Anomaly Detection Pipeline with Expert Feedback

Matteo Paltenghi
Domenico Giordano

Agenda

About me

1. Anomaly detection problem for CERN Compute Infrastructure
2. Algorithms for Anomaly Detection
3. Importance of Grafana annotation for feedback loops:
   - Anomaly detection pipeline
   - Extension of Grafana annotation (Grafana on steroids)
4. Future Work

About Me

Name: **Matteo Paltenghi**

Background: **MSc Double Degree in Data Science**

TU Berlin - Politecnico di Milano

Role: **TECH Student**

Start date: 1st March 2020

Project Title: **Data Analytics of CERN Cloud monitoring data**

The project activity consists in

- evaluate different Data Analytics algorithms (Deep Learning vs Machine Learning)
- identify the best algorithms for the CERN Cloud
- integrate the developed approaches in the MONIT and alarming infrastructure

My Master Thesis

- Close collaboration with my **Master Thesis supervisors:** experts in the Anomaly Detection field (e.g. ICML* publications) with **industrial experience.**

Prof. Giacomo Boracchi
Politecnico di Milano

Deep Learning expert in Anomaly Detection
Collaboration with **STMicroelectronics**

PhD Student Lukas Ruff
Technische Universität Berlin

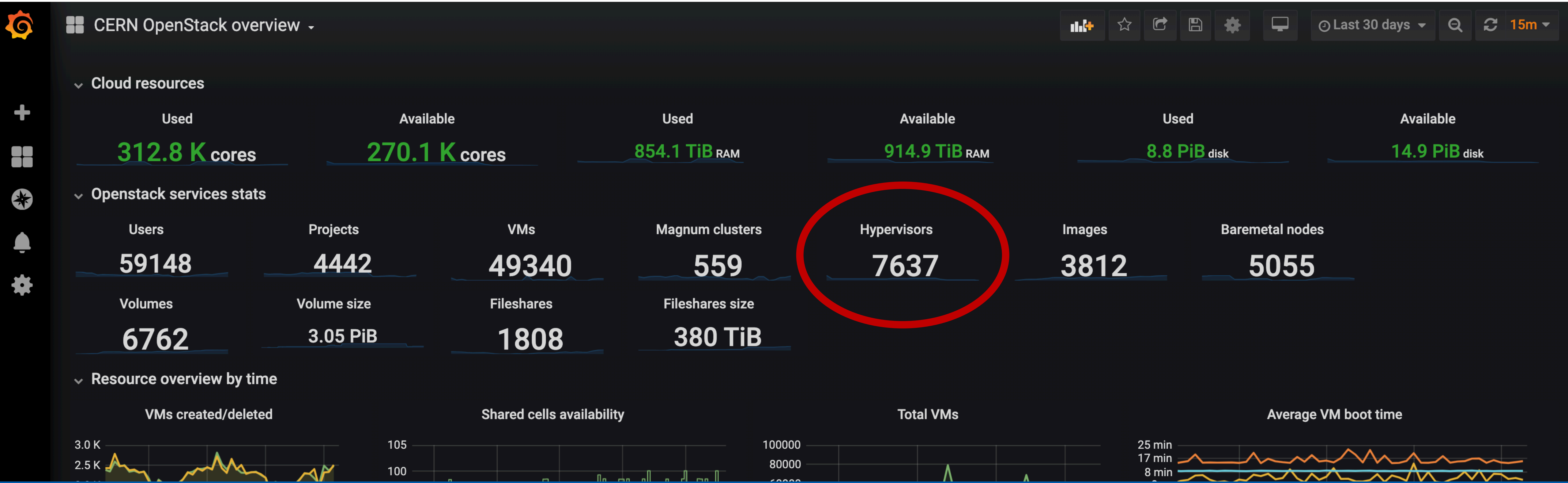
Machine Learning Department
PhD research area:
Deep Anomaly Detection

*ICML, International Conference on Machine Learning, top conference in the ML Field

1. Anomaly Detection Task for CERN Compute Infrastructure

Analysis Target

- Focus on hypervisors organized on 80 Openstack Cells (i.e. Hostgroups)



Our Big Data Scenario

- Each server produces **performance metrics (time series data)** thanks to a **system statistics collection daemon** (i.e. Collectd)
- Size of the problem
  More than 7k bare-metal servers and 38k service VMs
- High dimensionally:
  Up to 170 time series per server

Challenges of the Anomaly Detection task

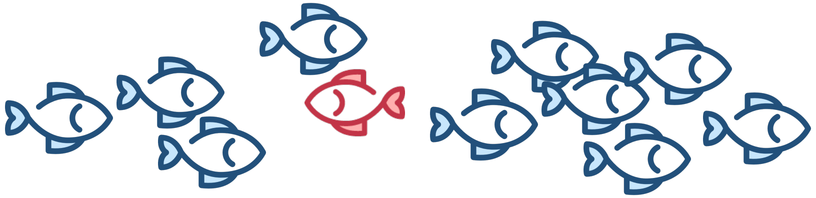
- Cloud Openstack Monitoring mainly **used** for
  - Grafana alarming with thresholds: prone to false positives/negatives
  - Post-mortem analysis with tedious manual inspection
- Overwhelming manual data exploration
- Missing underlying correlation between timeseries

Anomaly Detection Problem Formulations

1. **Swarm Series Outlier in one Openstack cell**

Each cell represents a "*swarm*" of machines.

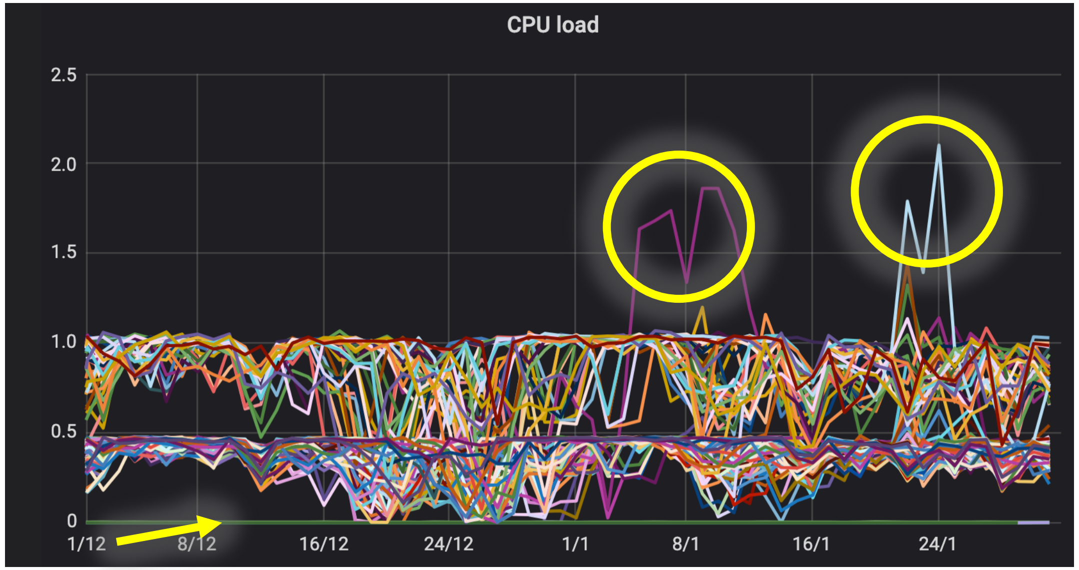
We want to spot if a machine is **deviating from the other machines' behaviour** in the same cell.



2. **Change detection**

We monitor each and every machine to detect if that machine is having a **strange behaviour with respect to its own past** (no peers confrontation).

