Anomaly detection in the CERN cloud infrastructure

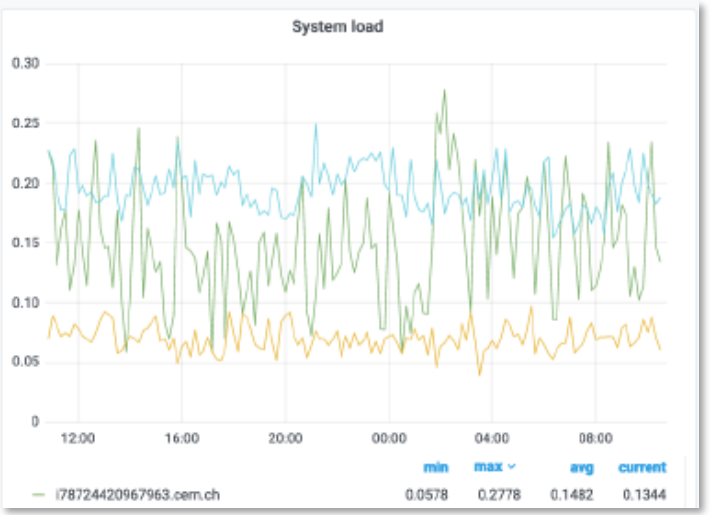
D. Giordano[1], M. Paltenghi[1], S. Metaj[1], A. Dvorak[2]

[1]CERN/IT, [2]Nuclear Physics Institute of the Czech Academy of Sciences

25th International Conference on Computing in High-Energy and Nuclear Physics 19 May 2021

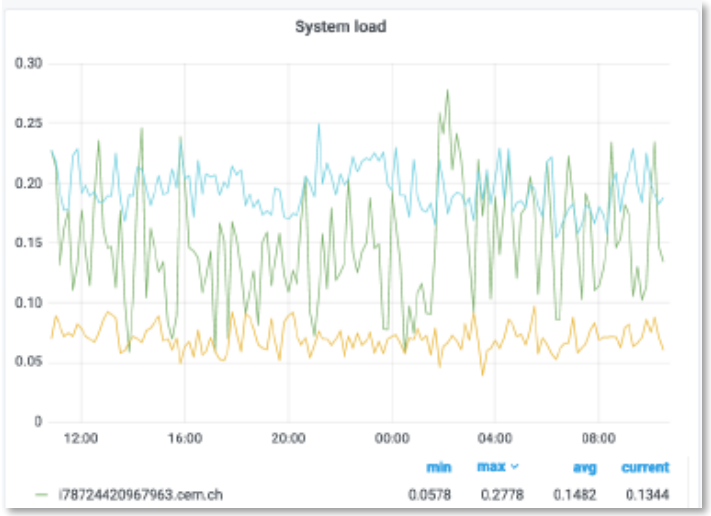
Motivations and Case of Study

- CERN Cloud Infrastructure: **~8k** bare-metal servers serving **~35k** VMs
- Monitoring:
  - Manual data exploration, thresholds based alarming, post-mortem analysis
  - **~150 Collectd** time-series metrics (cpu, mem, disk …) + logs



Motivations and Case of Study

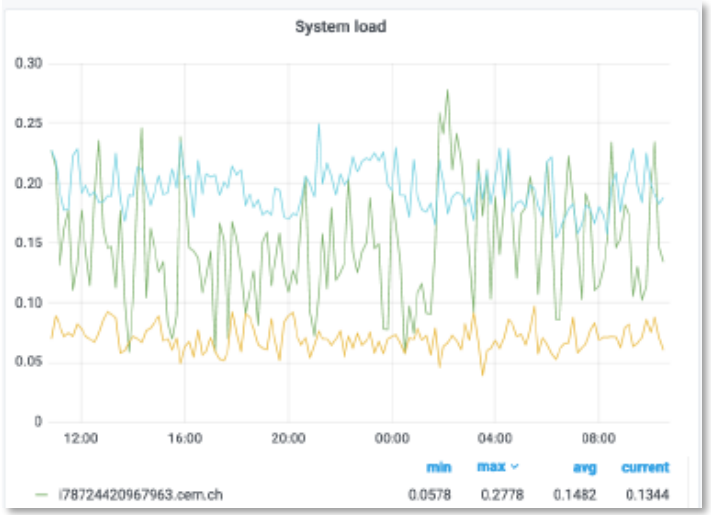
- CERN Cloud Infrastructure: **~8k** bare-metal servers serving **~35k** VMs
- Monitoring:
  - Manual data exploration, thresholds based alarming, post-mortem analysis
  - **~150 Collectd** time-series metrics (cpu, mem, disk …) + logs



