# Machine learning for DQM and DC in CMS

Data Quality Monitoring and Data Certification

Mantas Stankevičius (Fermilab)
*on behalf of the CMS collaboration*

# Outline

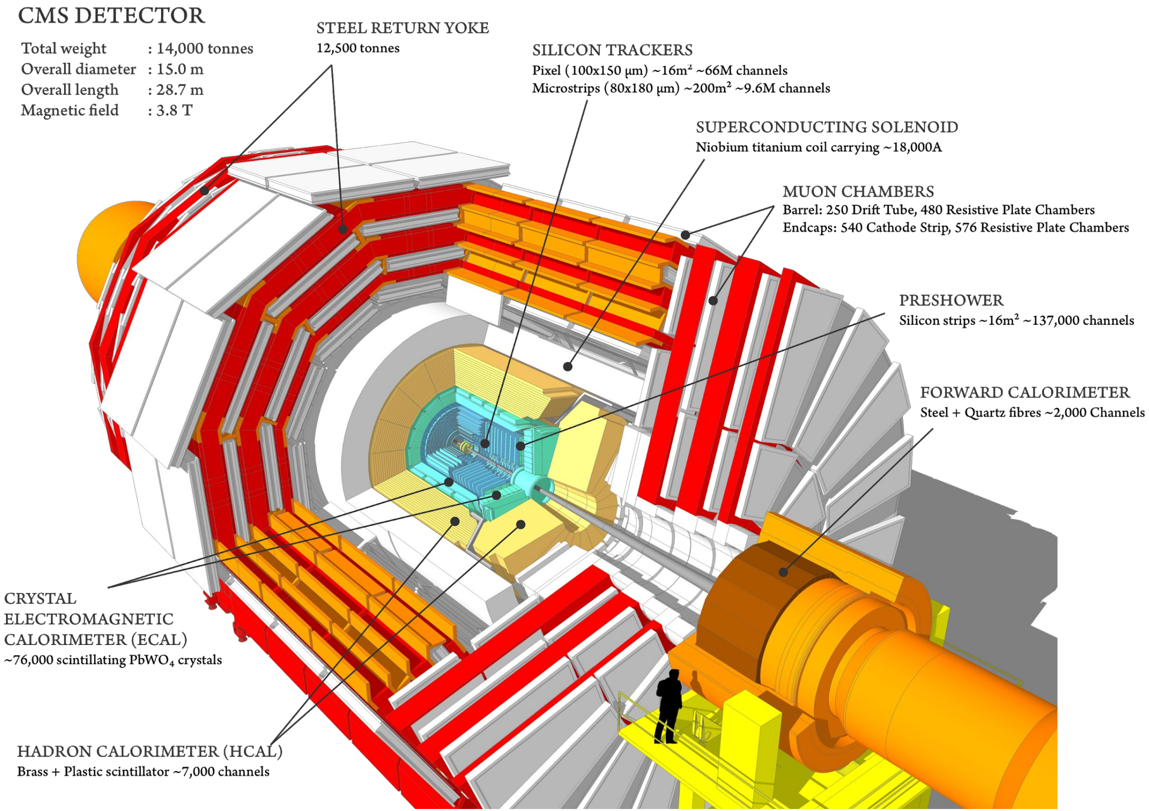
- Current DQM
  - Tools
  - Online: Detector monitoring
  - Offline: Data certification
  - Limitations
- ML-based DQM
  - How to fit ML into DQM operations
  - Applicability studies
    - Online
    - Offline

# CMS detector

Multi purpose detector at LHC

Approx 90 millions channels

Requires sophisticated DQM



# CMS DQM

Data Quality Monitoring

# Data Quality Monitoring [5]

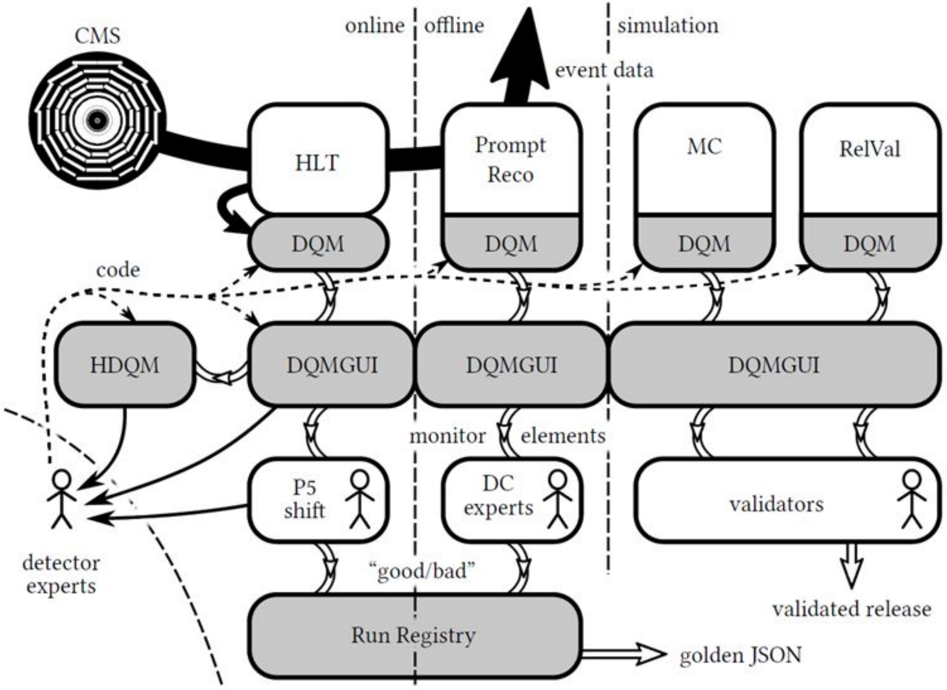
Collection of tools and processes to provide:

Monitoring. Detector and operation performance and malfunctions

Certification. Assess and record quality of data and software releases

Debugging. Provide detailed information in case of problems

Humans are a central part of DQM!



# Data Quality Monitoring: Online

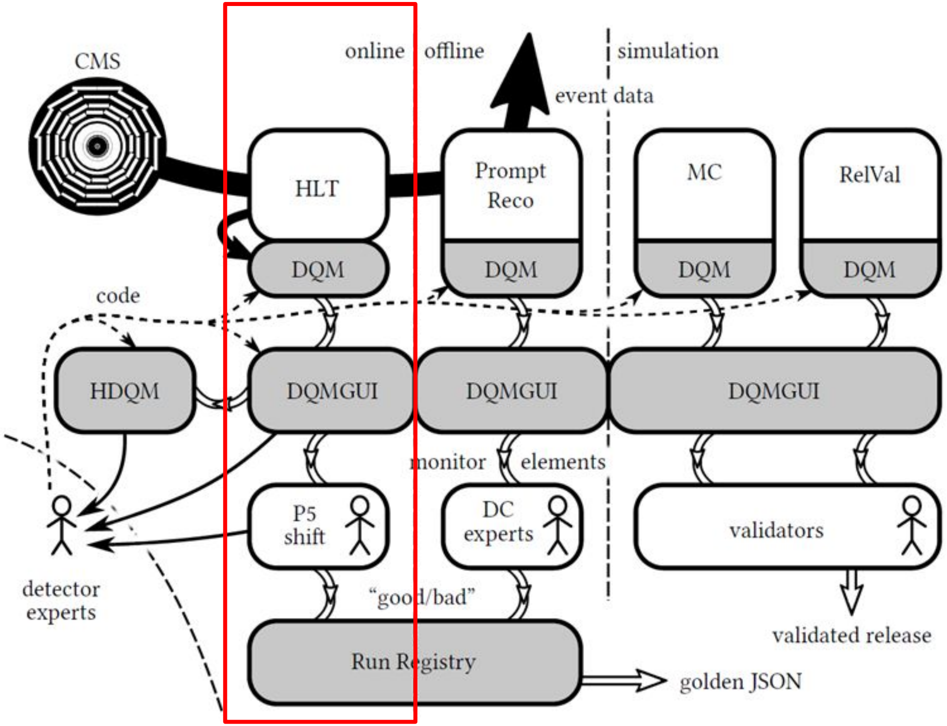
Collision data and detector status constantly flow from detector

Small subset is reconstructed and monitored real-time to give immediate feedback about detector status

Predefined Quality Tests are designed to identify known failures and raise alarm

Online DQM shifter at P5

- Inspect histograms to spot problems
- Certificate Run as GOOD if it has significant statistics and good hardware settings
- 3 shifts per day 8 hours each