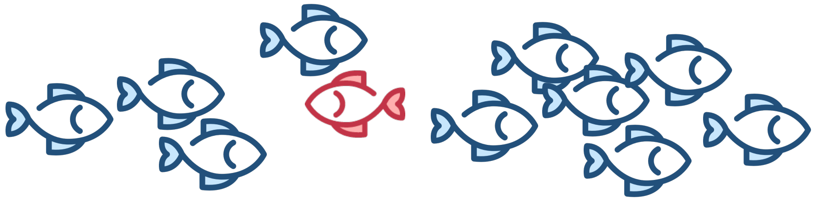
Openstack Cell: gva_project_014



Each time series is representing the CPU load of one server

Homogeneity

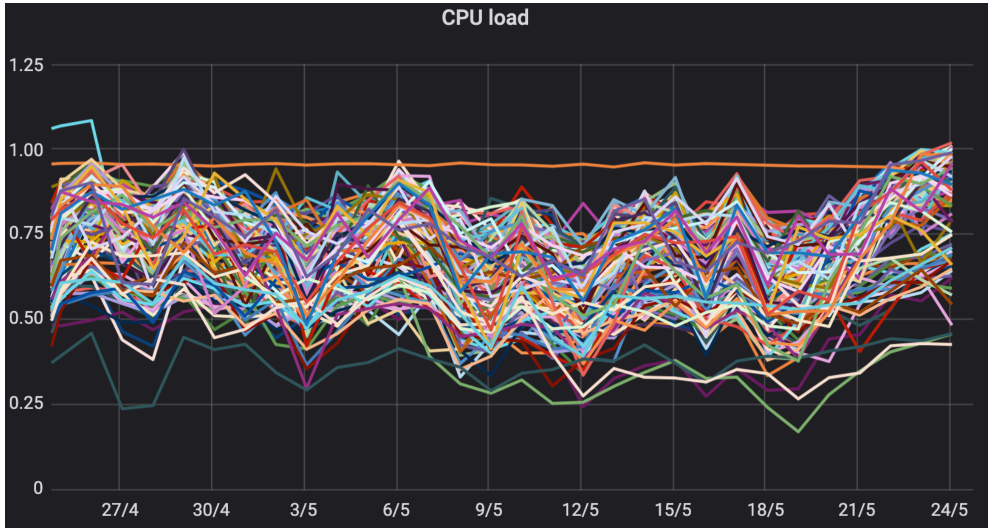
- Openstack Cell = *aggregate of servers all having:*
  - *same HW (by procurement/acquisition)*
  - *same HW setup in the Data Centre (DC)*
  - *same SW configuration (via Puppet hostgroups)*
  - *same target usage (batch/services)*



Two Main User categories

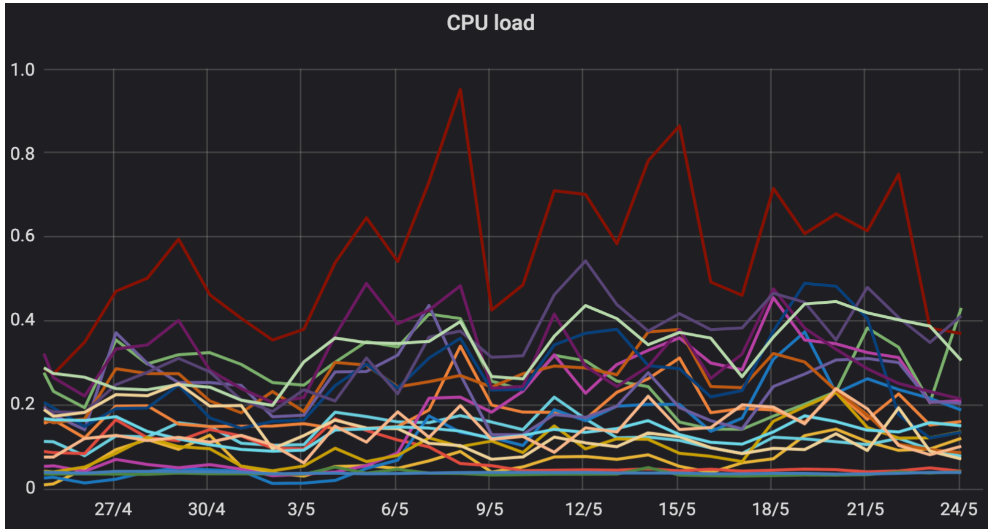
Starting scenario due to its regular pattern

Batch Cell
machines used for batch jobs, similar workloads



More challenging scenario to explore next

Shared (by services) Cell
Each machine is used by a CERN user/service, independent jobs



Each time series shows only CPU load of one server

2. Algorithms for Anomaly Detection

Overview Methods Anomaly Detection

- We will tackle the problem with two approaches:

Principle	**Traditional AD Methods**	**Deep Learning Methods**
Prediction Based	AR, MA, ARMA, ARIMA	RNN family (i.e. LSTM or GRU) CNN based
Reconstruction Based	PCA, Robust PCA, Kernel PCA	Auto Encoder Variational Auto Encoder Generative Adversarial Network
Distance/Cluster/Ensemble	Local Outlier Factor OCSVM Isolation Forest	

Integrated & Tested on a simple case

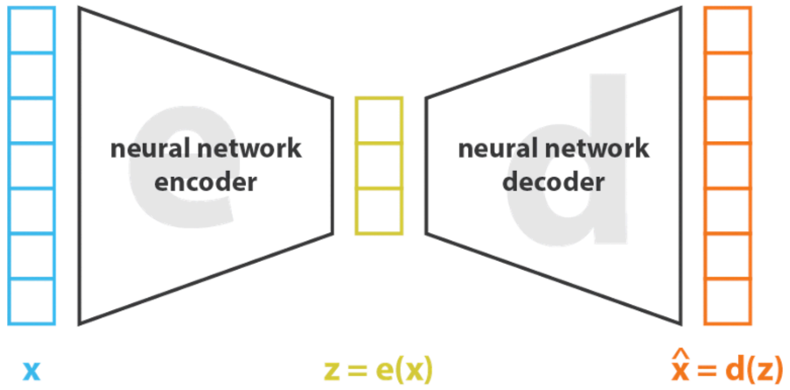


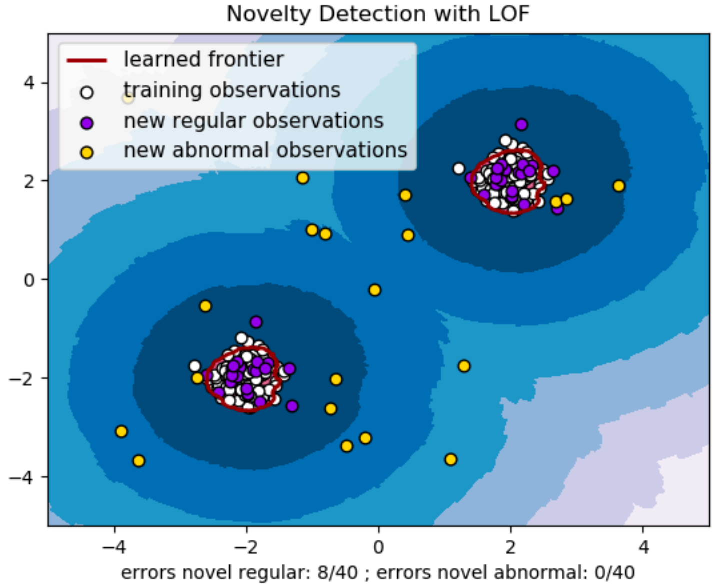
Image Source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

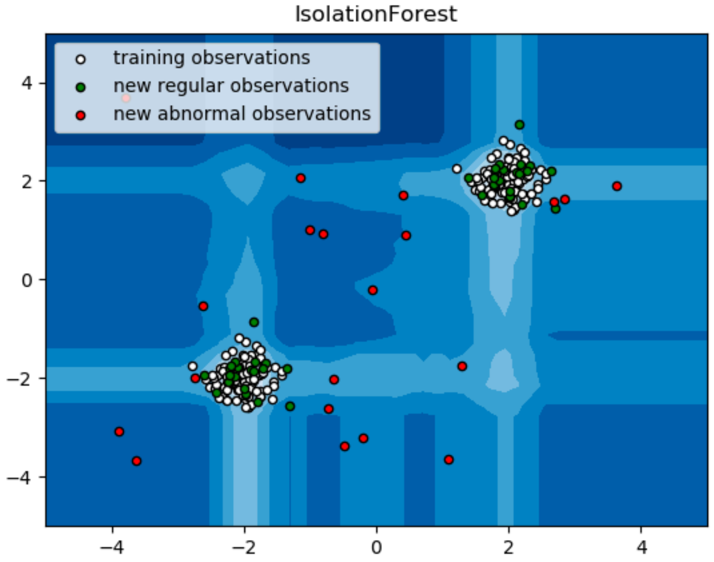
Traditional Methods

Local Outlier Factor (2000):
Consider the local density change with respect to the neighbours

Isolation Forest (2008):
Based on ensembles of decision tree

One-Class SVM (2000):
Based on Support Vector Machines





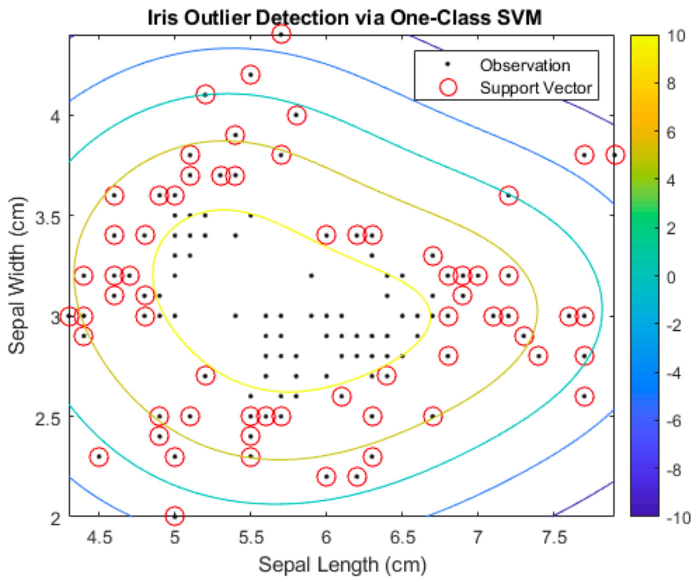


Image Sources: sklearn and MathWorks official documentation

Autoencoder

- An Autoencoder learns how to reconstruct the input from a lower representation

If **trained** on majority of **normal data**:

- it will be able to reconstruct them (normal data) really well,
- but with abnormal data it will make mistakes (high reconstruction error).