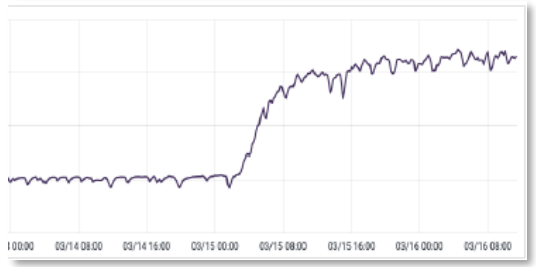
- Goals: automate the Anomaly Detection task, discover misbehaviors earlier, consider metrics correlation

Motivations and Case of Study

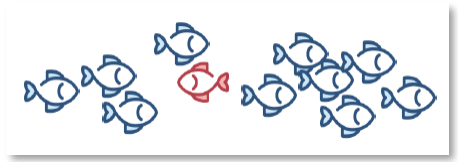
- CERN Cloud Infrastructure: **~8k** bare-metal servers serving **~35k** VMs
- Monitoring:
  - Manual data exploration, thresholds based alarming, post-mortem analysis
  - **~150 Collectd** time-series metrics (cpu, mem, disk …) + logs



- Goals: automate the Anomaly Detection task, discover misbehaviors earlier, consider metrics correlation



- Two possible approaches:
  - **Change Detection**: different behavior of a single machine w.r.t. its own past
  - **Outlier in Cloud Hostgroup**: different behavior of a single machine w.r.t. the other machines in the same Hostgroup (O(10^2) homogeneous machines, same HW/SW)



Anomaly Detection System Implementation

- Desire to cover the entire chain from input data sources to end-user GUIs
- Data Analytics Pipeline: produce the anomaly results
- Annotation Pipeline: to label data and create benchmark dataset