# Data Quality Monitoring: Offline

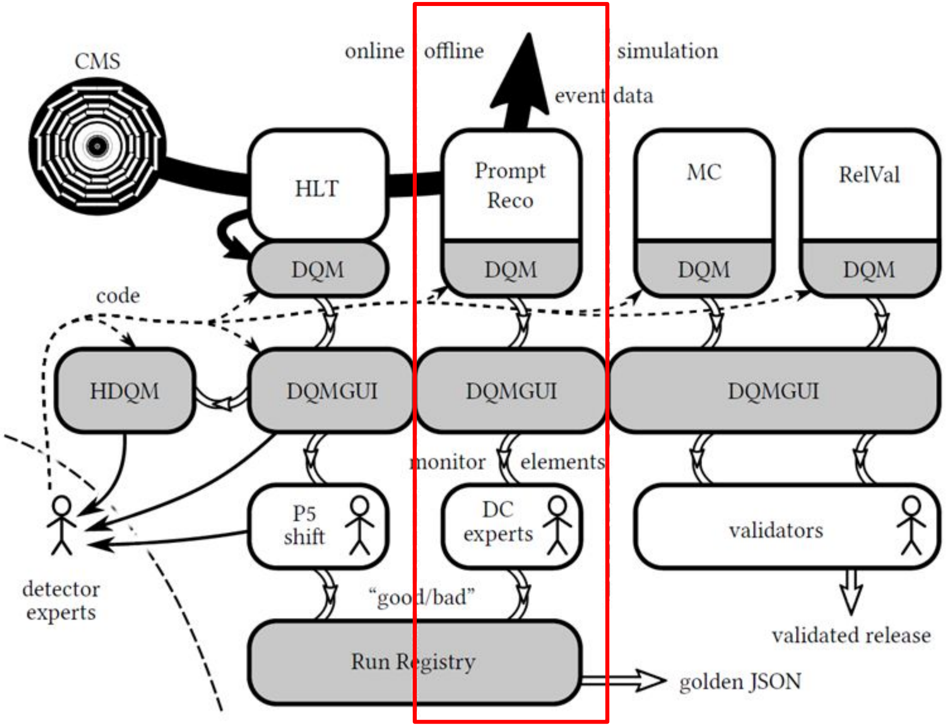
Data fully reconstructed a few days after being collected

Offline shifters and detector experts check dozens of distribution histograms to define goodness of data

Approx 30 Runs are certified per week

Certification is made on Run and Lumisection* levels

GoldenJSON is produced. List of only GOOD Runs and Lumisections




* Granularity of lumisection is a ~23sec of data-taking

# DQM GUI

Web service to collect and archive monitoring elements (ME)

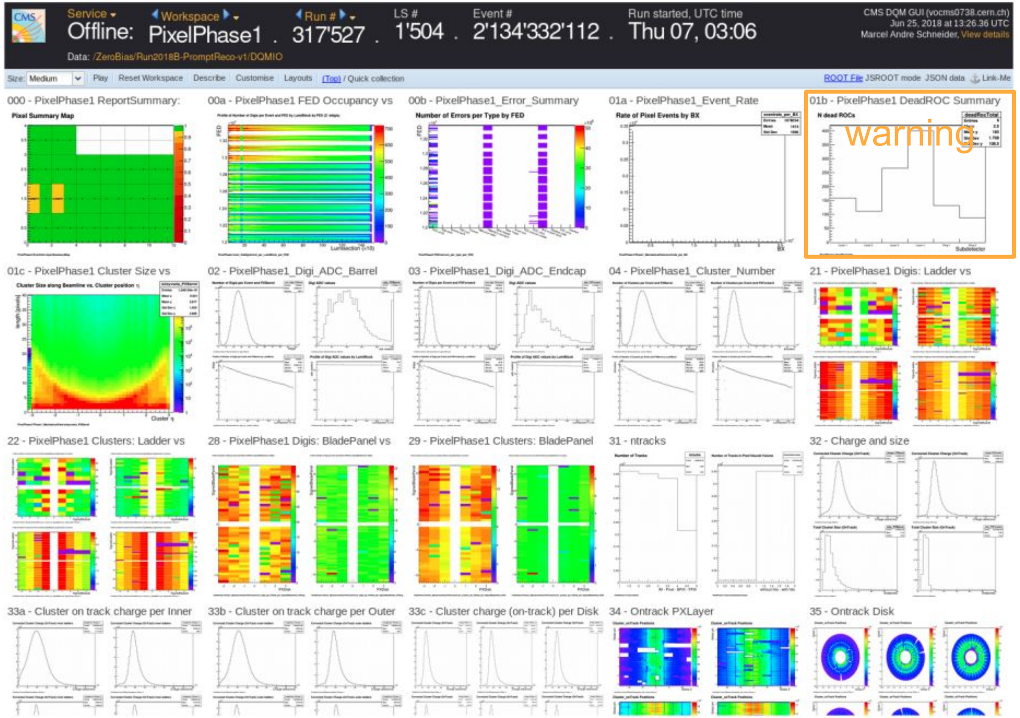
- ME = ROOT plot + Quality Test

Provides APIs for scripts

Web based interface to browse realtime and historical data

DQM GUI provides access to:

- Online: 22,000 runs, 650 GB
- Offline: 400,000 datasets, 4100 GB
- ~100k MEs per Run



# Run Registry

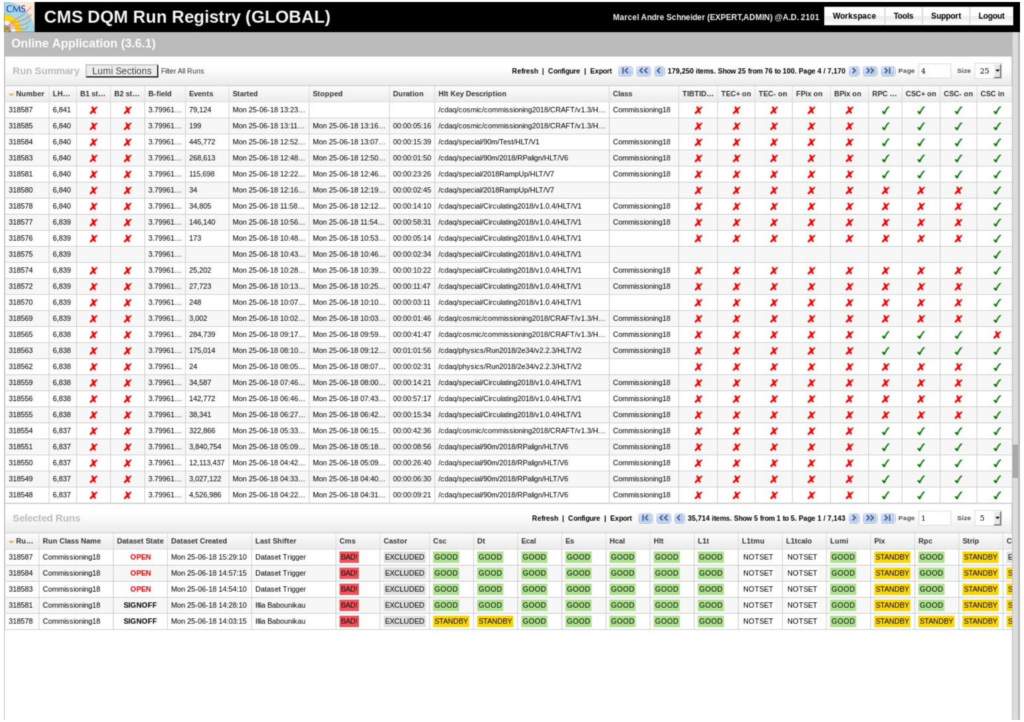
Automatically collects Run and Lumisection data

Web interface for experts to manually set quality flags on data (GOOD/BAD)

Provides APIs for scripts to produce final list of data ready for analysis (GoldenJSON)

Currently being re-developed for better usability and maintainability

Aim to accept input from ML services



# Limits of a Human-based DQM

- Problem spotting latency
- High manpower demand
  - 24/7 shifts + training
- Occasional involuntary human errors
  - There is a limit to the amount of quantities that a human can process in a finite time interval
  - Transient problem can be overlooked during visual comparison
  - Decision process depends on level of experience and understanding
- Changing running conditions
  - Reference samples change
  - Static thresholds do not scale
  - Maintenance of shifter instructions

