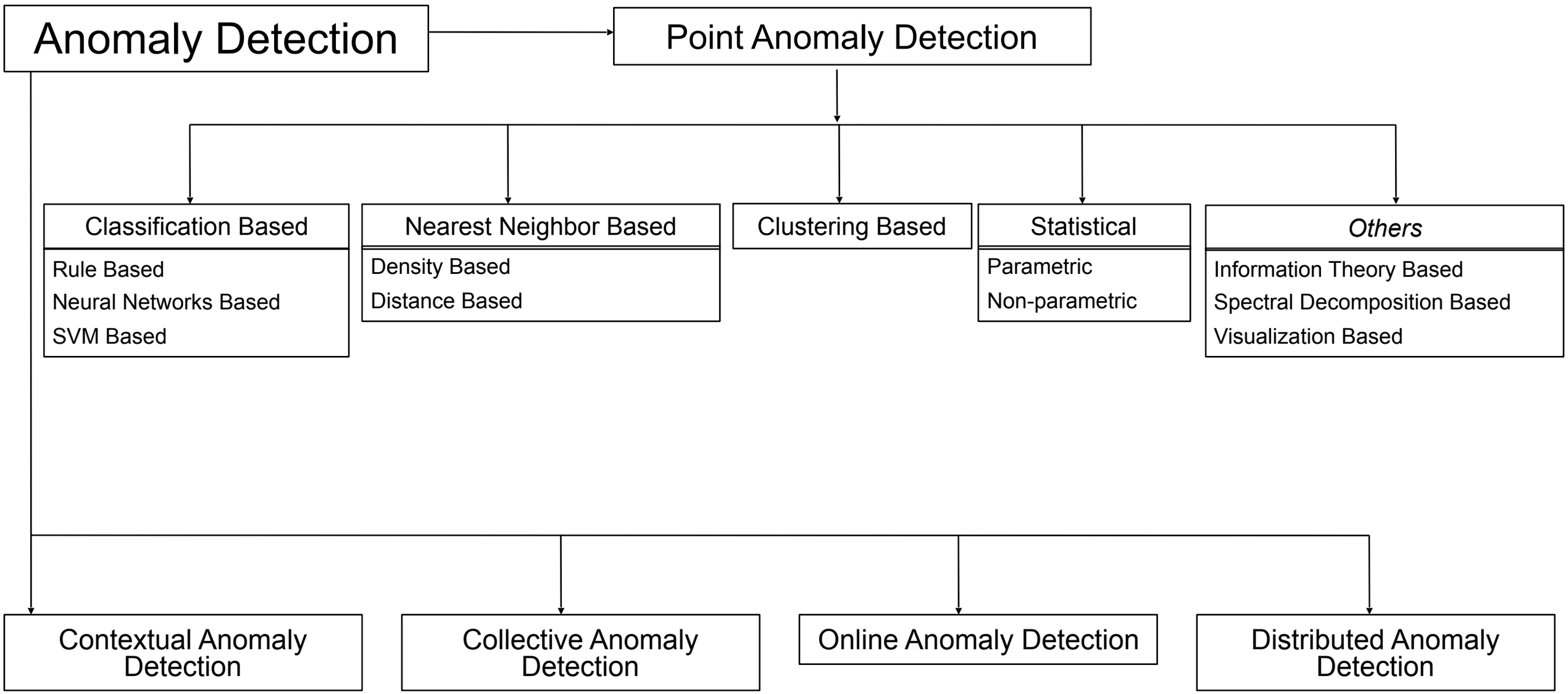


COLLECTIVE (population-level)



Images:
ACM Computing Surveys,
Vol 41, No. 3, Article 15

Taxonomy*



* Outlier Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, Technical Report TR07-17, University of Minnesota (Under Review)

United Technologies

University of Minnesota

Anomaly detection in HEP

- Monitoring and detection of problems in
  - DAQ and trigger
  - Distributed computing
  - Reconstruction and data quality monitoring
- Physics analysis
  - Unusual individual events
  - Unusual collective behaviour
- Topics for this talk
  - Report on the RAMP at the HEP software foundation workshop in Paris
  - Anomaly detection activities in ATLAS

RAMP at the Paris HSF workshop

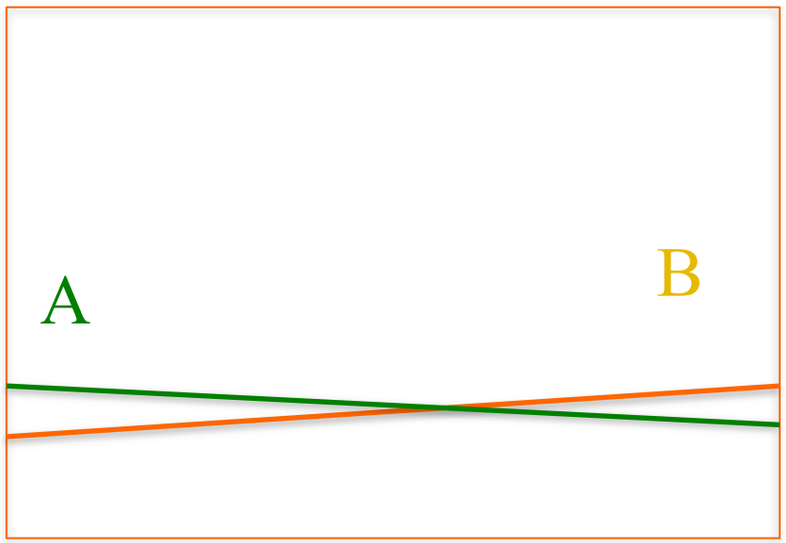
- The HEP Software Foundation held a workshop at LAL in Paris, May 2nd-4th
  - http://hepsoftwarefoundation.org/newsletter/2016/05/17/workshop-lal.html
- Machine Learning featured heavily at the meeting, as well as a decision on a logo for the HSF
- A full afternoon was devoted to a RAMP on anomaly detection