Domain Expert

Feedback

Individual Methods

Data source

Data Preparation

Unsupervised Anomaly Detection Methods

Traditional

Deep learning

Ensemble Strategy

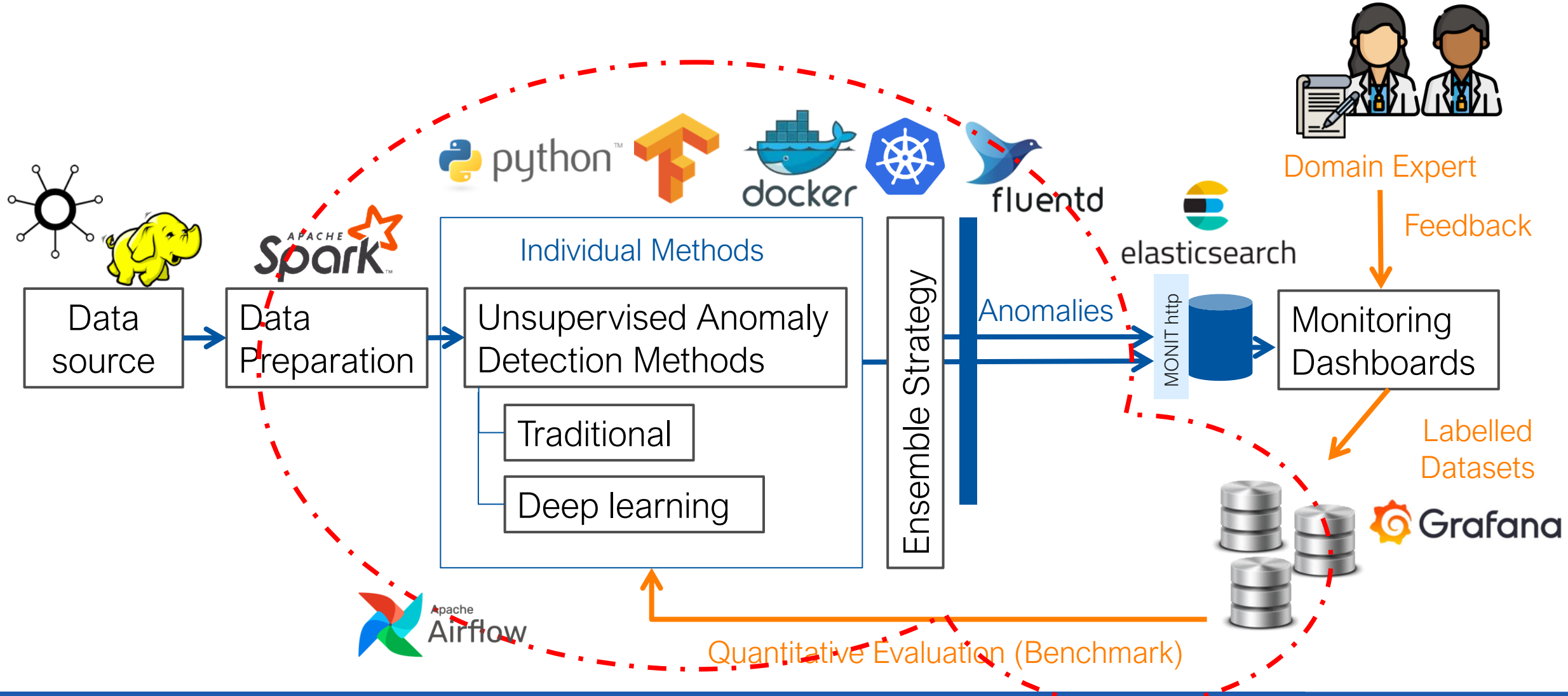
Anomalies

Monitoring Dashboards

Labelled Datasets

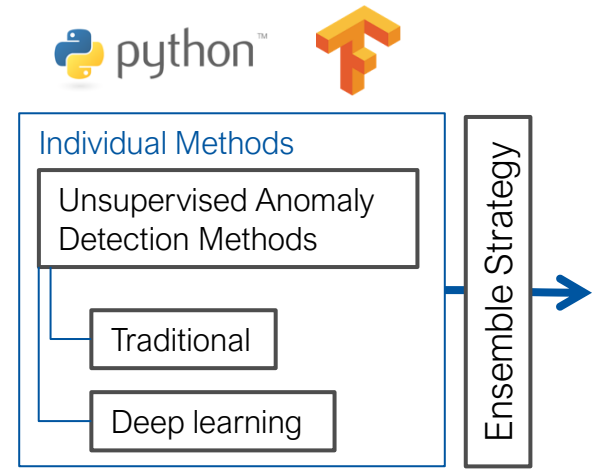
Quantitative Evaluation (Benchmark)

Anomaly Detection System Implementation



Anomaly Detection Algorithms

- Comparison of 10 unsupervised Anomaly Detection methods:
  - Use of the PyOD library for traditional methods
  - Adapt existing methods to work on time series data
  - Study of ensemble strategies



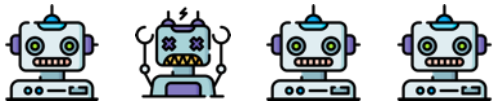
Traditional AD Methods	Deep Learning Methods	Ensemble Strategies
• K-Nearest Neighbours • Local Outlier Factor • One Class SVM • Isolation Forest • Principal Component Analysis *Time Series Specific* • Vector Autoregression Forecaster*	• Fully Connected Autoencoder • CNN Autoencoder *Time Series Specific* • LSTM Autoencoder • CNN Forecaster*	• Min • Max • Cumulative Sum • Average • Median • Linear Regression*

* Novel variations

Anomaly Detection Algorithms

- Scoring function that assigns to each sample an anomaly score indicating its degree of anomalousness (then step function to get 0 or 1)