# Real life example

Power supply issue on the Pixel detector

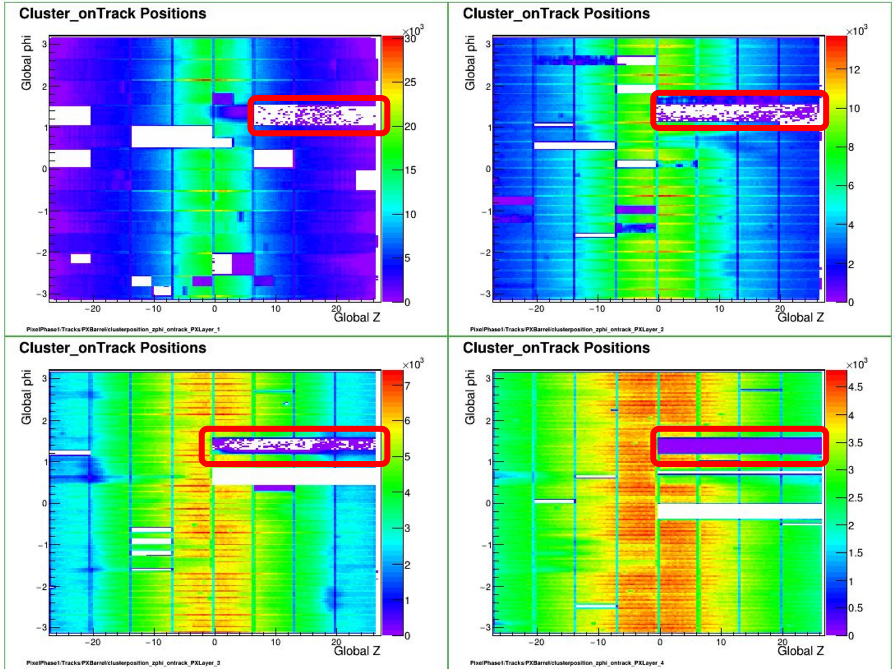
- Dead regions in 4 layers of the Pixel barrel
- Missing track seeds in that region
- Data certified as BAD (300 $pb^{-1}$)

Quality Tests based on # of dead Read-Out Chips (ROC) are not optimal

- OK - randomly distributed dead ROCs
- NOT OK - dead region in multiple layers

ML can be used to develop mode intelligent tests checking relative position of dead ROCs

Layer 1 Layer 2



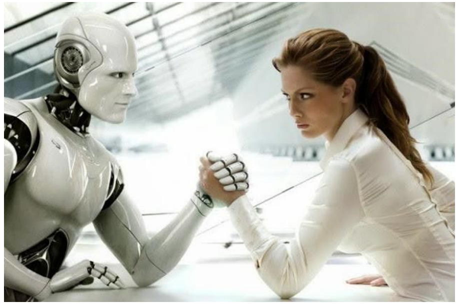

Layer 3 Layer 4

# Outline

- CMS detector
- Current DQM
    - Tools
    - Online: Detector monitoring
    - Offline: Data certification
    - Limitations
- ML-based DQM
    - How to fit ML into DQM operations
    - Applicability studies
        - Online
        - Offline

# Towards ML-based DQM

From rules to (un)supervised models



# How to fit ML in DQM operations?

Reduce manual labor by doing tedious work faster

- Tons of data (histograms) to compare
- Computer does not get tired

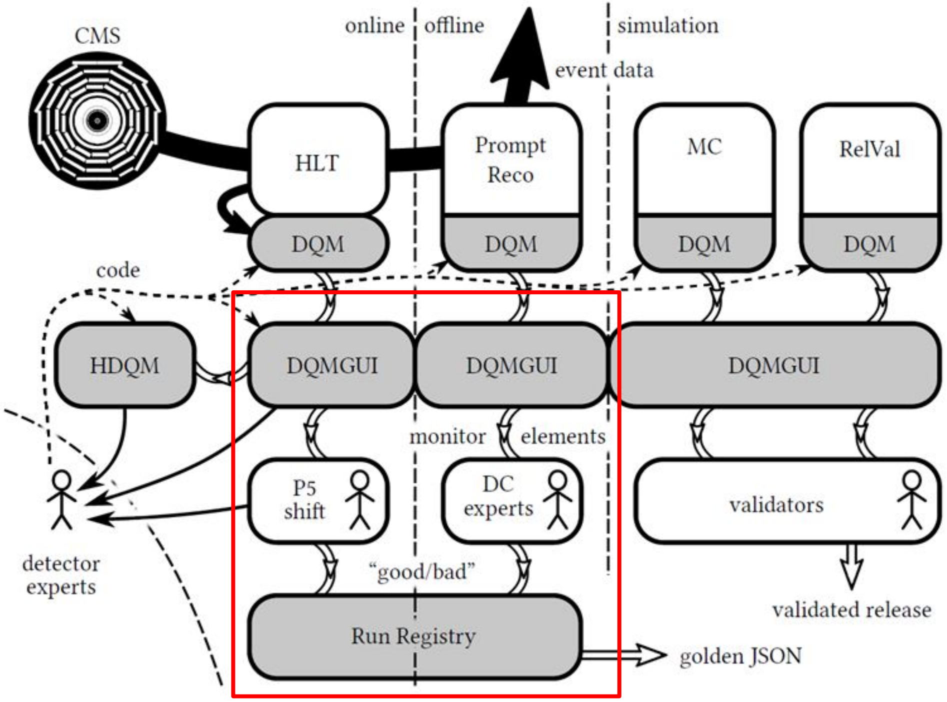
Minimize human errors and optimize human input

Detect anomalies with lower latency

Improve certification quality on lumisection level

Dynamically adapt to conditions change

Provide report of the classification results



# Learning techniques

**Supervised**

All data is labeled

Methods:

- Classification
- Regression

**Semi-supervised**

Some data is labeled

Combination of methods

Expensive to label data

**Unsupervised**

All data is unlabeled

Methods:

- Clustering
- Association

# Outline

- CMS detector
- Current DQM
  - Tools
  - Online: Detector monitoring
  - Offline: Data certification
  - Limitations
- ML-based DQM
  - How to fit ML into DQM operations
  - Applicability studies
    - Online
    - Offline

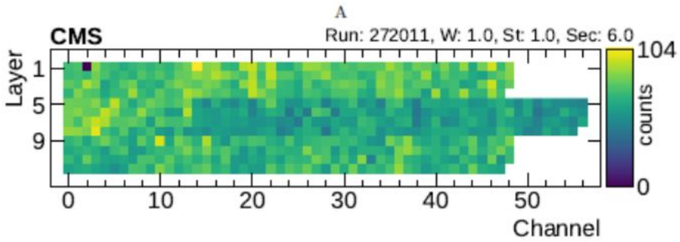
# Online: detector monitoring

# Occupancy plots

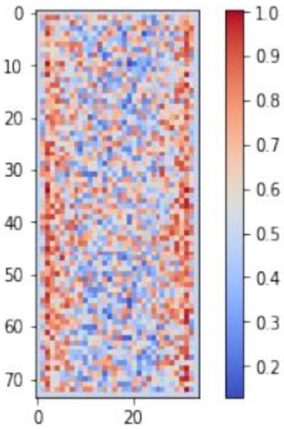
Overall occupancy plots are among the most important DQM plots and is used as input for ML studies

They show the frequency of hits in given detector channels

Used to identify anomalies and diagnose problems




Drift Tubes
(DT)




Hadronic calorimeter
(HCAL)

# Drift Tubes (DT) [4]

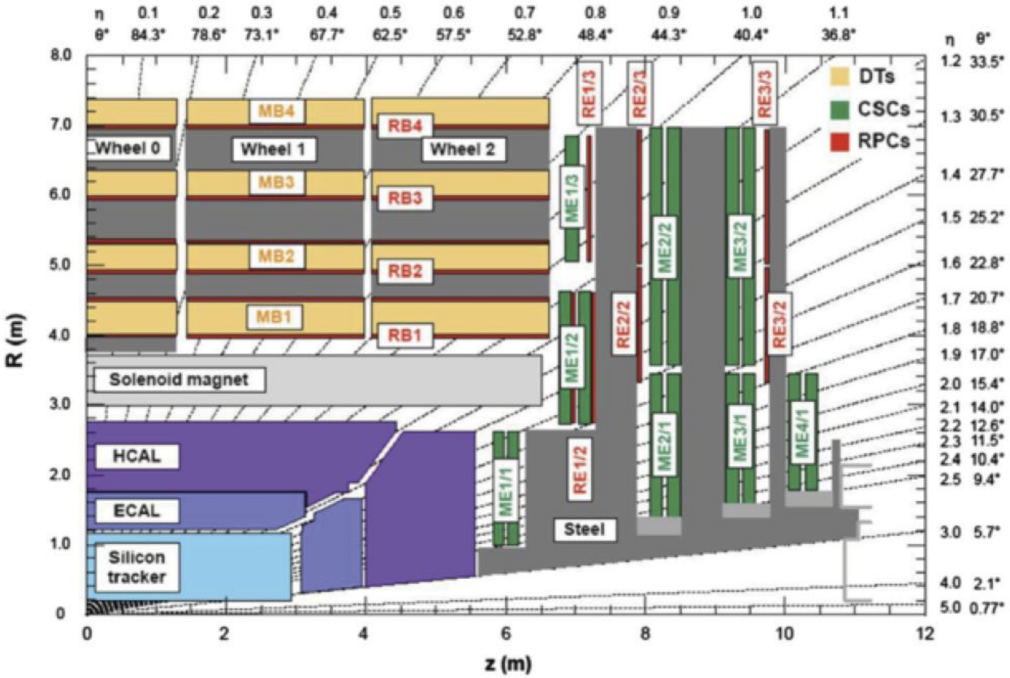
Barrel Muon sub-detector ( |η|≲1.1 )