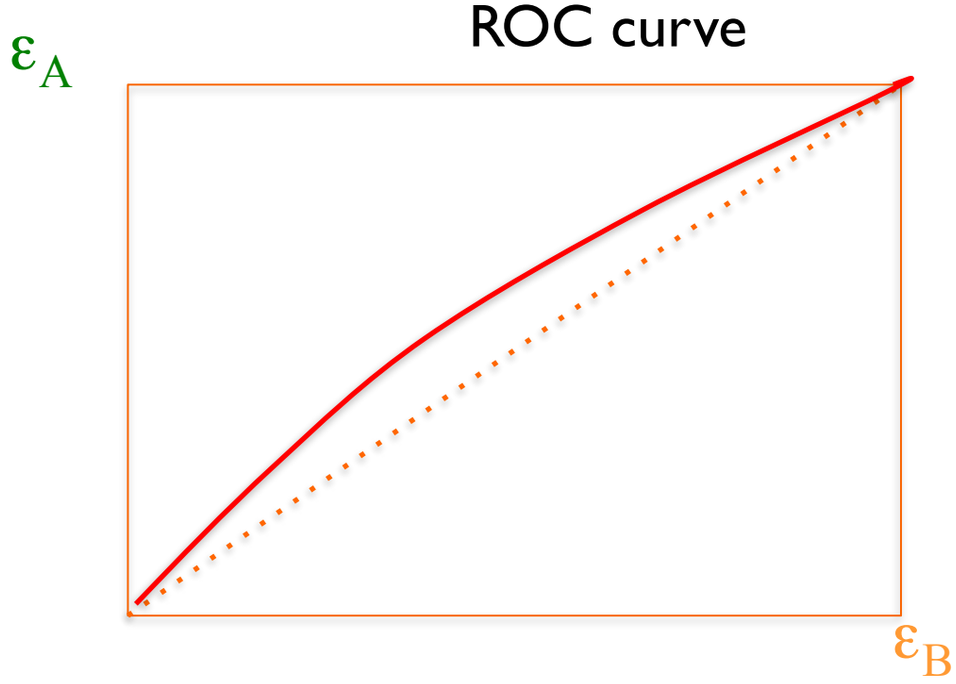
Aside: supervised (population-level) anomaly detection

- Suppose you have two independent samples A and B, supposedly statistically identical. E.g. A and B could be:
  - MC prod 1, MC prod 2
  - MC generator 1, MC generator 2
  - Derivation V12, Derivation V13
  - G4 Release 20.X.Y, release 20.X.Z
  - Production at CERN, production at BNL
  - Data of yesterday, Data of today
- How to verify that A and B are indeed identical ?
  - Standard approach: overlay histograms of many carefully chosen variables, check for differences (e.g. KS test)
  - ML approach: train a classifier to distinguish A from B, histogram the score, check the difference (e.g. AUC or KS test)
    - → only one distribution to check
    - If the classifier makes any headway in separating A from B, there must be some systematic difference

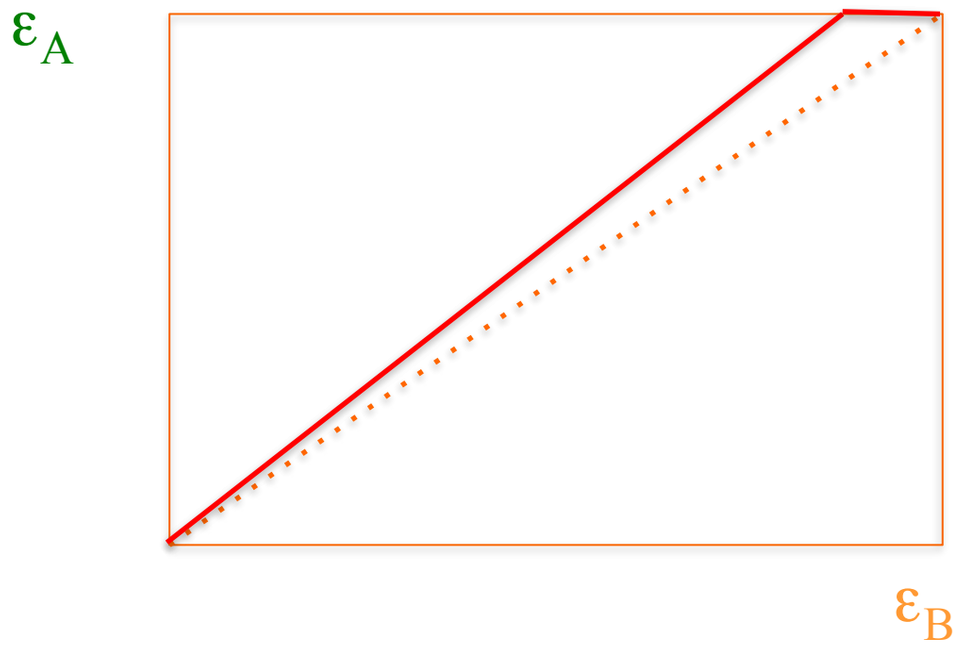
Aside: supervised (population-level) anomaly detection

Small non-local difference





Local big difference (e.g. non overlapping distribution, hole)



What is a RAMP?