Servers in the same Hostgroup

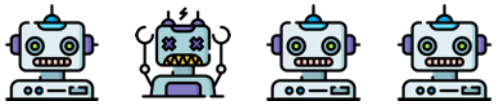


0.03 3.24 0.12 0.08

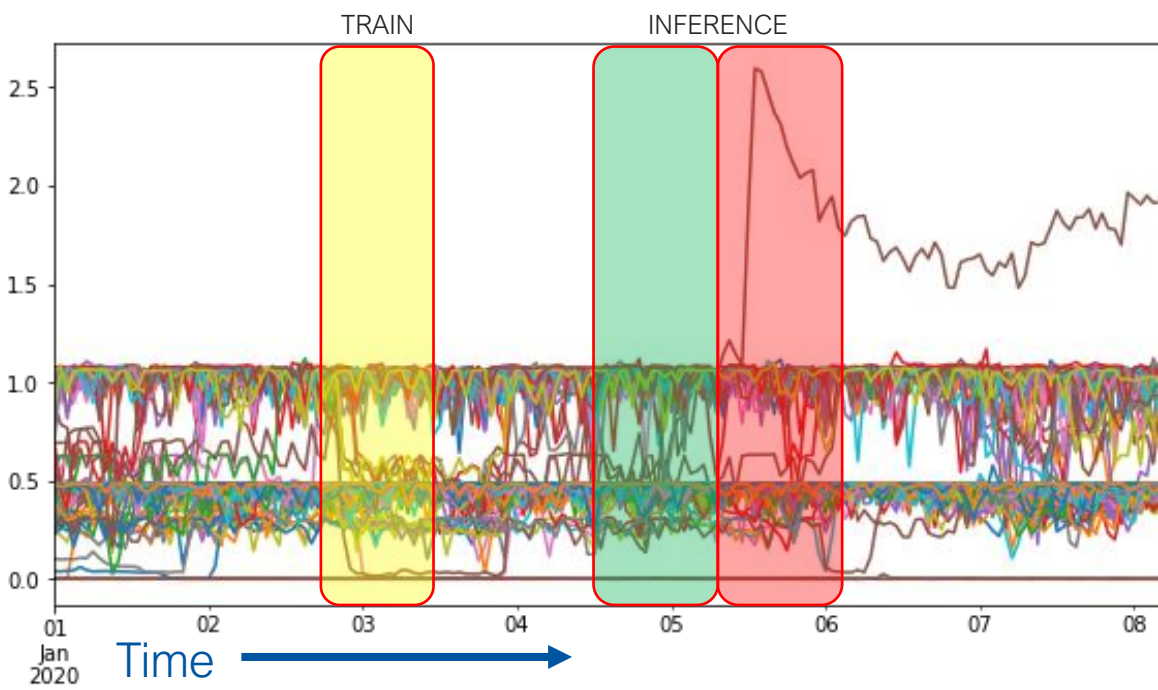
Anomaly Detection Algorithms

- Scoring function that assigns to each sample an anomaly score indicating its degree of anomalousness (then step function to get 0 or 1)
- Time windows of 8 hours (configurable) $W_{i,w}^{k}(h) = \{\vec{m}_i(h), \vec{m}_{i+1}(h), ..., \vec{m}_{i+w-1}(h)\}$