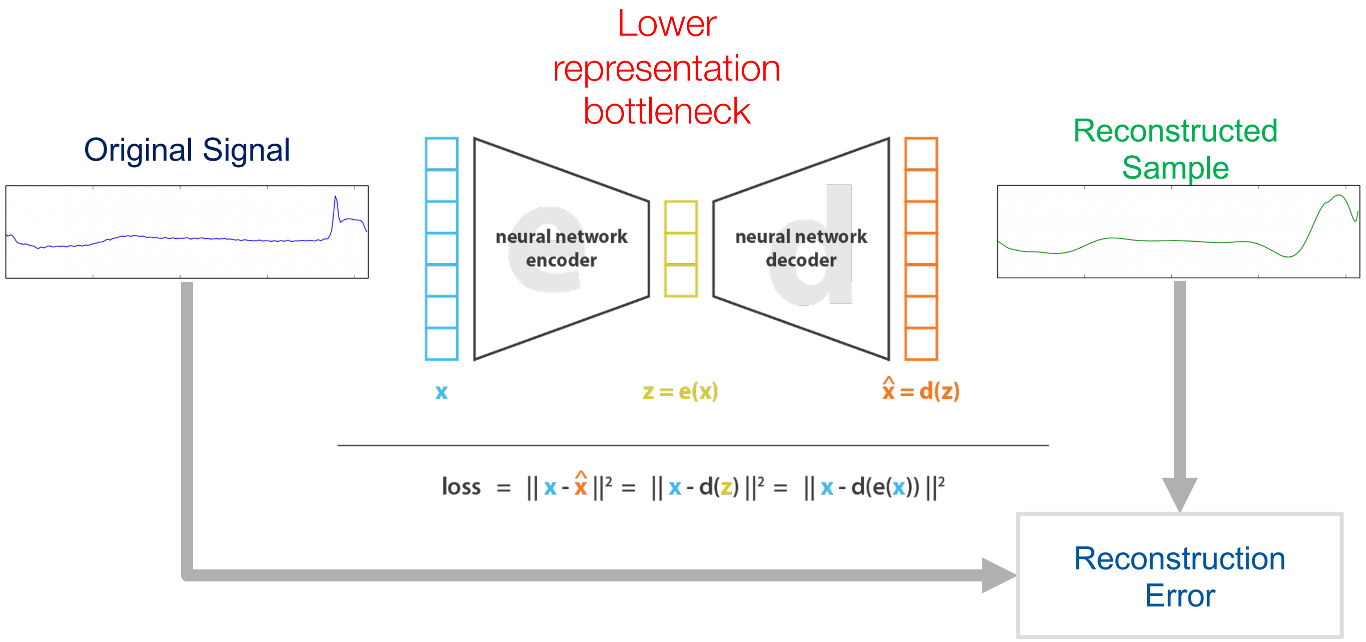


Image Source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

PyOD: A Python Toolbox for Scalable Outlier Detection

- Implement lots of Algorithms
- Well maintained codebase & Doc
- Potential contribution on AD on timeseries

Welcome to PyOD documentation!

Deployment & Documentation & Stats

pypi v0.7.8.2 | docs passing | launch binder | stars 3.1k | forks 627 | downloads 842k | downloads/month 61k

Build Status & Coverage & Maintainability & License

build passing | build passing | PASSED | coverage 96% | maintainability B | license BSD-2-Clause

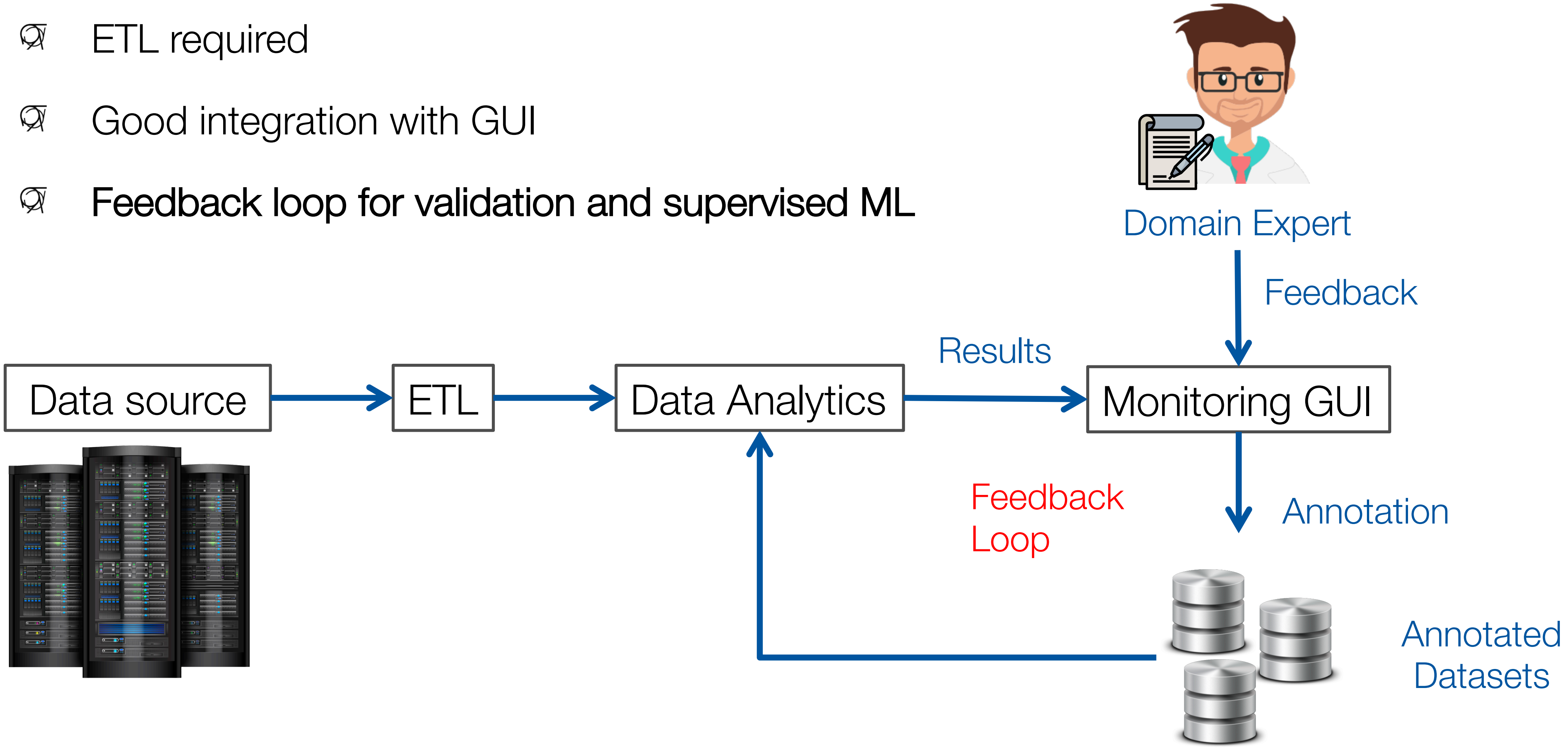
Source: https://github.com/yzhao062/pyod