- Rapid Analytics and Model Prototyping
- Real-time data challenge; ideal developed by the Paris-Saclay Centre for Data Science
  - Participants are in the same room, or at least participating in real time
  - They have access to a subset of the data (training) but then must submit their code to a server for evaluation on unseen data
  - Results appear on a leader board, based on some performance metric (see later)
  - Once submitted, code can be seen and cloned by all participants, and thus improved ("competitive-collaborative")
  - Uses Python; allows use of SciKitLearn (etc) and IPython notebook
- Full details: https://indico.cern.ch/event/496146/contributions/1174809/attachments/1268459/1878677/slidesEID1603.pdf

Download starter kit

```
jupyter notebook
```

Appears in browser

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New

HEP_detector_anomalies_starting_kit.ipynb ← Click

classifier.py

public_train.csv.gz

user_test_submission.py

In [1]:
```
from __future__ import division, print_function
import os
import sys
import glob

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

try :
    import seaborn as sns; sns.set()
except ImportError:
    print("seaborn not found on your computer. "
          "Install it if you want pretty charts \n"
          "If you have internet access you can run in a cell : \n"
          "!pip install -U seaborn ")

pd.set_option('display.max_columns', None)
```

In [2]:
```
%matplotlib inline
```

Exploratory data analysis

Loading the data

In [3]:
```
data = pd.read_csv('public_train.csv.gz', compression='gzip')
X_df = data.drop(['isSkewed'], axis=1)
Y_df = Y_df = data[['isSkewed']]
```

In [4]:
```
data.head()
```

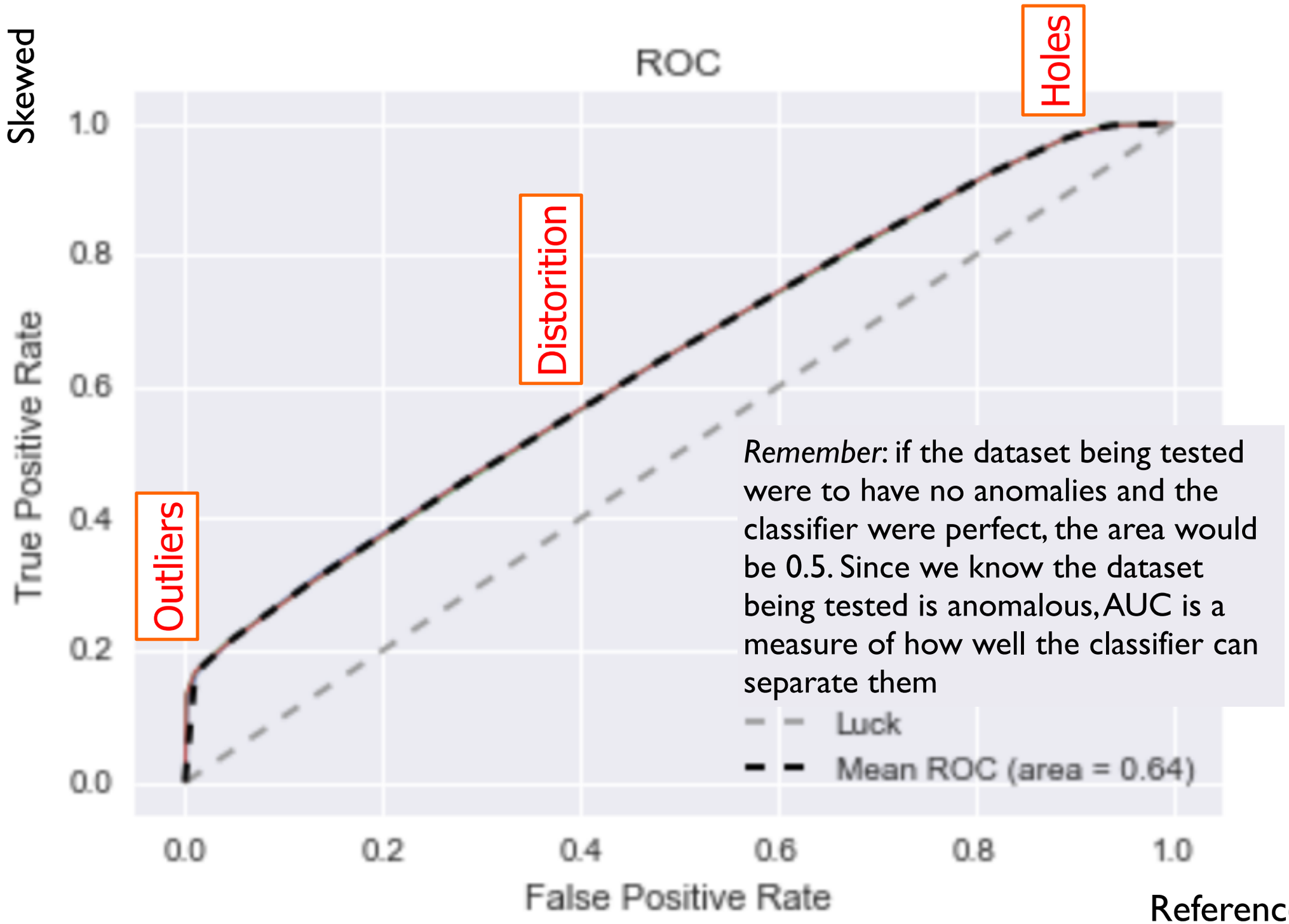
Out[4]:

	DER_mass_transverse_met_lep	DER_mass_vis	DER_pt_h	DER_deltar_tau_lep	DER_pt_tot	DER_sum_pt	DER_pt_ratio_lep_tau	DER_met_
0	1.937	64.546	41.791	2.301	7.975	105.305	0.926	1.087
1	69.892	61.447	157.875	1.618	2.756	238.670	1.239	1.311
2	4.252	73.423	56.958	3.018	77.300	266.566	0.823	-0.622
3	76.695	60.761	1.770	2.077	1.770	70.690	1.578	-1.381
4	21.288	60.049	13.166	2.966	71.566	265.146	0.826	1.393