Servers in the same Hostgroup

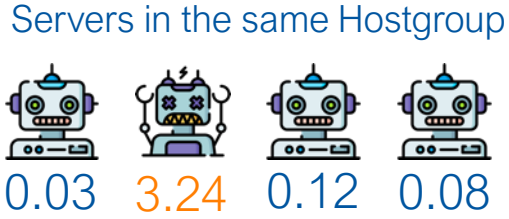


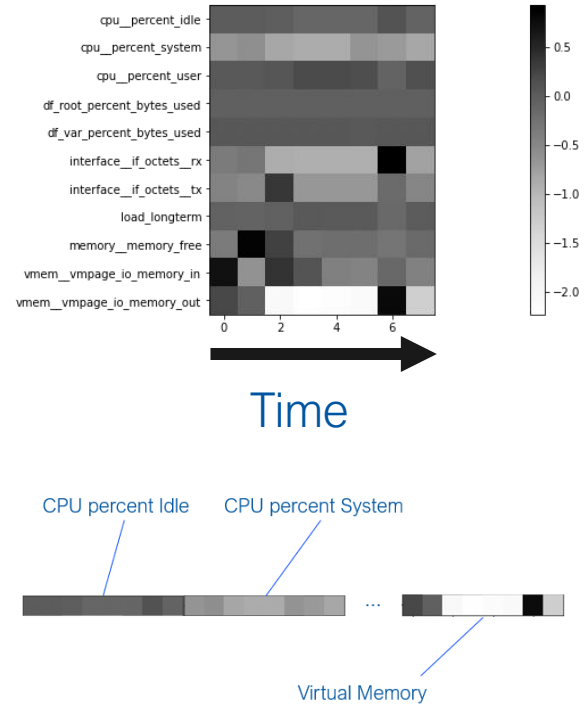
0.03 3.24 0.12 0.08



Anomaly Detection Algorithms

- Scoring function that assigns to each sample an anomaly score indicating its degree of anomalousness (then step function to get 0 or 1)
- Time windows of 8 hours (configurable) $W_{i,w}^{k}(h) = \{\vec{m}_i(h), \vec{m}_{i+1}(h), ..., \vec{m}_{i+w-1}(h)\}$
- Selected 11 performance metrics as input and performed window encoding
  - **Image-Like ("Grey-Map")**: time information, used with CNN and LSTM methods
  - **Vectorization**: compatible with traditional methods (LOF, IForest, PCA, OCSVM)

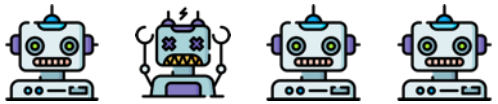




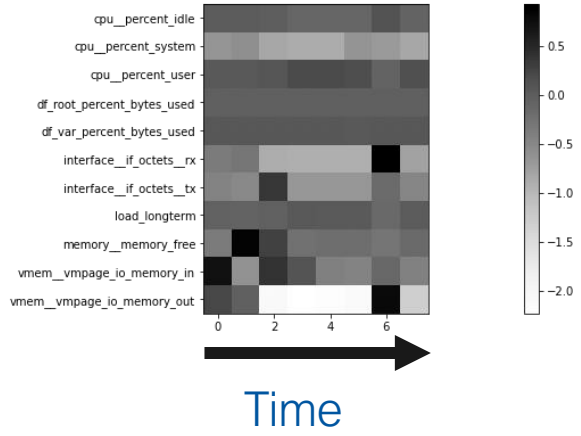
Anomaly Detection Algorithms

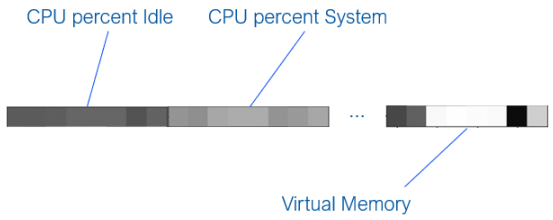
- Scoring function that assigns to each sample an anomaly score indicating its degree of anomalousness (then step function to get 0 or 1)
- Time windows of 8 hours (configurable) $W_{i,w}^{k}(h) = \{\vec{m}_i(h), \vec{m}_{i+1}(h), ..., \vec{m}_{i+w-1}(h)\}$
- Selected 11 performance metrics as input and performed window encoding
  - **Image-Like ("Grey-Map")**: time information, used with CNN and LSTM methods
  - **Vectorization**: compatible with traditional methods (LOF, IForest, PCA, OCSVM)