Type	Abbr	Algorithm
Linear Model	PCA	Principal Component Analysis (the sum of weighted proj
Linear Model	MCD	Minimum Covariance Determinant (use the mahalanobis
Linear Model	OCSVM	One-Class Support Vector Machines
Linear Model	LMDD	Deviation-based Outlier Detection (LMDD)
Proximity-Based	LOF	Local Outlier Factor
Proximity-Based	COF	Connectivity-Based Outlier Factor
Proximity-Based	CBLOF	Clustering-Based Local Outlier Factor
Proximity-Based	LOCI	LOCI: Fast outlier detection using the local correlation in
Proximity-Based	HBOS	Histogram-based Outlier Score
Proximity-Based	kNN	k Nearest Neighbors (use the distance to the kth neares
Proximity-Based	AvgKNN	Average kNN (use the average distance to k nearest nei
Proximity-Based	MedKNN	Median kNN (use the median distance to k nearest neig
Proximity-Based	SOD	Subspace Outlier Detection
Probabilistic	ABOD	Angle-Based Outlier Detection
Probabilistic	FastABOD	Fast Angle-Based Outlier Detection using approximation
Probabilistic	SOS	Stochastic Outlier Selection
Outlier Ensembles	IForest	Isolation Forest
Outlier Ensembles		Feature Bagging
Outlier Ensembles	LSCP	LSCP: Locally Selective Combination of Parallel Outlier E
Outlier Ensembles	XGBOD	Extreme Boosting Based Outlier Detection **(Supervised)**
Outlier Ensembles	LODA	Lightweight On-line Detector of Anomalies
Neural Networks	AutoEncoder	Fully connected AutoEncoder (use reconstruction error a
Neural Networks	VAE	Variational AutoEncoder (use reconstruction error as the
Neural Networks	SO_GAAL	Single-Objective Generative Adversarial Active Learning
Neural Networks	MO_GAAL	Multiple-Objective Generative Adversarial Active Learni

3. Anomaly Detection Pipeline

Data Analytics Pipeline - Process description

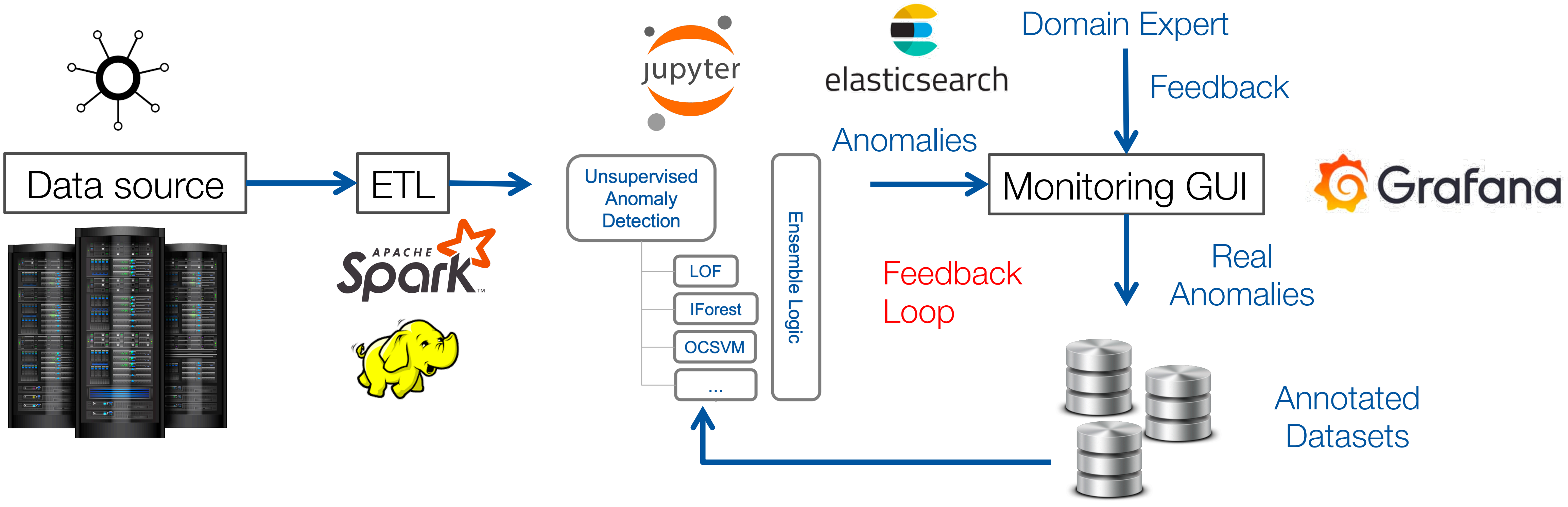
- ETL required
- Good integration with GUI
- Feedback loop for validation and supervised ML



Anomaly Detection Pipeline - Process Description

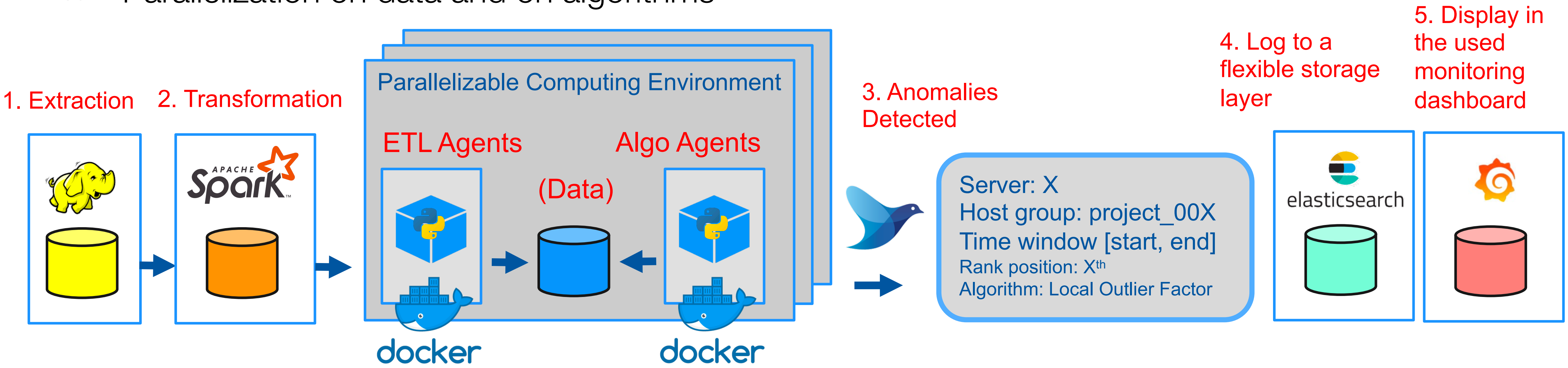
- Pipeline fully based on

  CERN central monitoring infrastructure

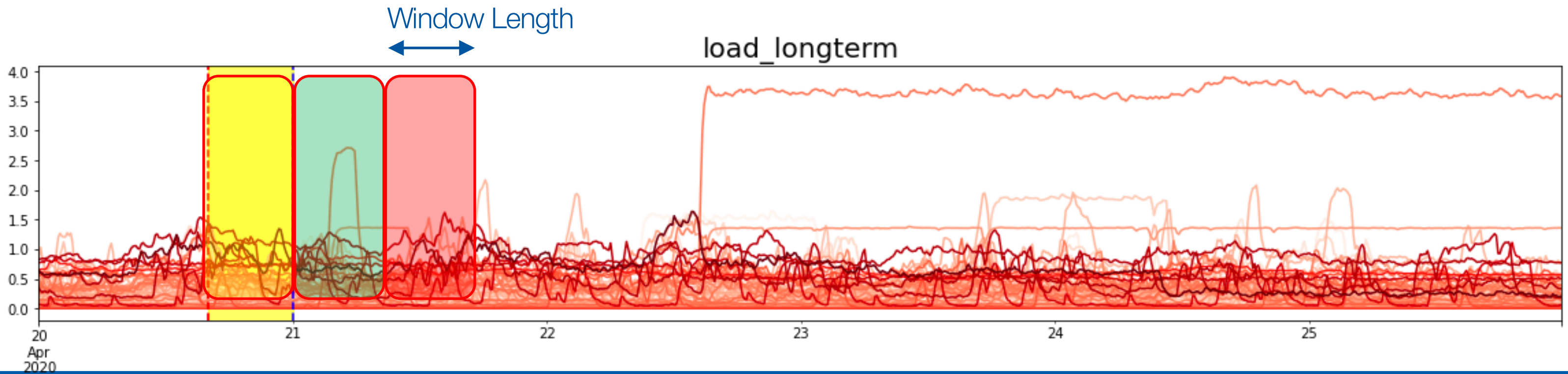


Parallelization and Temporal Dimension

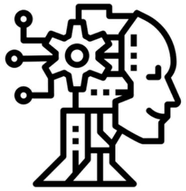
- Parallelization on data and on algorithms