Anomaly detection RAMP

- Approximately 30 participants (not all submitted code)
- Used the HiggsML dataset: http://opendata.cern.ch/record/328
  - H→ττ and SM backgrounds
  - Primary variables: lepton, and tau hadron 3-momentum, MET
  - Derived variables (computed from the above) from Htautau analysis
  - Jet variables dropped (to avoid -999.0 when no selected jets available)
- "Reference dataset": subset of the above
- "Distorted dataset": different subset, distorted as follows (one per event):
  - Small scaling of p(τ) [DISTORTION]
  - Holes in ηϕ efficiency map of lepton and τ hadron [HOLE]
  - Outliers introduced, each with 5% probability [OUTLIER]
  - η(τ) set to large non possible values [OUTLIER]
  - P lepton scaled by factor 10 [DISTORTION]
  - Missing ET + 50 GeV [DISTORTION]
  - Phi tau and phi lepton swapped ➔ derived variables inconsistent with primary one [DISTORTION]
- Participants provided with part of both of the above, remainder held back to build the leader board
  - This is supervised anomaly detection, since the classifiers see both the anomalies and the normal data in training

Performance metric: area-under-ROC (AUC)



Remember: if the dataset being tested were to have no anomalies and the classifier were perfect, the area would be 0.5. Since we know the dataset being tested is anomalous, AUC is a measure of how well the classifier can separate them

Reference

Leader board

Breakthrough : add new variable:

$\Delta m_T = \sqrt{(2P_{lT} * MET * (1-\cos(\varphi_l - \varphi_{MET})))} - m_T$

Non zero for some outliers

➔classifiers were unable to guess it

http://www.ramp.studio/events/HEP_detector_anomalies/leaderboard

Combined score: 0.677

Leaderboard

team	submission	accuracy	nll	auc	contributivity	historical contributivity	train time	test time
dhrou	adab2_mt1	0.611	0.633	0.675	35	22	21	1
mcherti	adab2_mt1_calibrated	0.611	0.600	0.675	7	0	42	5
kegl	adab2_mt1_calib_25	0.610	0.600	0.673	2	0	45	6
glouppe	boosting-duo	0.595	0.617	0.656	6	6	300	3
glouppe	bags2	0.594	0.618	0.656	6	0	950	10
kazeevn	GradientBoosting	0.596	0.617	0.656	2	0	837	4
mcherti	adaboost2	0.594	0.644	0.655	19	0	16	1
dhrou	adab2_mt1_log1	0.594	0.644	0.655	0	0	20	1
dhrou	adab2_mt1_log1_dt1	0.594	0.644	0.655	0	0	39	3
glouppe	bags	0.593	0.619	0.654	0	5	2112	21
djabbz	beta tester	0.591	0.618	0.653	0	0	12	1
mcherti	adaboost1	0.593	0.634	0.653	11	52	8	0
soobash	ExtraTreesClassifier	0.576	0.651	0.619	0	0	3	1
mcherti	extratrees1	0.562	0.669	0.596	1	1	6	4
dhrou	DRv0	0.553	0.653	0.577	2	0	0	0
calaf	starting_kit_paolo	0.526	0.671	0.552	2	0	0	0
kegl	starting_kit2	0.526	0.671	0.552	0	0	1	0
marcevrard	keras_nn_2	0.527	0.670	0.532	0	0	1	0
marcevrard	keras_nn_1	0.527	0.670	0.532	0	0	1	0
kegl	starting_kit_copy	0.527	0.670	0.532	6	0	1	0
FabriceJourdan	Fabrice01	0.527	0.670	0.532	0	0	1	0

RAMP conclusions

- RAMP-style competitions can be highly productive and lead to rapid developments
- Unlike a data challenge:
  - Submitted code is viewable and usable by all participants
  - Participants can discuss with each other (since they are generally in the same room)
  - Positive social aspects (e.g. coffee breaks, drinks/dinner afterwards)
  - Ideal to hold as part of a larger meeting/workshop/conference as we did at HSF
- Hopefully some of the participants at the HSF event will now apply their ideas in their own collaborations
- RAMP platform: http://www.ramp.studio/

ATLAS anomaly detection activities

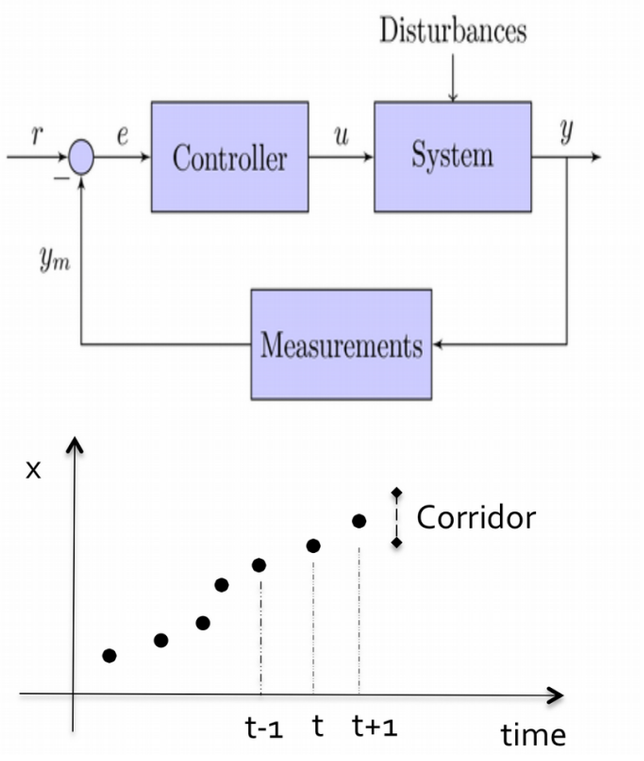
- DAQ: contextual anomaly detection in a time series
- Distributed computing
- Data quality monitoring and physics analysis (one class classification)