Servers in the same Hostgroup

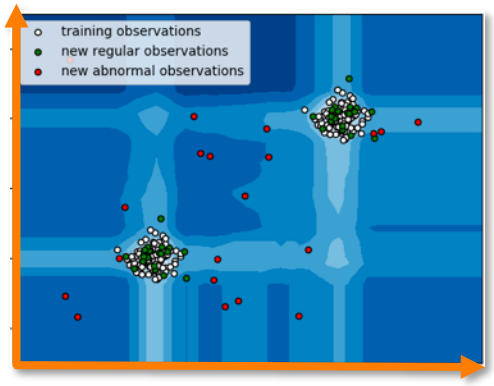


0.03 3.24 0.12 0.08

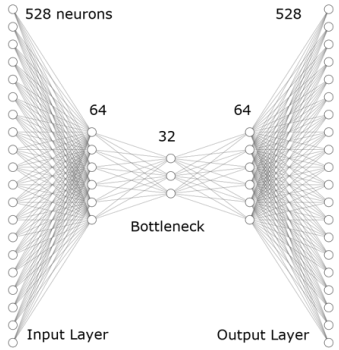




Isolation Forest (2008):
Ensemble technique based on Isolation Trees



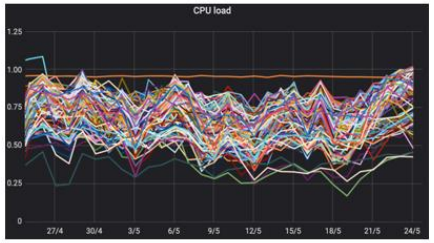
AutoEncoders:
Reconstruction Objective (also using Convolutions or LSTM)

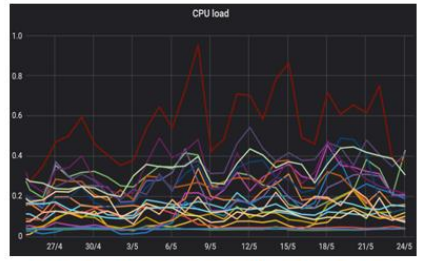


Evaluation and Results

- To be threshold independent, the performances are estimated using the AUC of the ROC curve
- Different experiments for different user categories: **Batch** and **Shared**
- Selected 1 Hostgroup per category, 8 months of data analysed

Batch category
Machines used for Batch computing



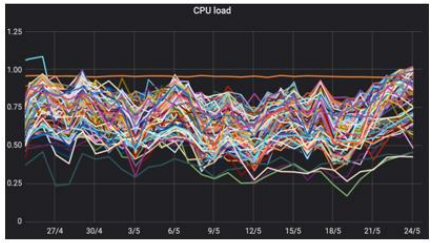


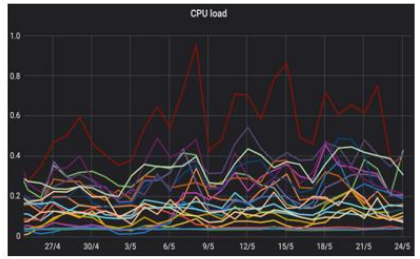
Shared (services) category
Machines used by different users and services.
More complex, not easily predictable

Evaluation and Results

- To be threshold independent, the performances are estimated using the AUC of the ROC curve
- Different experiments for different user categories: **Batch** and **Shared**
- Selected 1 Hostgroup per category, 8 months of data analysed

Batch category
Machines used for Batch computing

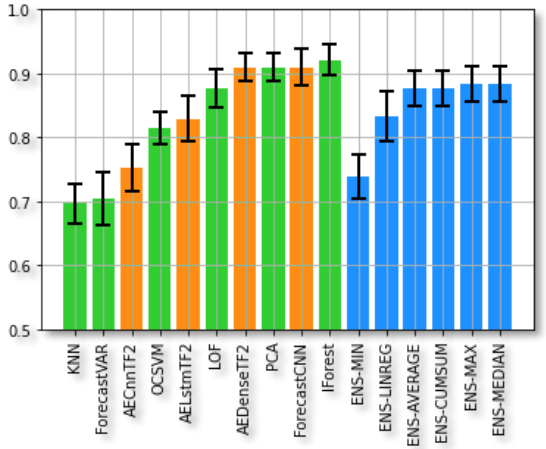




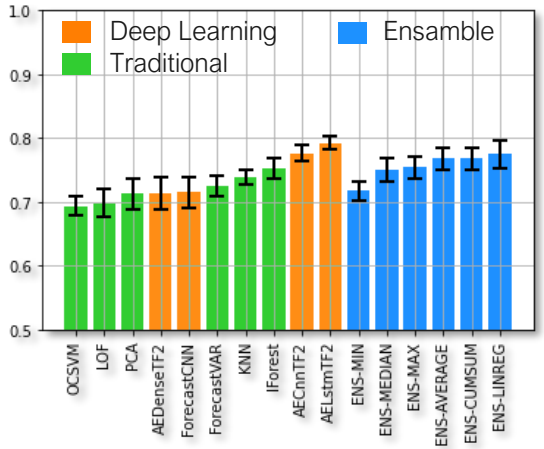
Shared (services) category
Machines used by different users and services.
More complex, not easily predictable

- The AUC-ROC is measured on several, independent weeks.
  We show the **average** performances (together with the **std**).