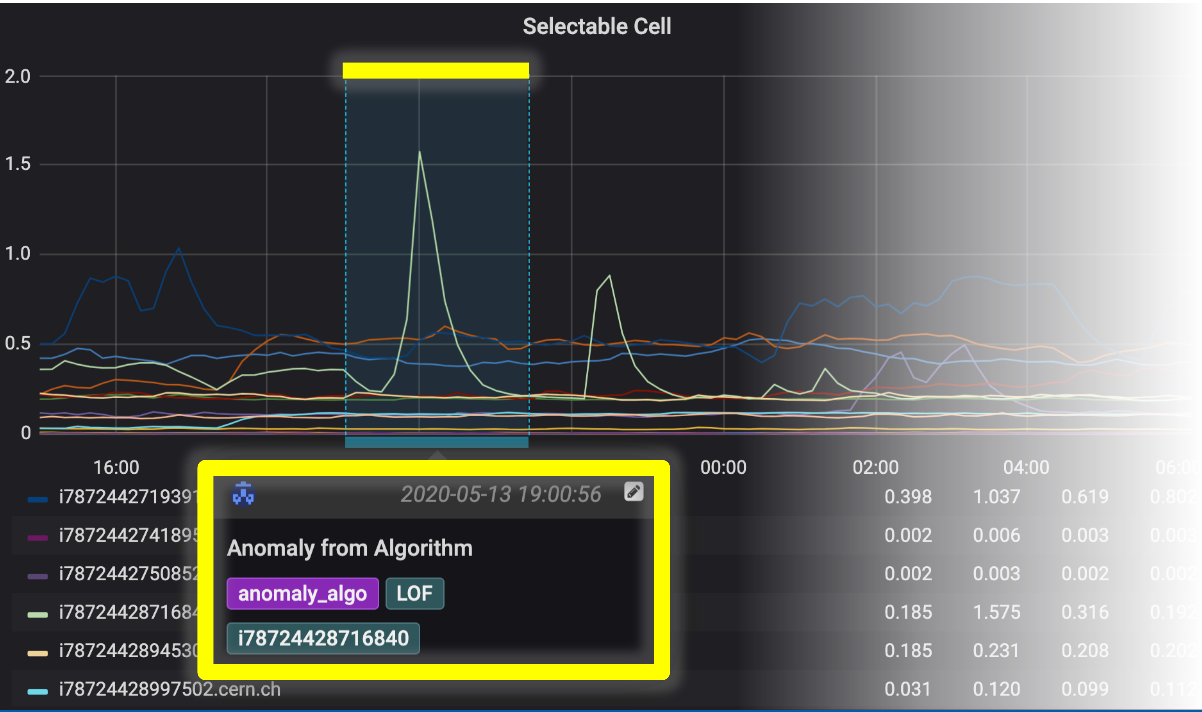
- Analysis of data in windows
- Configurable parameters:
  - Window length
  - Slide step



Grafana Annotation – How we use it (1)



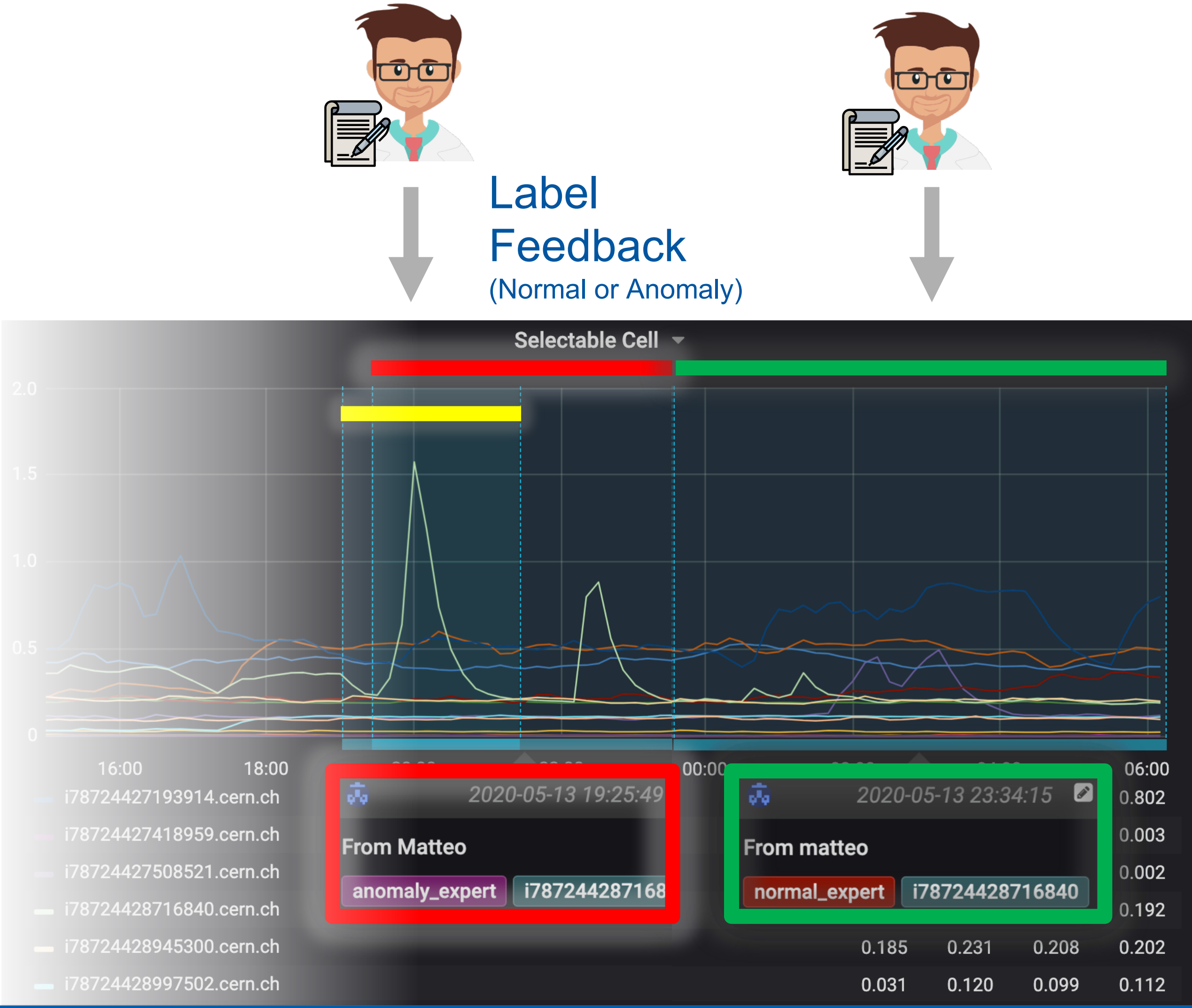
Algorithm Prediction
(Anomaly)



- Show anomalies via Grafana annotations
  - Display data as time intervals
  - Also from ES data source

Grafana Annotation – How we use it (2)

- Let user to insert anomalies
- Benefits:
  - Grafana Annotation can be extracted with Rest API for the feedback loop
- Drawbacks
  - Manual insertion of tags (e.g. anomaly/normal and machine name)
  - Tedious and Error prone operations



Idea: Extension of Grafana Annotation functionality

Requirements:

- Add two buttons for our use case
- Automatically add the template variables of the current dashboard as tags of the annotation

Benefits:

- More automation => Less errors
- Faster one-click annotation => More annotations

New interface of our simple patch frontend

Extension of Grafana Annotation: Implemented!

- Just few line of client code on the web UI (one JavaScript function and couple of HTML buttons)
- Hosted on our Repo Gitlab
  - Differences with respect to the original one can be verified (a.k.a. **you can trust it**)
- Usable with simple **Chrome/Firefox plugin** to override Grafana JavaScript locally
- Discussion upstream with Grafana Community to get a more general functionality https://github.com/grafana/grafana/issues/24674

grafana_patch/original_file.js → grafana_patch/modified_file.js

```
+              <button type=submit class="btn btn-primary" ng-click=ctrl.save_anomaly_with_vars()>Anomaly</button> \
+              <button type=submit class="btn btn-primary" ng-click=ctrl.save_normal_with_vars()>Normal</button> \
               \n\t\t\t\t<button type=submit class="btn btn-primary" ng-click=ctrl.save()>Save</button>\n\t\t\t\t<button ng-if=ctrl.event.id type=submit click=ctrl.delete()>Delete</button>\n\t\t\t\t<a class=btn-text ng-click=ctrl.close();>Cancel</a>\n\t\t\t</div>\n\t\t</div>\n\t</form>\n</div>\n'
            );
          },
@@ -112017,6 +112019,110 @@
                }
              },
            },
+           {
+             key: "save_anomaly_with_vars",
+             value: function () {
+               var e = this;
+               if (this.form.$valid) {
+                 var t = u.a.cloneDeep(this.event);
+                 t.tags.push("anomaly_expert");
+                 var nrMaxTemplateVars = this.panelCtrl.dashboard.templating
+                   .list.length;
+                 for (var i = 0; i < nrMaxTemplateVars; i++) {
+                   var currentTmpVar = this.panelCtrl.dashboard.templating
+                     .list[i];
+                   var nrMaxValues = currentTmpVar.options.length;
+                   for (var j = 0; j < nrMaxValues; j++) {
+                     var currentValueForTmpVal = currentTmpVar.options[j];
+                     if (currentValueForTmpVal.selected === true) {
+                       var nameSelectedVal = currentValueForTmpVal.value;
+                       if (nameSelectedVal !== "$__all") {
+                         console.log(
+                           "template var name: " + nameSelectedVal.toString()
+                         );
+                         t.tags.push(nameSelectedVal);
```