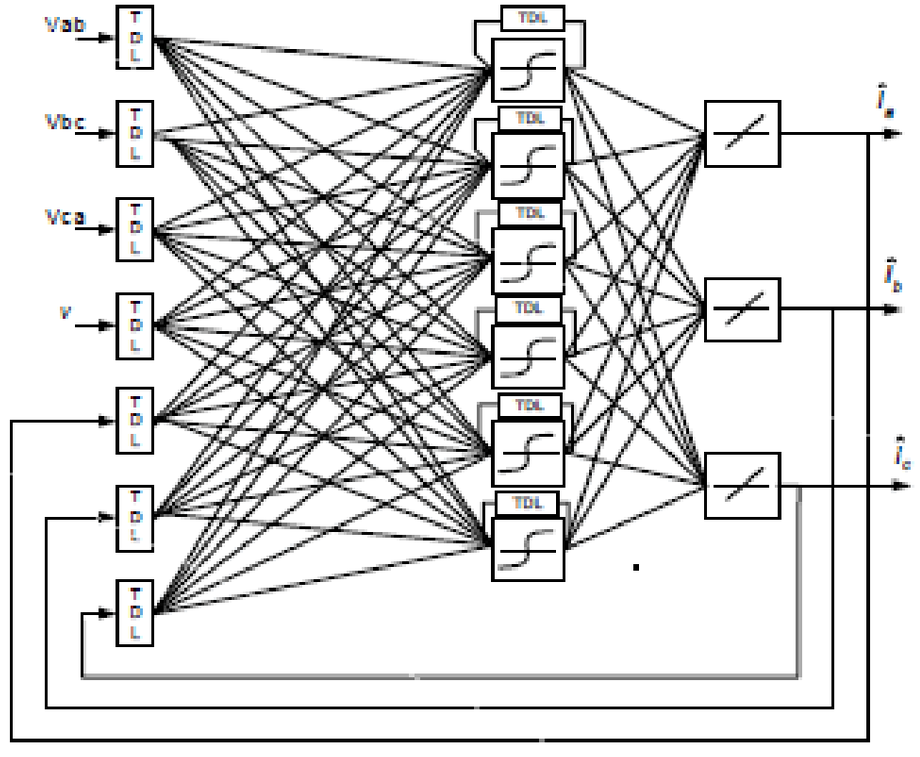
Anomaly detection in DAQ

- The DAQ system publishes several values (luminosity, L1 rates, dead times,...) during operation. These values over time create time series that can be monitored for anomalies
- Anomaly detection can be seen as a data quality problem in the time series. An anomaly can be detected as an outlier
- Thus, the goal of the project is to detect and predict anomalies online in the DAQ system and issue a warning

Proposed approach

- A neural network(NARX) trained to predict the next value in the time series will build a validation corridor.





Corridor defined by: neural_network_prediction - dx , neural_network_prediction + dx

dx = k * (sum of variances so far)

Anomaly detection in Distributed Computing

- Distributed computing systems used by ATLAS process around a million events per day, and transfers between WLCG sites is about 6PB per week
- Such a heavily used and complex system inevitably experiences degradations and so LHC data management and processing must tolerate a continuous stream of failures, errors, and faults
- Traditional approach: re-try the failed jobs, and keep on re-trying until they succeed
  - Inefficient, leads to unpredictable delays
- Machine-learning could be applied to the modelling of data processing and data management
  - guide the application of novel fault tolerance strategies
  - significantly reduce turnaround times for production
- Particular strength of this idea is the rich body of metadata generated by the production system during the past years of operation
  - Treasure trove of training data

Anomaly detection in Distributed Computing

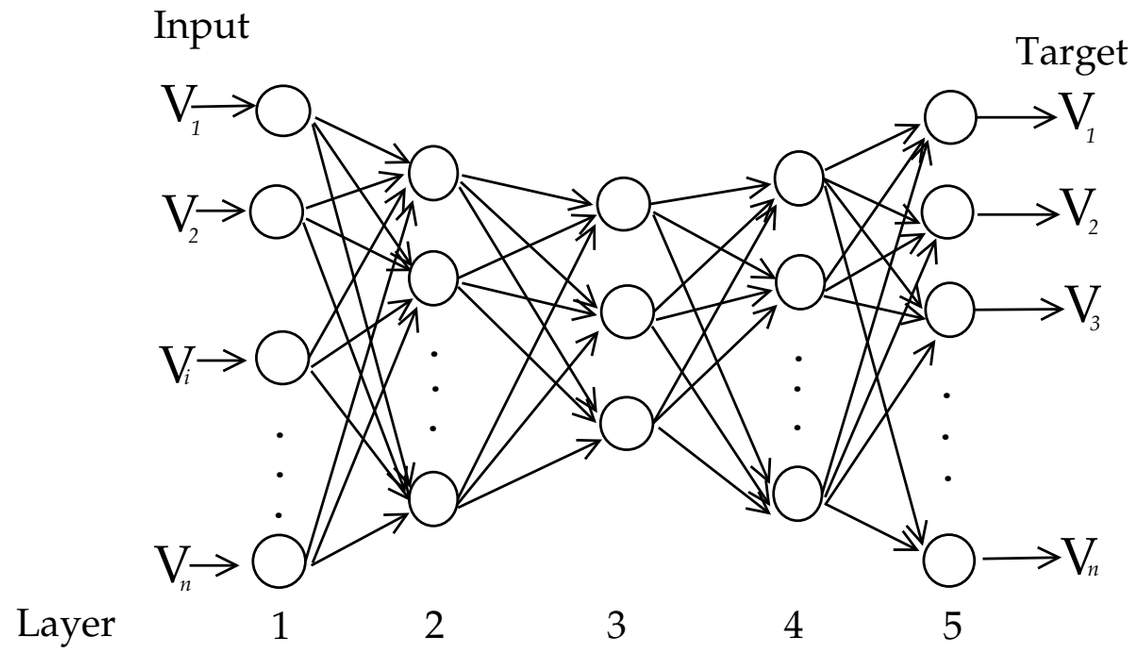
- WLCG demonstrator project, jointly with LHCb
  - investigate the nature of the failures and anomalies in the distributed computing infrastructure of WLCG, including networks
  - develop a data-driven model of short-lived correlated failures producing observed heavy-tail, non-Gaussian distributions in the evolving stream of random failures
  - apply time series modeling and online learning techniques to the continuous stream of information about processing failures and retries
  - identify clustered failures, and to determine failure characteristics that could support automated decision making regarding retry strategies
- Example
  - certain kinds of clustered failures are likely to be site- or node-specific, or rather indicative of issues that might occur more widely
  - can guide automated decision making, helping to determine whether it is worthwhile to rerun the job on the same site or somewhere else