~ 180k channels

250 chambers

2 x 2.5m in size

12 layers  ~60 ch/each



# Dataset

Hit occupancy contains the total number of electronic hits at each readout channel: 2-dimensional array

Dataset 21.000 occupancy plots

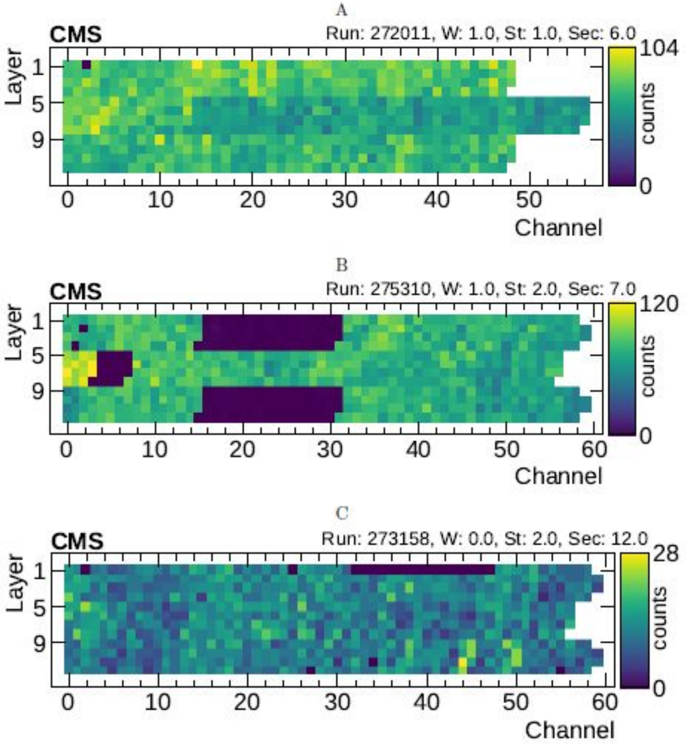
Labels (provided by humans):

- 5668 : 612 (GOOD : BAD)
- 90:10 class distribution ratio

**A:** Dead one channel

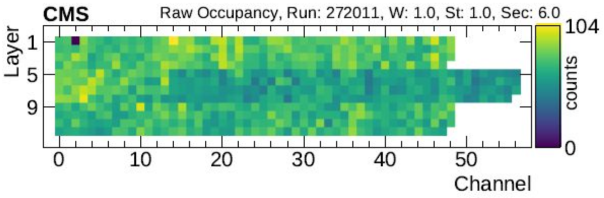
**B:** Dead regions in multiple layers

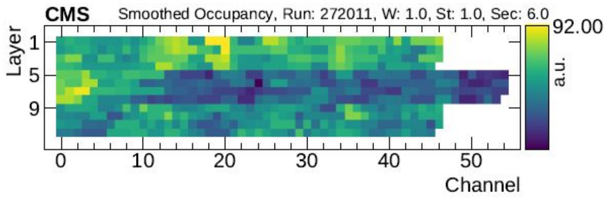
**C:** Dead region in one layer



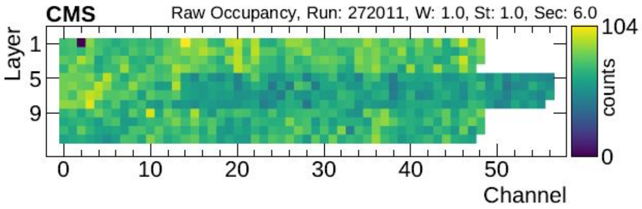
# Data preprocessing

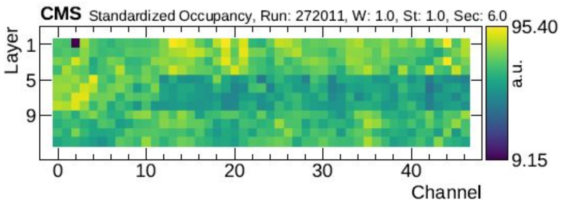
Smoothing. According to CMS DT experts isolated misbehaving channels are not considered a problem




Standardization into fixed dimensionality. 1D Linear interpolation



# Approaches to the anomaly detection in DT

**Local:**

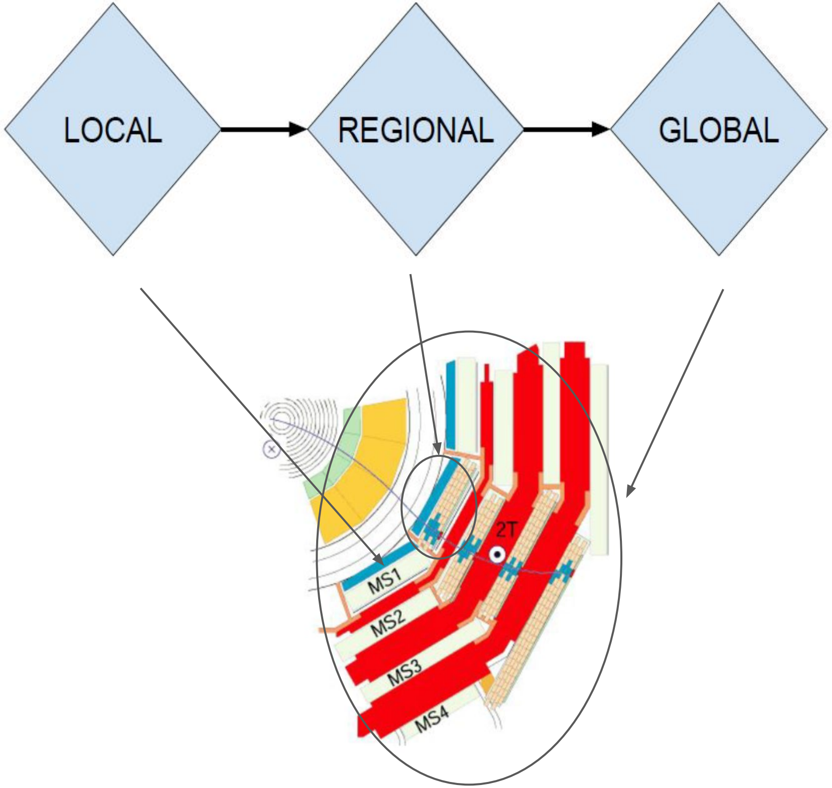
- Each layer is treated independently from the other layers within a chamber

**Regional**:

- Extend the local approach to account for intra-chamber problems
- Use information of all layers in a chamber, but each chamber independently from the others

**Global:**

- Use information of all the chambers for a given acquisition run
- The position of the chamber in the CMS detector impacts occupancy distribution of the channel hits



# Local strategy: scope, methods & results

Convolutional neural network (CNN) outperforms other methods. ROC AUC = 0.995

Activations: ReLU and softmax

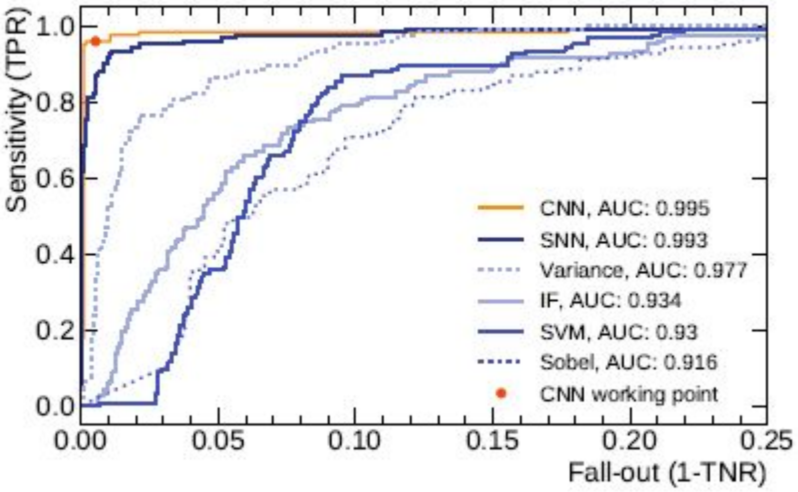
Optimizer: Adam

Loss function: cross entropy

Filters out most of the anomalies

Assessing the (mis)behavior with high-granularity (few channels)