Achievements

- Usage and test of 4 algorithms (Isolation Forest, Local Outlier Factor, One-Class SVM, Autoencoder)
  - Adaptation for timeseries input
  - Usage of O(10 plugins) as input (not only load)
- Anomalies visible in our index on **ElasticSearch** (MONIT-Kibana)
- Anomalies visible in our **Grafana dashboard** (Anomaly Detection Results)
- Click on one host – Anomalies are displayed as annotation intervals
- Example of usage of the **new patch** extended Grafana feature

Data Extracted from Grafana Annotation API

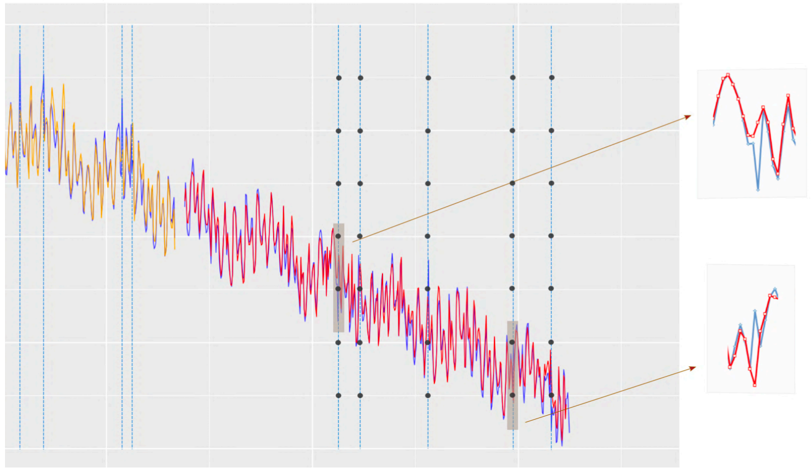
- Annotations can be easily extracted to create a dataset.

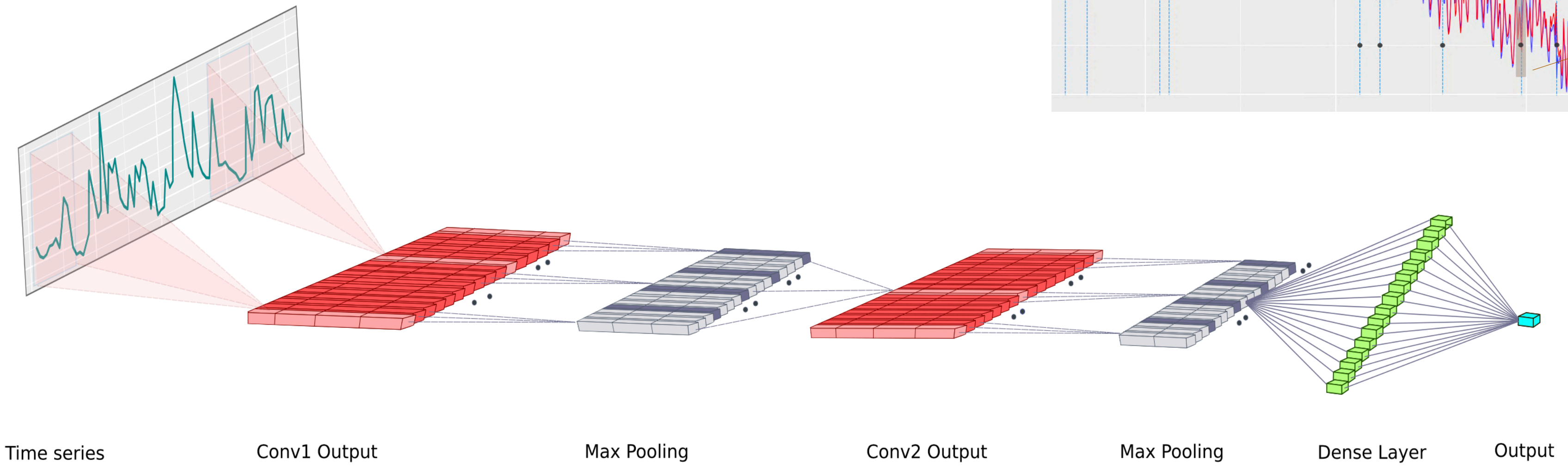
	hostname	hostgroup	ts_start_milli	ts_end_milli	is_anomalous	author	description
17	p06253944a91141	cloud_compute/level2/batch/gva_project_014	1586283511028	1586648577267	0	matteo.paltenghi@cern.ch	normal utilisation
16	p06253944a21006	cloud_compute/level2/batch/gva_project_014	1587517434915	1587619653462	1	matteo.paltenghi@cern.ch	half utilisation
15	p06253944a91141	cloud_compute/level2/batch/gva_project_014	1588167252820	1588641838930	1	matteo.paltenghi@cern.ch	half utilisation load
12	p06253944e77642	cloud_compute/level2/batch/gva_project_014	1588582293607	1588859249303	1	matteo.paltenghi@cern.ch	drop in cpu user
14	p06253944e77642	cloud_compute/level2/batch/gva_project_014	1588665117879	1588689403111	1	matteo.paltenghi@cern.ch	increased process fork activity
13	p06253944y87408	cloud_compute/level2/batch/gva_project_014	1588685141358	1588805856574	0	matteo.paltenghi@cern.ch	standard load utilisation
11	p06253944n17852	cloud_compute/level2/batch/gva_project_014	1588996664496	1589152426064	0	matteo.paltenghi@cern.ch	normal load
10	p06253944e77642	cloud_compute/level2/batch/gva_project_014	1589341775118	1589403032866	0	matteo.paltenghi@cern.ch	normal memory ops
8	p06253944e77642	cloud_compute/level2/batch/gva_project_014	1589467843241	1589861141203	1	matteo.paltenghi@cern.ch	high load
5	p06253944n17852	cloud_compute/level2/batch/gva_project_014	1589484709862	1589884927144	1	matteo.paltenghi@cern.ch	increased memory operations

Future Work – Next Steps

Algorithms

- Measure the performance of the selected algos (see slide “Overview Methods Anomaly Detection”)
- Further extend the algorithm pool (if there is time)
  - Ensemble Learning
  - CNN/LSTM AutoEncoders
  - CNN/LSTM prediction based models

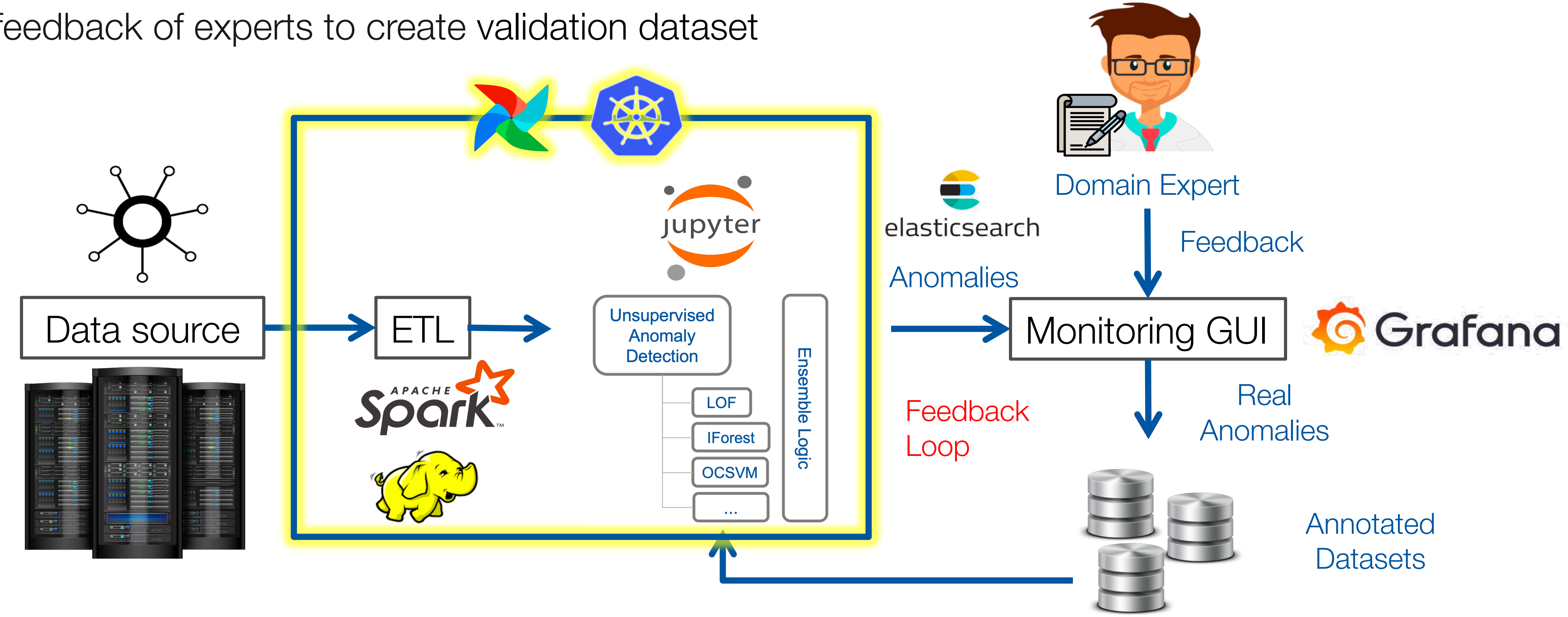




Source: DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series, Munir et al.