Data quality monitoring

- Reminder: are two datasets that are supposed to be statistically compatible, really compatible?
- Two approaches under study:
  - Supervised approach as per the RAMP: classifier trained on part of the reference AND the dataset to be tested, and asked to separate them
  - One-class classifier that only sees the "reference" data [semi-supervised]
- Start with minimum-bias datasets, with reference ("good") datasets, and those with known defects in (e.g.) alignment, different regions of the detector, etc

One class classifier: auto-encoder

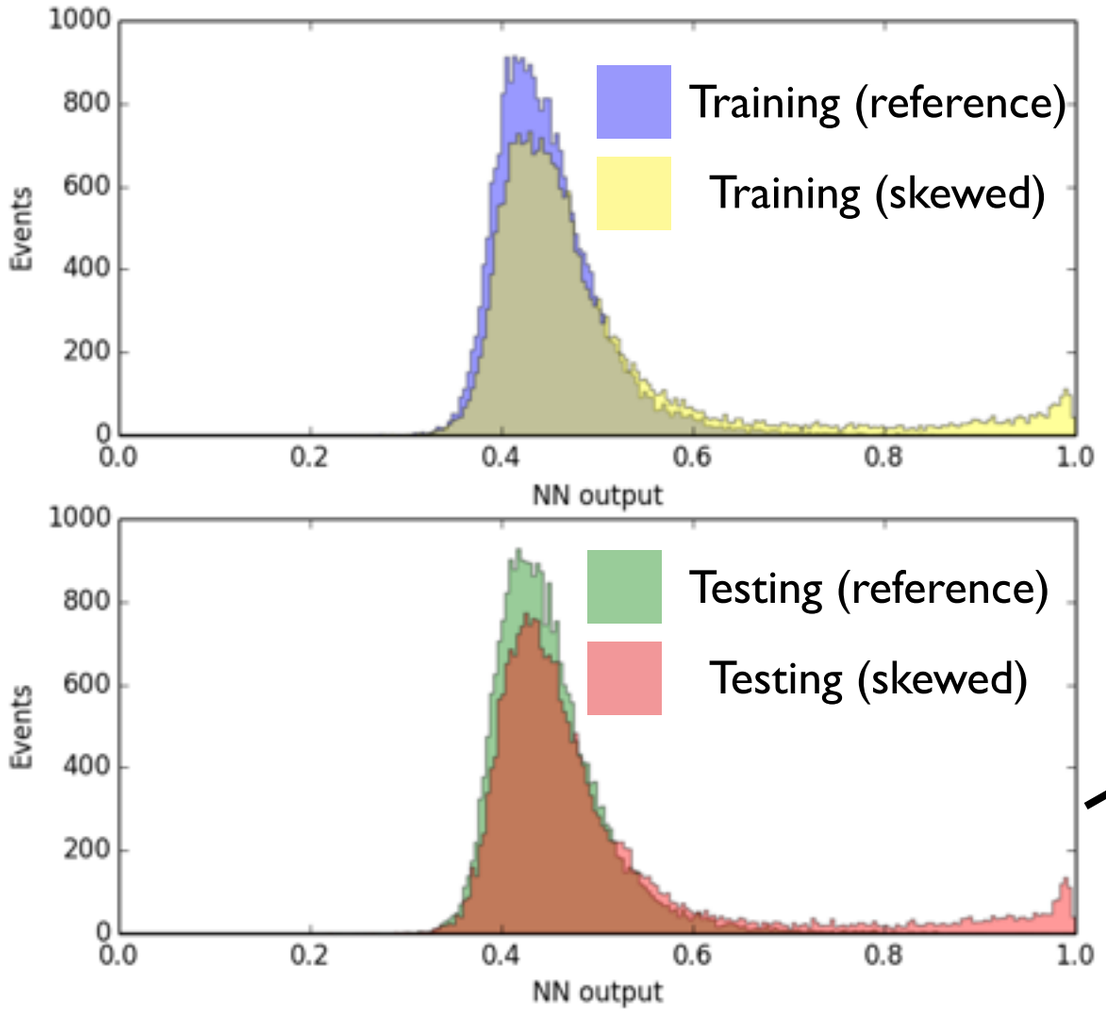
- Auto-encoder: NN trained on its own input
  - Usually includes a bottle-neck to compress the features of the data (e.g. PCA)