Each layer is treated independently from the other layers



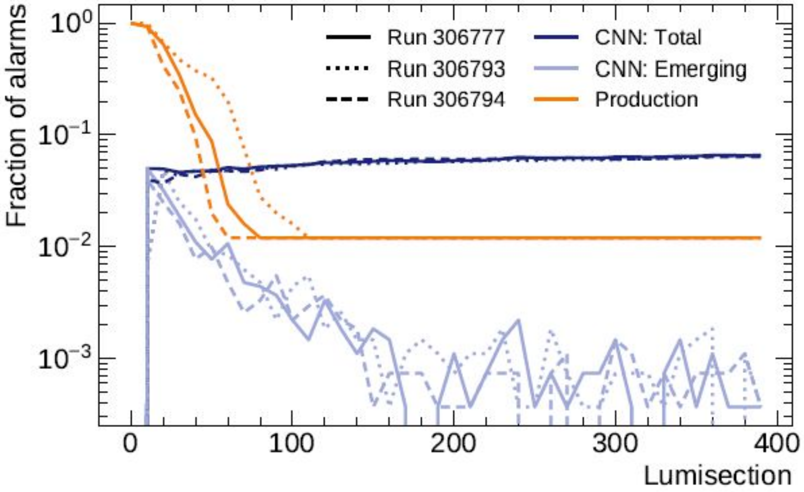

- Unsupervised
  - Sobel filter
- Semi-supervised
  - SVM
  - Isolation Forest
- Supervised
  - Shallow neural network (SNN)
  - Convolutional neural network (CNN)

# Local strategy: scope, methods & results

The local approach has satisfactory performance and was successfully implemented in production (the DT experts still test it)

Stability of the CNN and the production algorithm as a function of time (number of lumisections) for three different runs

The proposed strategy is generic enough to be applicable to other kinds of CMS muon chambers, as well as to other sub-detectors

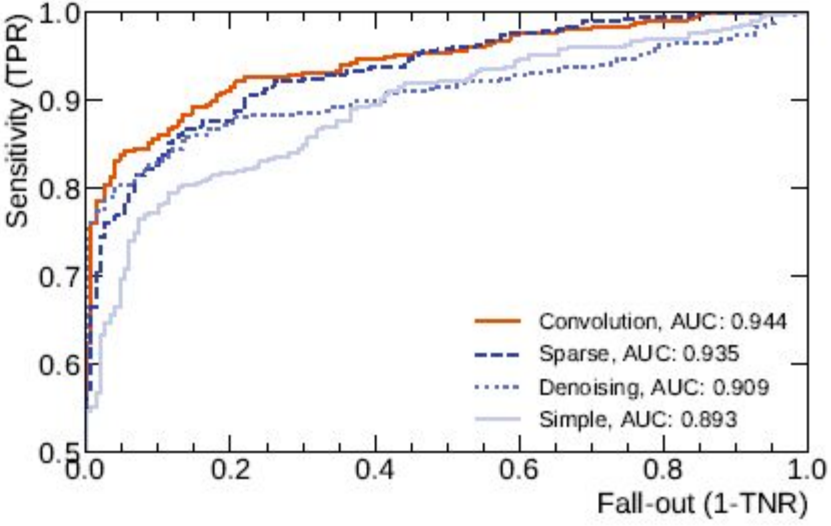


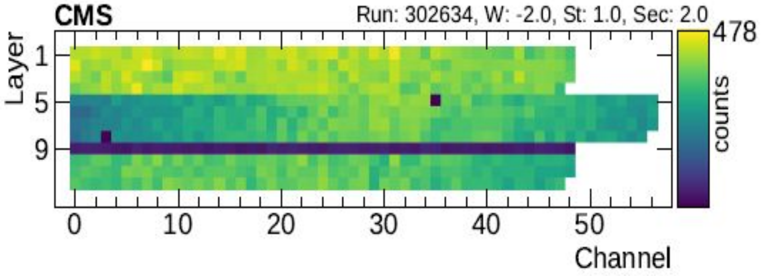
# Regional strategy: scope, methods & results

Extends local strategy to filter out anomalies not seen by the previous approach

Accounts for intra-chamber problems: simultaneously consider all layers in a chamber

The occupancy pattern within a chamber depends on the layer (row) information




Semi-supervised autoencoder variations:
- (simple) bottleneck
- Denoising
- Sparse
- Convolutional

# Global strategy: scope, method

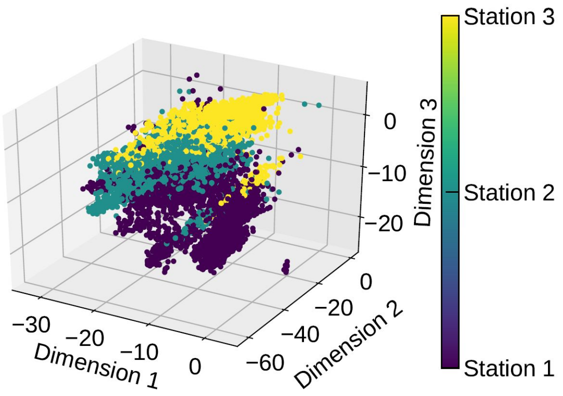
Simultaneous use of all the chambers data

The position impacts expected occupancy pattern

Autoencoders learn a compressed representation of chamber data

When the bottleneck of the autoencoder is 3-dimensional one can visually inspect those representation

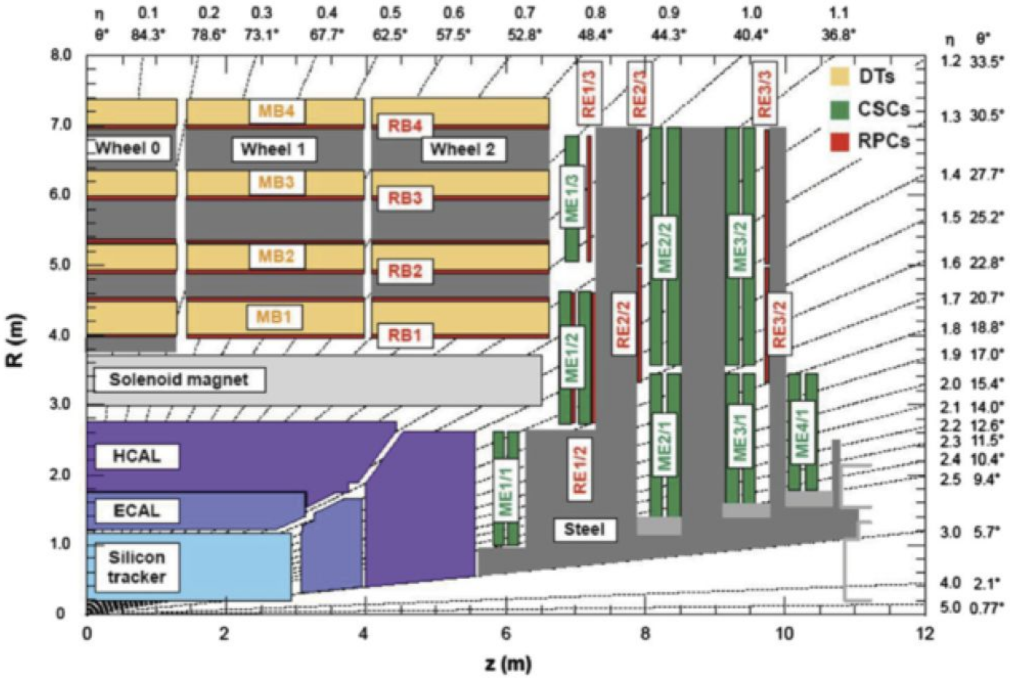
The global approach is then potentially capable to spot an unusual behavior of DT chambers taking into account the geographical constraints



# HCAL

Hadronic Calorimeter (HCAL)

- brass-scintillator sampling calorimeter
- coverage up to |η|≈ 3
- ~13k channels