"Batch" category



"Shared" category

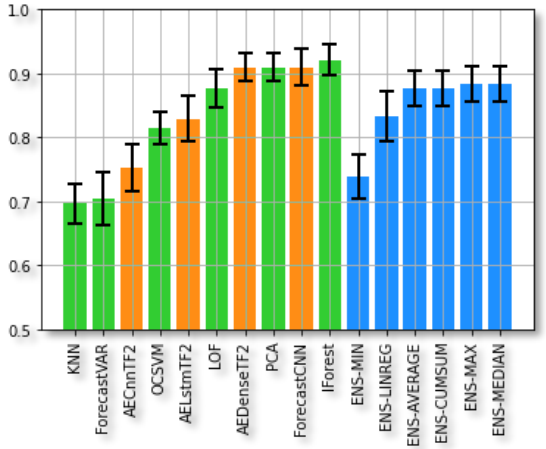


Evaluation and Results

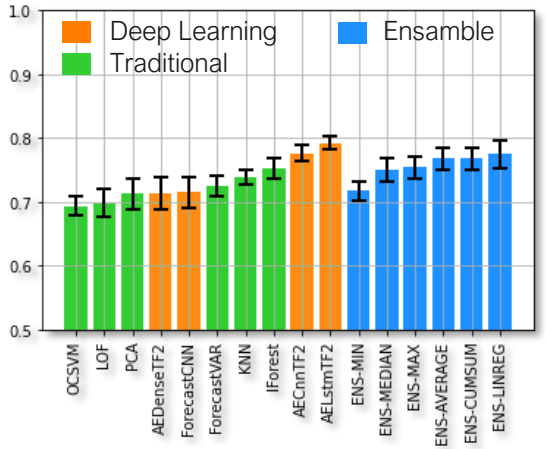
- To be threshold independent, the performances are estimated using the AUC of the ROC curve
- Different experiments for different user categories: **Batch** and **Shared**
- Selected 1 Hostgroup per category, 8 months of data analysed

- The AUC-ROC is measured on several, independent weeks.
  We show the **average** performances (together with the **std**).

"Batch" category



"Shared" category



- **IForest and PCA** perform as good as Deep methods in Batch category, **higher AUC-ROC** in general
- **Deep Learning** methods score slightly better in Shared category, **lower AUC-ROC** in general
- Ensemble methods **underperform individual ones** (need more investigation, correlated individual scores)

Summary

- Designed an Anomaly Detection System with Expert Feedback
- First quantitative evaluation of unsupervised algorithms looks promising for their adoption in the Anomaly Detection for the CERN Cloud use case
- Defined a procedure to annotate and collect time-series datasets

- Future work:
  - Improve the size and quality of the **annotated datasets** (including more info)
  - Explore different **hyper-parameters** and **metrics** (optimization)

More details about the project: indico event, ITTF slides, Gitlab Repository

Image Attribution

- AI Robot: by photo3idea_studio from Flaticon.com https://www.flaticon.com/free-icon/ai_1693746
- Server image: http://pngimg.com/download/25951
- Server Image: https://www.flaticon.com/free-icon/data-server_2911789
- Datasets: http://clipart-library.com/database-icon.html
- Expert: https://www.freepik.com , http://www.flaticon.com
- Contract icon: https://www.flaticon.com/free-icon/contract_2942912
- Red fish: https://www.flaticon.com/free-icon/fish_300597
- Blue fish: https://www.flaticon.com/free-icon/fish_300407
- Broken Robot: by freepik from Flaticon.com
  https://www.flaticon.com/free-icon/robot_3398611
  https://www.flaticon.com/free-icon/robot_3398643
- Database: https://www.flaticon.com/free-icon/server_689360