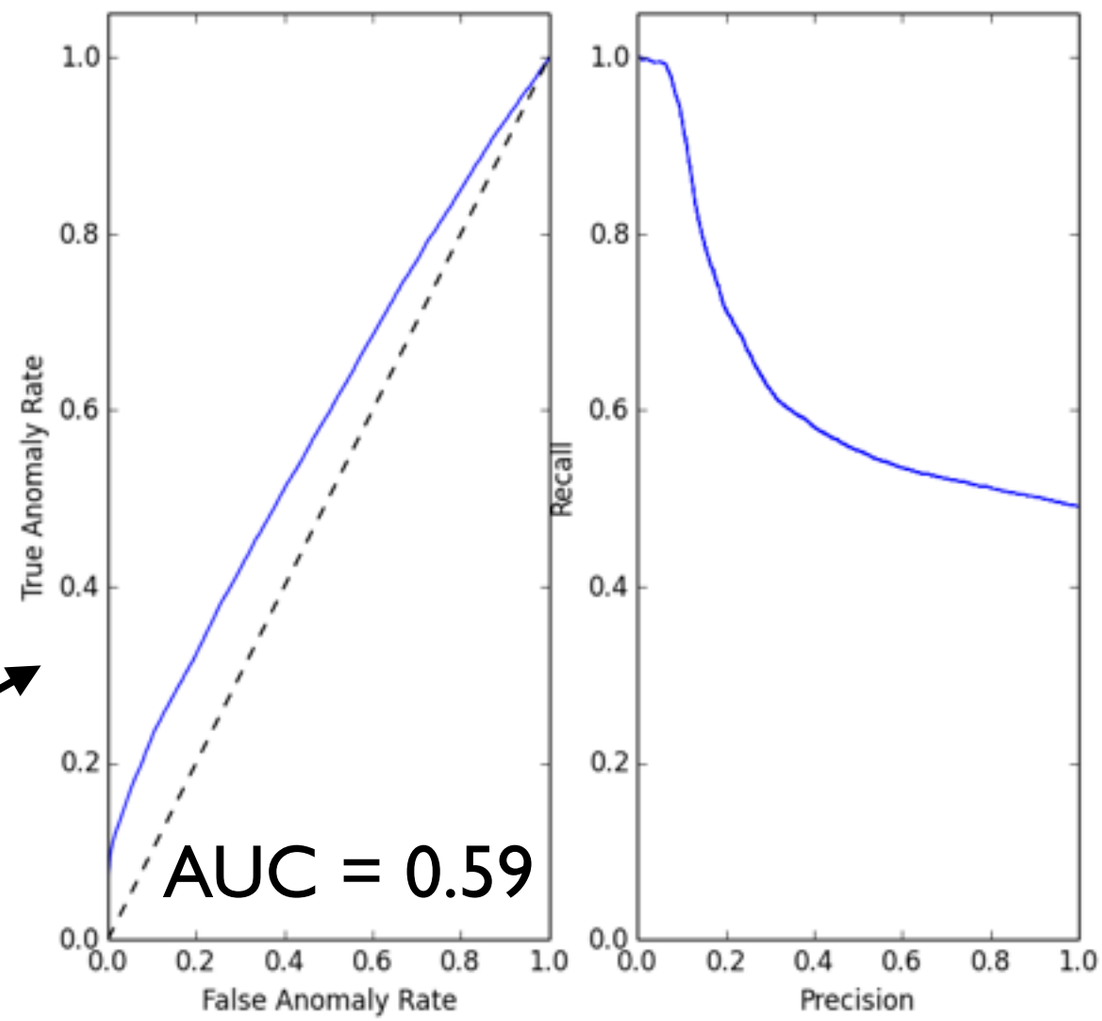


- Normally used for dimension-reduction but proposed as a means of anomaly detection in (e.g.)
  - http://link.springer.com/chapter/10.1007%2F3-540-46145-0_17
  - http://link.springer.com/chapter/10.1007%2F978-3-319-13563-2_27
- Idea: a trained replicator neural network should reconstruct new examples taken from the bulk (normal) data with low error, but when presented with an anomalous example, will reconstruct it with a high error since it contains qualities that have not previously been encoded
  - Provides a natural measure of abnormality: the reconstruction error (difference between the input and the output)
  - Reconstruction error per event = $\sum_{i=1}^{N}(x_i^{in} - x_i^{out})^2$ N is the number of features

Toy illustration (using the RAMP dataset)

Trivial neural network classifier (1 hidden layer, 15 units, ReLU activation)
Trained to distinguish the reference from the skewed dataset