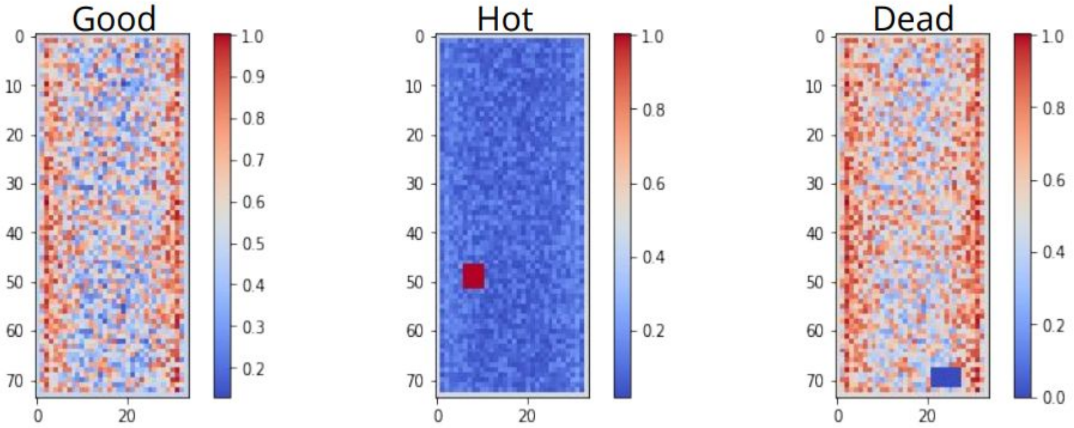
# HCAL: dataset

Hit occupancy contains the total number of electronic hits at each readout channel: 2-dimensional array

Have mostly good data

Manually simulate bad data by setting region

- Dead (no activity)
- Hot (high activity)



# HCAL: supervised

Convolutional neural network

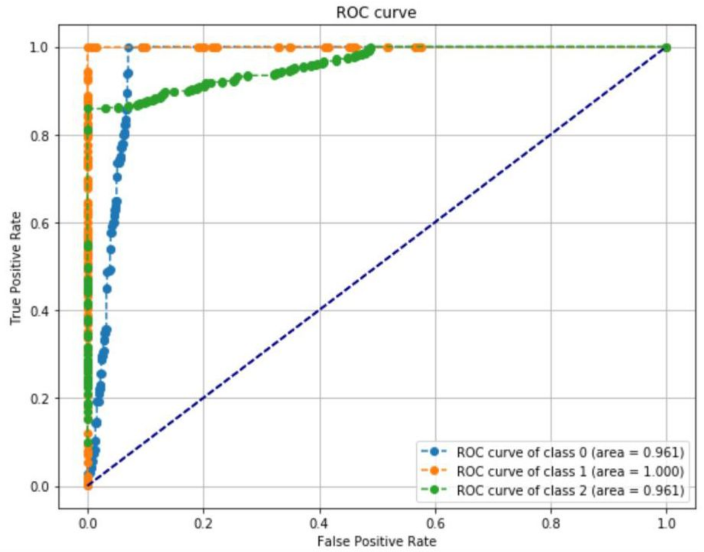
Activation: ReLU

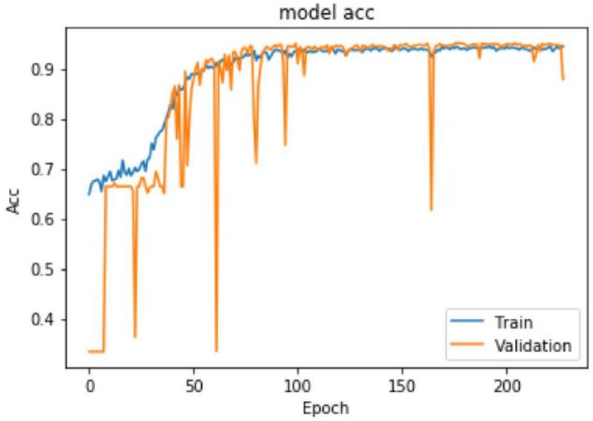
Optimizer: Adam

Loss function: categorical cross entropy

Accuracy: 0.95

ROC AUC: 1, 0.961, 0.961



# HCAL: semi-supervised

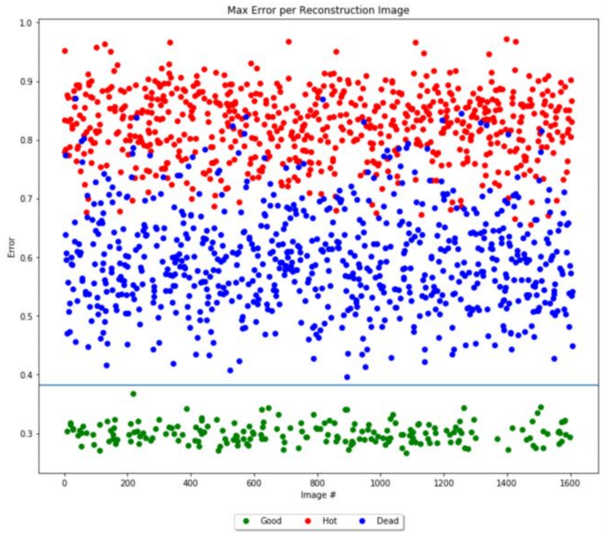
Bottleneck autoencoder with convolutional layers

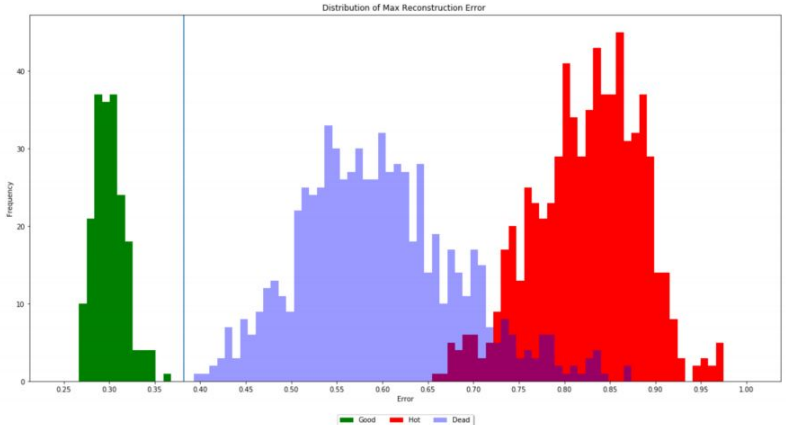
Activation: ReLU

Optimizer: Adadelta

Loss function: mean square error

GOOD vs BAD(hot/dead) are well differentiable even with simple parameters



# Outline

- CMS detector
- Current DQM
  - Tools
  - Online: Detector monitoring
  - Offline: Data certification
  - Limitations
- ML-based DQM
  - How to fit ML into DQM operations
  - Applicability studies
    - Online
    - Offline

# Offline: data certification

# Dataset 2010

Collected by CMS in 2010. Reconstructed data

Available through CERN OpenData

Use only minimal bias, muon, photon streams

16.000 lumisections

891 features:

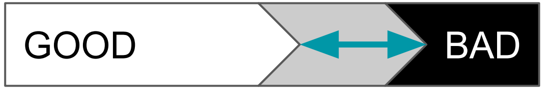
- 267 muon, 232 photon, 126 PF jets, 266 calo jets
- observables: transverse momentum, angle, coordinates, mass, etc

# Towards automation of data quality system for CERN CMS experiment [8]

Classification into 3 categories

- Definitely GOOD (white zone)
- Definitely BAD (black zone)
- Ambiguous (gray zone)
  - Decision can’t be made automatically
  - Human intervention is required

Aim to minimize gray zone (Rejection Rate)




Gradient Tree Boosting classifier

10 fold cross validation

$$\text{Rejection Rate} = \frac{\text{Rejected}}{\text{Total}} \to \min,$$

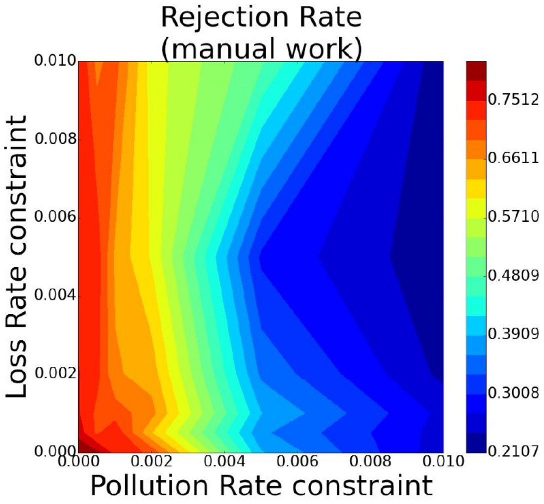
$$\text{Loss Rate} = \frac{\text{False Negative}}{\text{True Positive} + \text{False Negative}} \leq L_0,$$

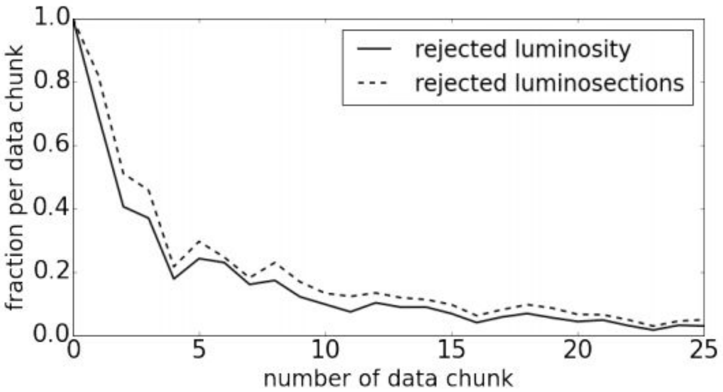
$$\text{Pollution Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Positive}} \leq P_0,$$

# Towards automation of data quality system for CERN CMS experiment [8]

System is able to automatically process at least 20% of samples keeping pollution and loss rates on negligible level

Less strict restrictions on pollution and loss  increase performance of the system significantly.