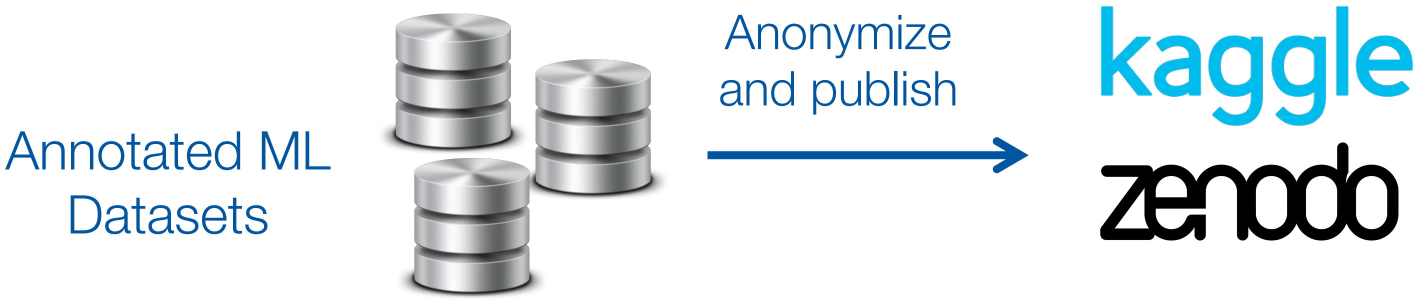
Finalize Anomaly Detection Pipelines

- Adding Orchestration and scheduling components (Airflow + k8s)
- Run extensively on data and produce candidate anomalies
- Collect feedback of experts to create validation dataset



Long Term View

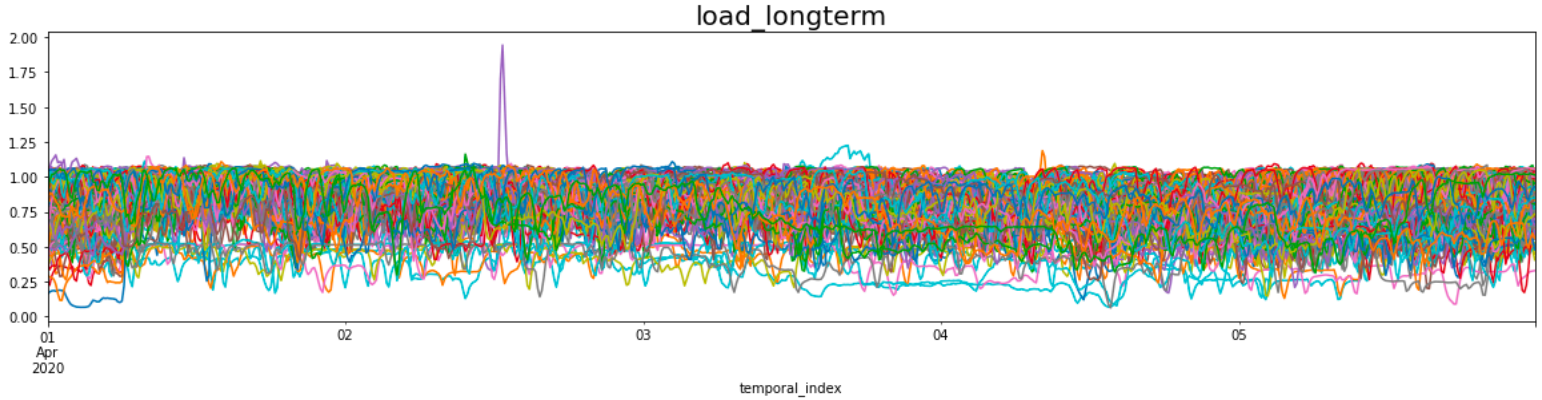
- Document this pipeline and share with our IT colleagues
- Possibly publishing dataset to help the community
- Benefit: get data science community onboard via Kaggle platform