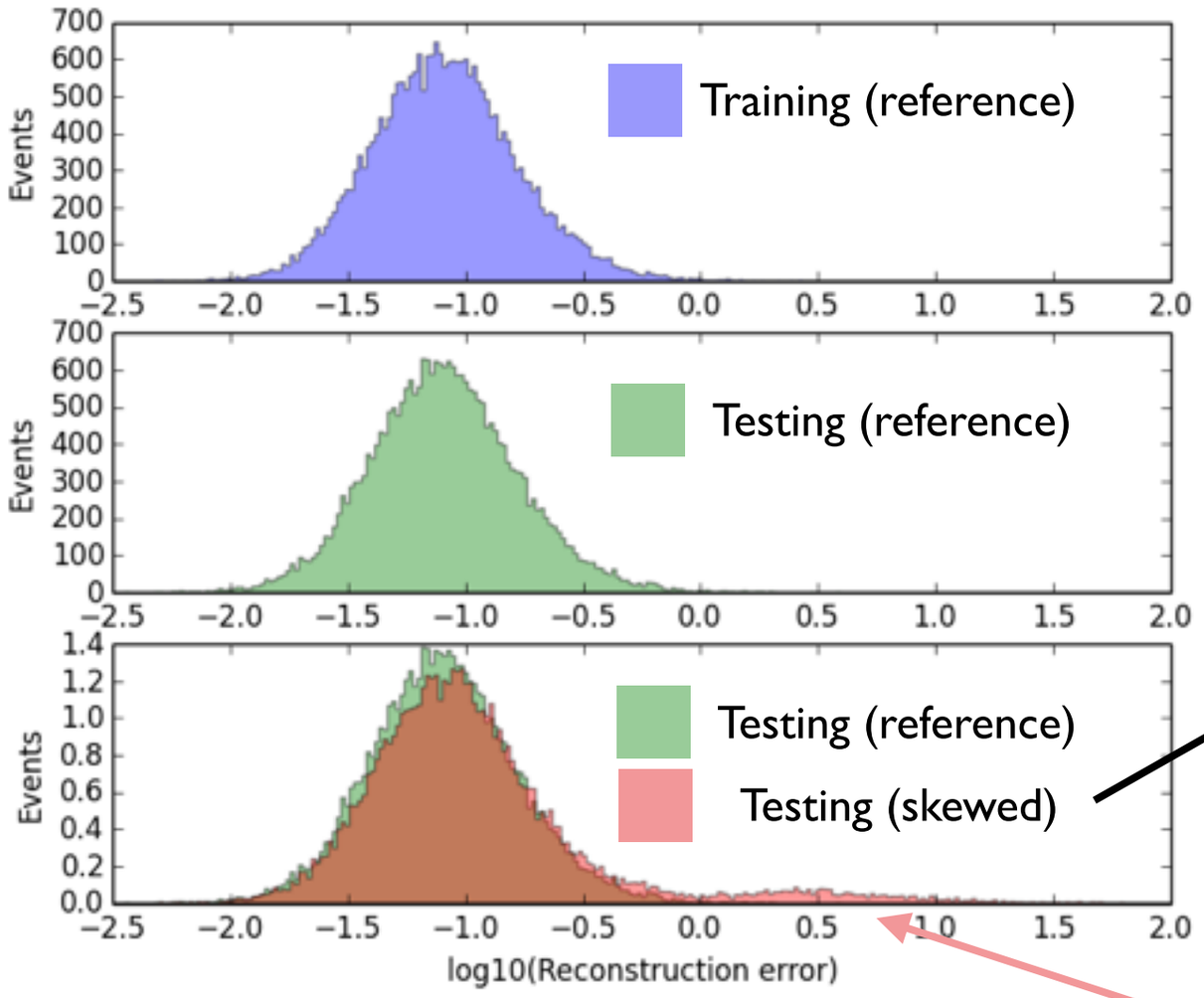


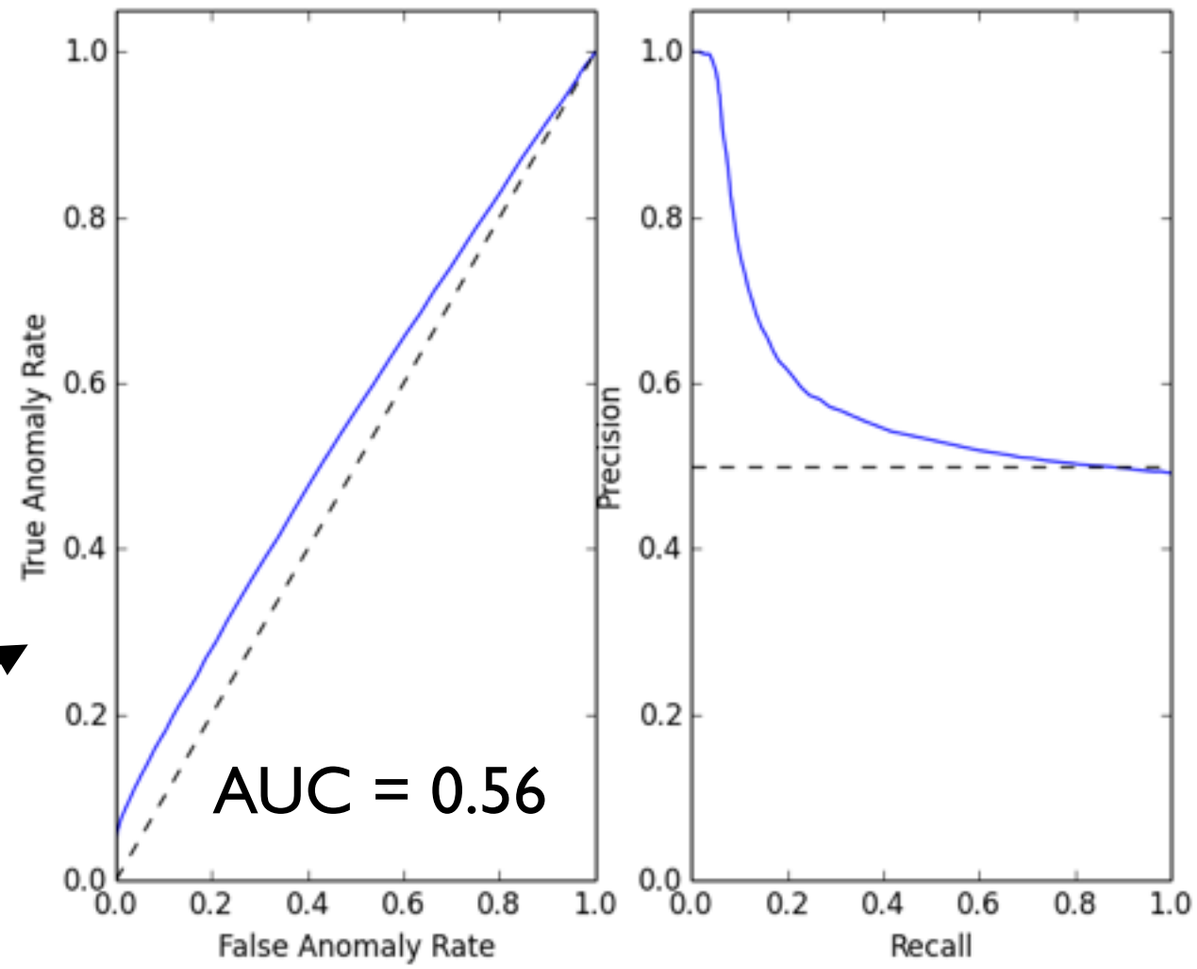


Rather poor performance compared to the best RAMP results (no extra variables),
but can clearly be seen to be picking out differences between the datasets -
you wouldn't sign off the skewed dataset as being compatible with the reference!

Toy illustration (using the RAMP dataset)

Autoencoder (1 hidden layer, 15 units, ReLU activation)
Trained to reconstruct (apply the identity function on) the reference dataset





Poorer performance still, but still identifies differences despite not having seen any skewed data in the training

Other activities

- So far we have only scratched the surface
- Other potential methods, applications
  - Use in early warning systems (picking out strange events for detailed scrutiny as they are reconstructed)
  - One-class classification in physics analysis?
    - Only train classifiers on background samples?
    - Can they outperform traditional classifiers trained on signal MC, where the testing signal differs significantly from the training? [arXiv:1112.3329v3]
- Plenty of interesting work to do!