# Deep learning for inferring cause of data anomalies [2]

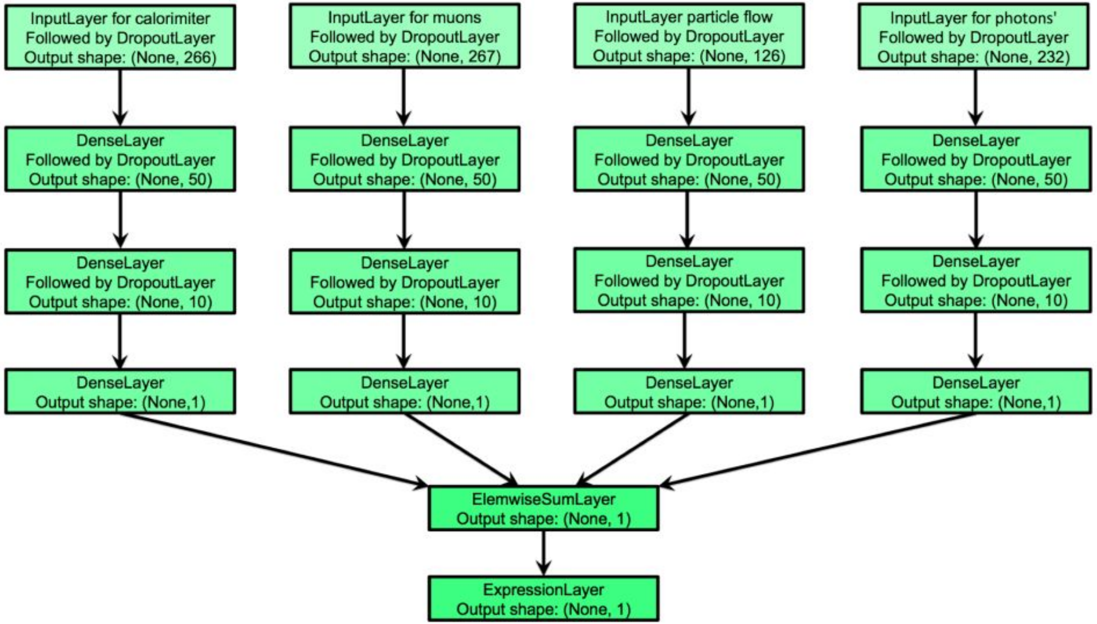
Determine which sub-detector is responsible for anomaly

4 NN for each particle type

- Photons
- Muons
- Particle Flow Jets
- Calorimeter Jets

Output is determined by `Fuzzy AND`

Loss function: dynamic cross-entropy



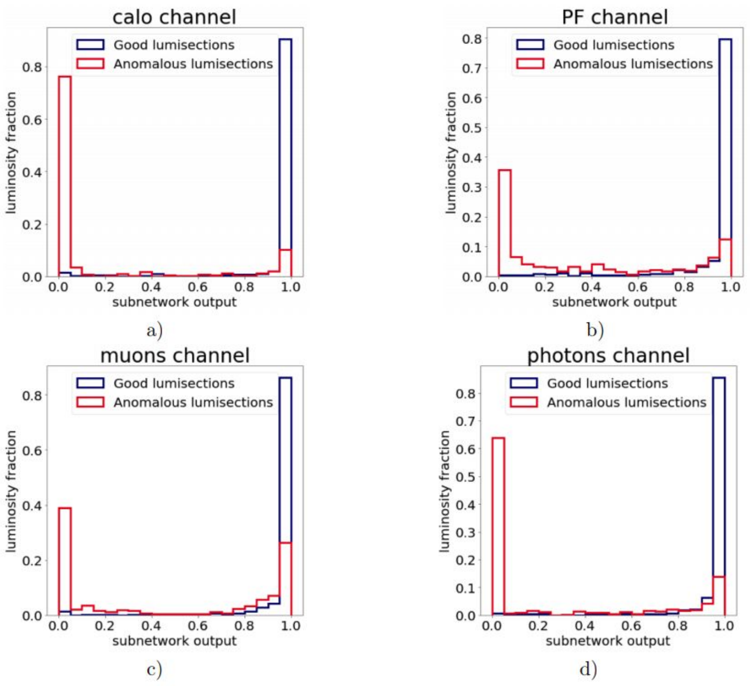
# Deep learning for inferring cause of data anomalies [2]

Each neural network returns a number:

- Close to 0 for BAD lumisections
- Close to 1 for GOOD lumisections
    - Invisible anomaly by this NN

10% of data for validation

ROC AUC = 0.96



# Dataset 2016

Collected by CMS in 2016. Reconstructed data

Dataset for Jet analysis. Jets probe most of the CMS sub-detectors

2807 features (401 * 7)

- Physics objects: photons, muons, etc
- Observables: energy, eta, phi, etc
- 7 = (Mean, RMS, Q1, Q2, Q3, Q4, Q5)

160.000 lumisections

98:2 class distribution ratio (GOOD:BAD)

# Anomaly detection using Autoencoders [3]

Semi-supervised approach

Train on only good data

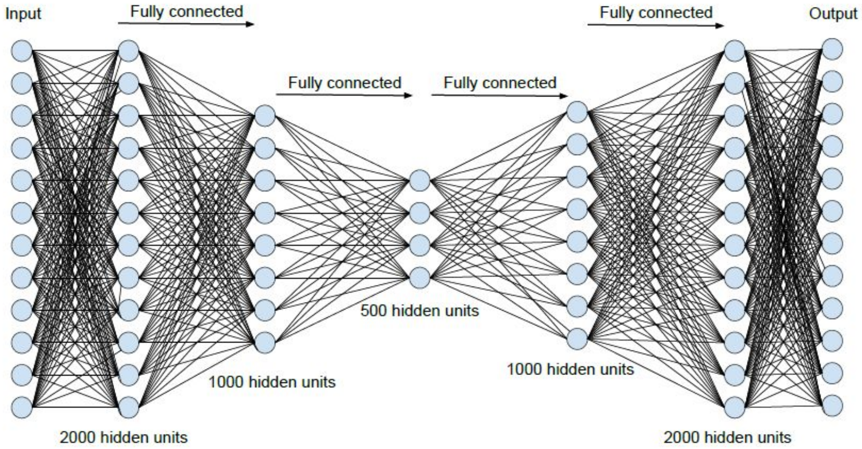
Data is sorted time-wise

Activations: PReLU

Optimiser: Adam (LR=0.0001)

Loss function: mean square error

Training-Validation-Test (60-20-20)