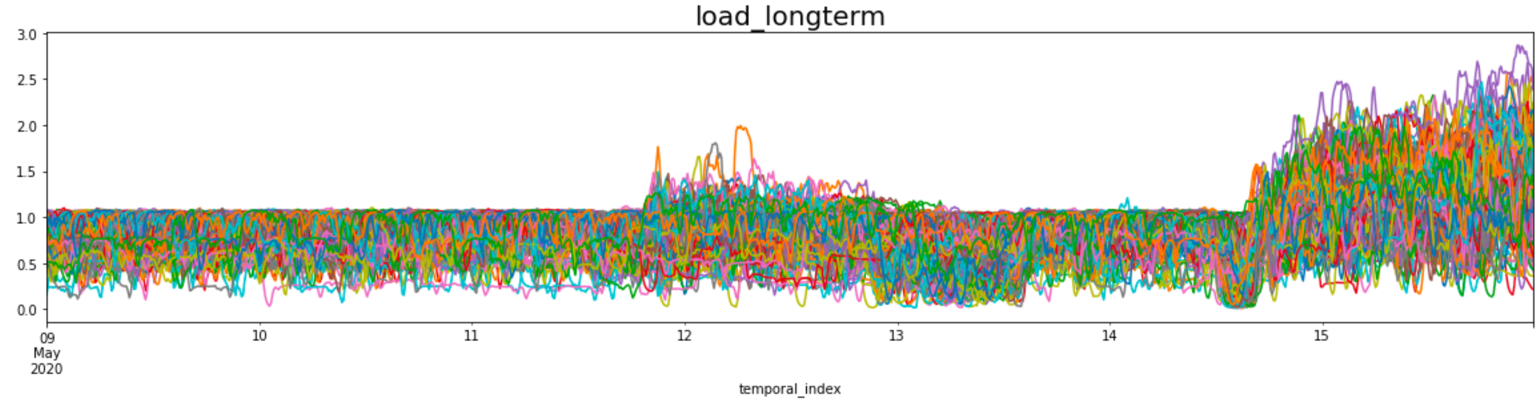
CERN

A Visual Example – Isolation Forest

Trained on April 2020 data

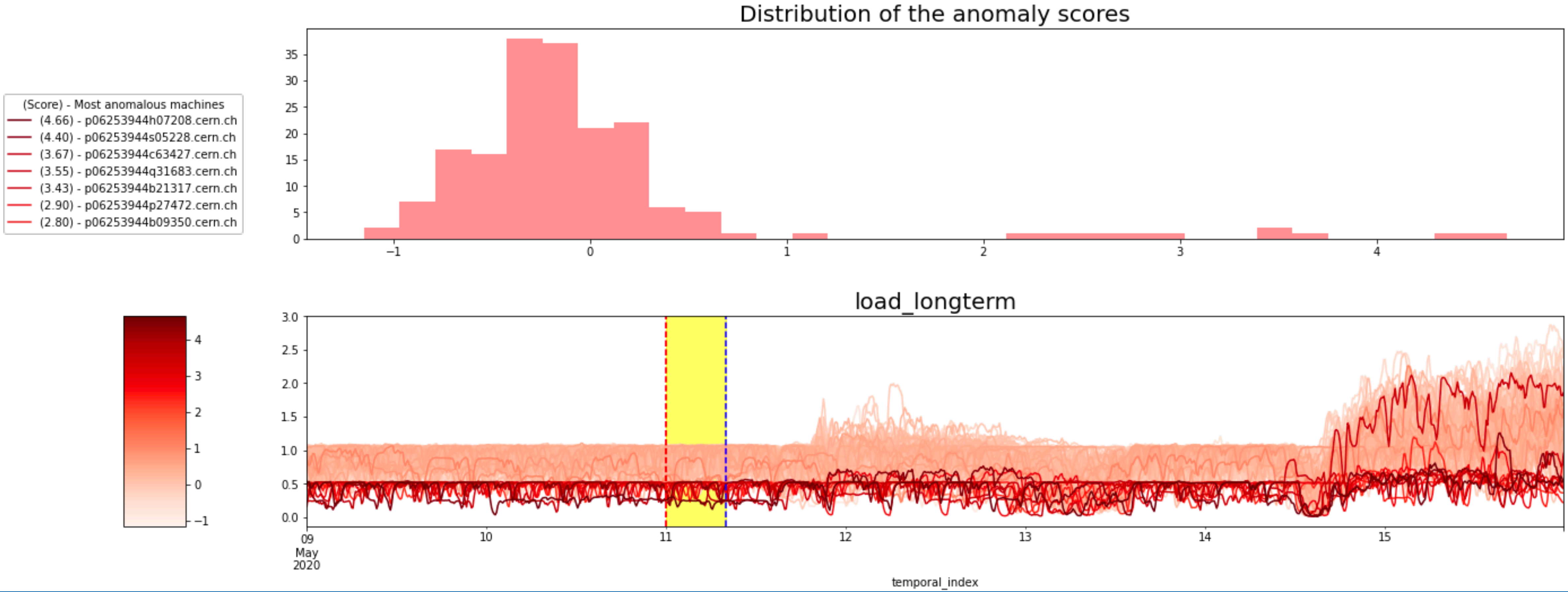


Tested on May 2020 data



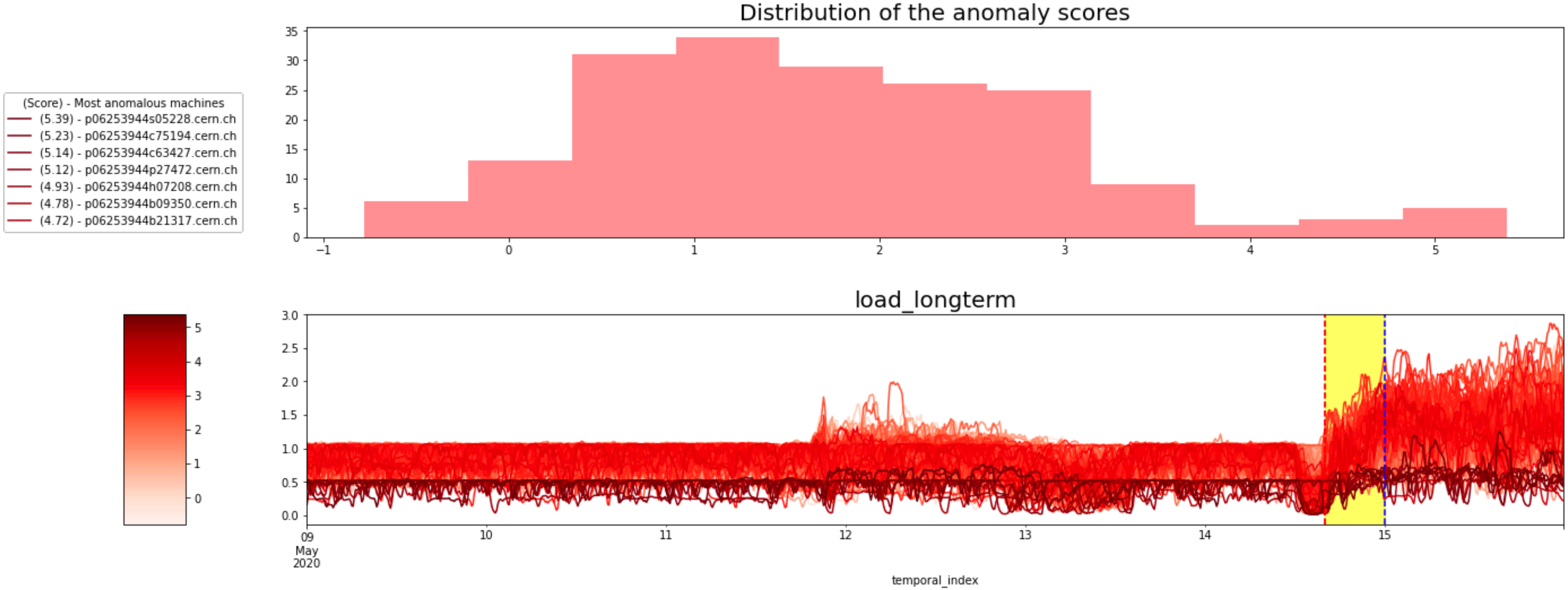
A Visual Example – Isolation Forest

- The algorithm is analysing only data in the yellow window



A Visual Example – Isolation Forest

- The algorithm is analysing only data in the yellow window



Extract Grafana Annotation

We can then extract those **precious annotated information** via a simple curl thanks to the Grafana Annotation API and create datasets for Machine Learning tasks.

```
----------------------------------------------------------------------------
GET /api/annotations?from=1506676478816&to=1507281278816&tags=tag1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
-----------------------------------
```

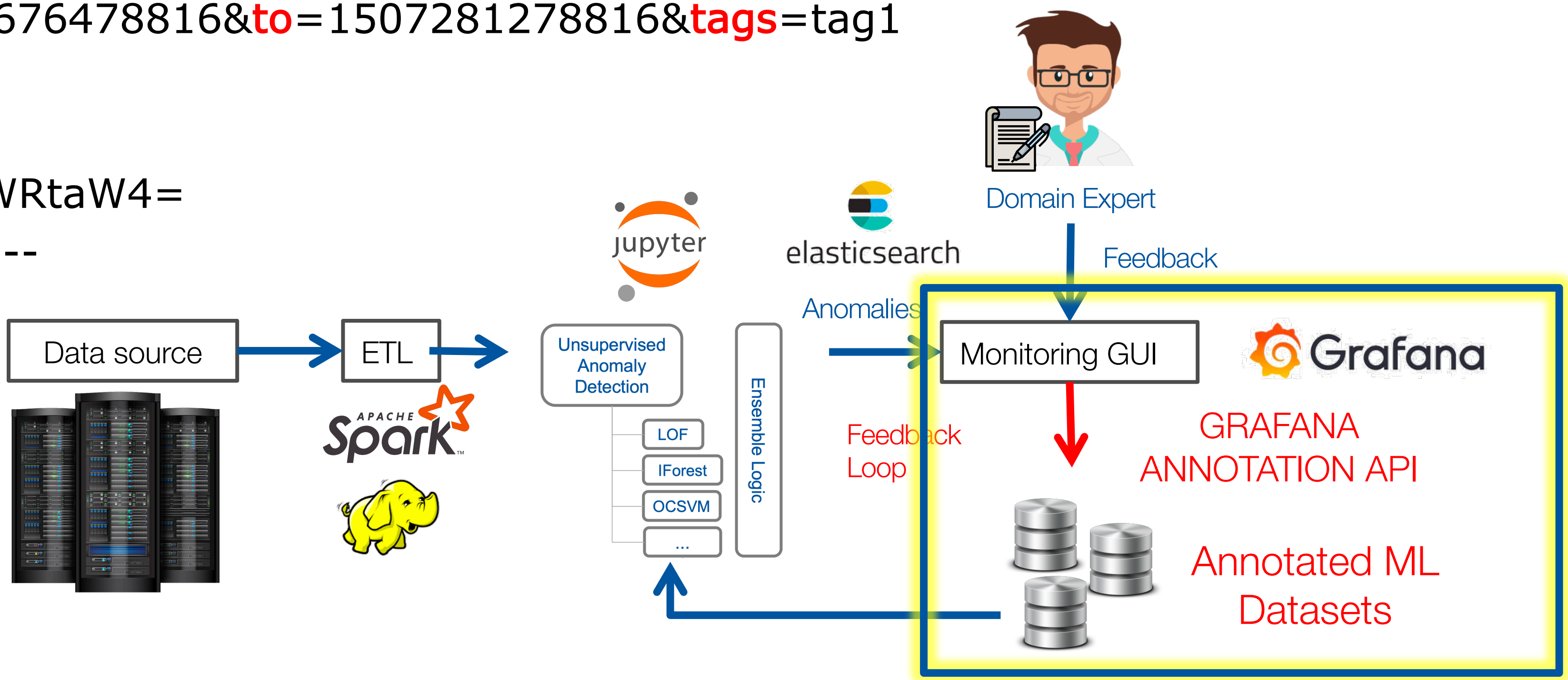


Image Attribution

- AI Robot: by photo3idea_studio from Flaticon.com
  https://www.flaticon.com/free-icon/ai_1693746
- Server image: http://pngimg.com/download/25951
- Datasets: http://clipart-library.com/database-icon.html
- Expert: https://openclipart.org/detail/262568/doctor-holding-clipboard-fixed-arm-and-whiter-coat
- 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
- Server Room: https://www.pexels.com/photo/hosting-server-server-room-servers-1570918/