# Anomaly detection using Autoencoders [3]

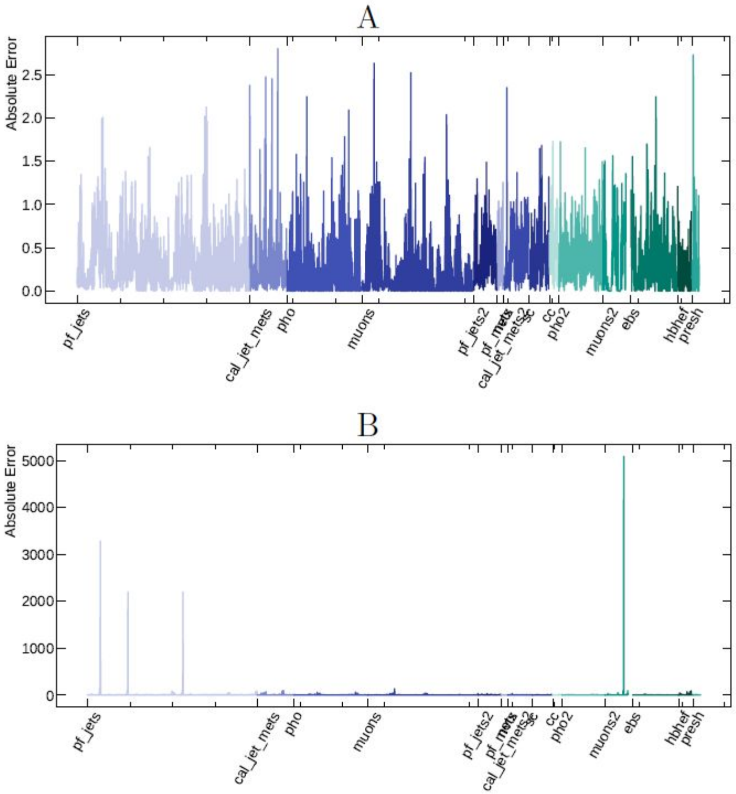
Features are grouped by physics object (x-axis)

**A**: GOOD lumisection. Reconstruction error is low

**B**: BAD lumisection. Reconstruction error is HIGH

- Observable peaks for anomalous features
- In this case muons and jets look anomalous

ROC AUC = 0.978



# Comparison of supervised ML models [6]

- Naive Bayes
  - Fast training
  - Poor predictive power
- SVM
  - Large number of high-dimensional data badly affected performance
- ANN (Sequential)
  - Average predictive power
  - Slow search of hyper parameters
- Random Forest
  - Fast training
  - Good predictive power
- Gradient Boosted Trees (XGBoost)
  - Good predictive power
  - Average training speed
  - High memory usage during training

# Comparison of supervised ML models [6]

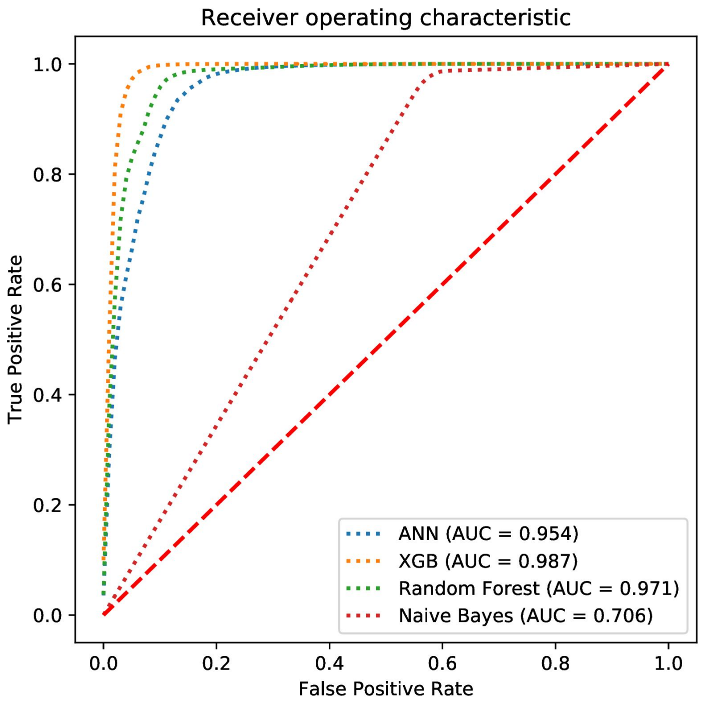
Class weights - more attention to minority class

Shuffle stratified 10 fold cross validation

Performance metrics:

- ROC AUC
- ACC
- $F_1$ score
- Training time

| | AUC | ± | ACC | ± | $F_1$ | ± | time | ± |
|---|---|---|---|---|---|---|---|---|
| XGB | **0.987** | 0.004 | **0.997** | 0.000 | **0.998** | 0.000 | 108.09 | 2.621 |
| Random Forest | 0.970 | 0.004 | 0.980 | 0.001 | 0.990 | 0.000 | **44.925** | 2.490 |
| ANN | 0.954 | 0.005 | 0.961 | 0.015 | 0.979 | 0.008 | 130.236 | 38.413 |
| Naive Bayes | 0.706 | 0.008 | 0.971 | 0.002 | 0.985 | 0.001 | 10.529 | 1.289 |



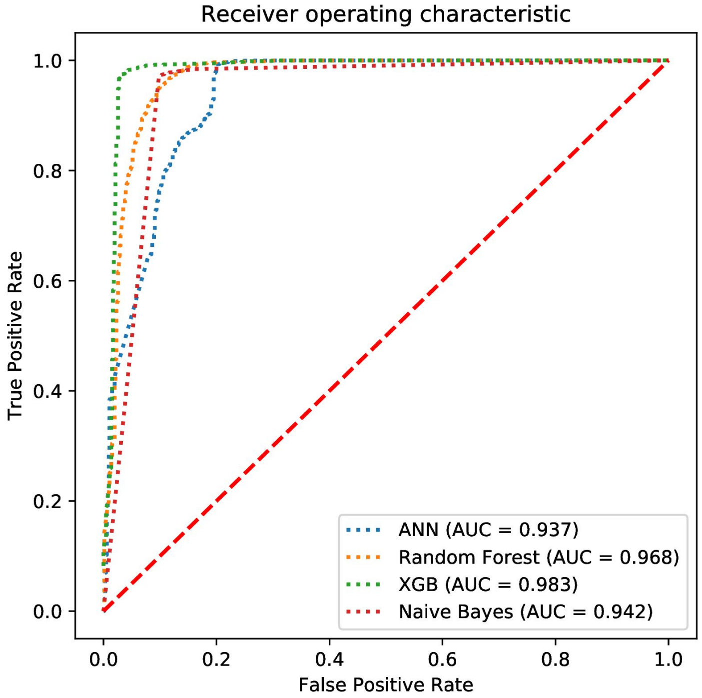
# Comparison of supervised ML models [6]

random_state = *my fav number*

Train and test set distribution trap

Naive Bayes model performs ~25% better than in previous experiment. NOT good!

Lesson learned: always use cross validation



# CMS partnership with industry

In the past few years the CMS experiment successfully engaged in partnership with IBM and Yandex through CERN Openlab framework

Objectives:

With IBM: to support automatization of online data quality monitoring using ML [1]

With Yandex: to support automatization of offline data certification process using ML [8]

# Run 3

Experience we learned from studying ML4DQM and ML4DC has been extremely valuable

Some prototype implementation already in hands with promising results

Plan to integrate ML tools in the standard Monitoring and Data Certification procedures for Run 3

We don’t expect to replace people

In Run 3, we still expect to have online/offline shift people, however, with ML, we expect much improved data quality monitoring and certification

Keep synergy with industry

Use detector metadata (HV, temp, etc) to predict hardware failures

# Recommendations for ML

Go supervised!

Go labels!

Go cross validation!

Questions, ideas, feedback

cms-ml4dc@cern.ch

cms-ml4dqm@cern.ch

# References


[1] Virginia Azzolini et al, “Improving the use of data quality metadata via a partnership of technologies and resources between the CMS experiment at CERN and industry”, CHEP 2018, https://indico.cern.ch/event/587955/contributions/2935731/

[2] Virginia Azzolini et al, “Deep learning for inferring cause of data anomalies”, ACAT 2017, http://inspirehep.net/record/1637193/files/arXiv:1711.07051.pdf

[3] Adrian Alan Pol et al, “Anomaly detection using Deep Autoencoders for the assessment of the quality of the data acquired by the CMS experiment”, CHEP 2018, https://indico.cern.ch/event/587955/contributions/2937523/

[4] Adrian Alan Pol et al, “Online detector monitoring using AI: challenges, prototypes and performance evaluation for automation of online quality monitoring of the CMS experiment exploiting machine learning algorithms”, CHEP 2018, https://indico.cern.ch/event/587955/contributions/2937517/

[5] Marcel Andre Schneider et al,  “The Data Quality Monitoring Software for the CMS experiment at the LHC: past, present and future”, CHEP 2018, https://indico.cern.ch/event/587955/contributions/2937597/

[6] Mantas Stankevičius et al, “Comparison of Supervised Machine Learning Techniques for CERN CMS Offline Data Certification”, Baltic DB&IS2018, http://ceur-ws.org/Vol-2158/paper18dc6.pdf

[7] Cesare Calabria, “Monitoring tools for the CMS muon detector: present workflows and future automation” https://indico.cern.ch/event/587955/contributions/2937547/

[8] Fedor Ratnikov, “Towards automation of data quality system for CERN CMS experiment”, http://iopscience.iop.org/article/10.1088/1742-6596/898/9/092041

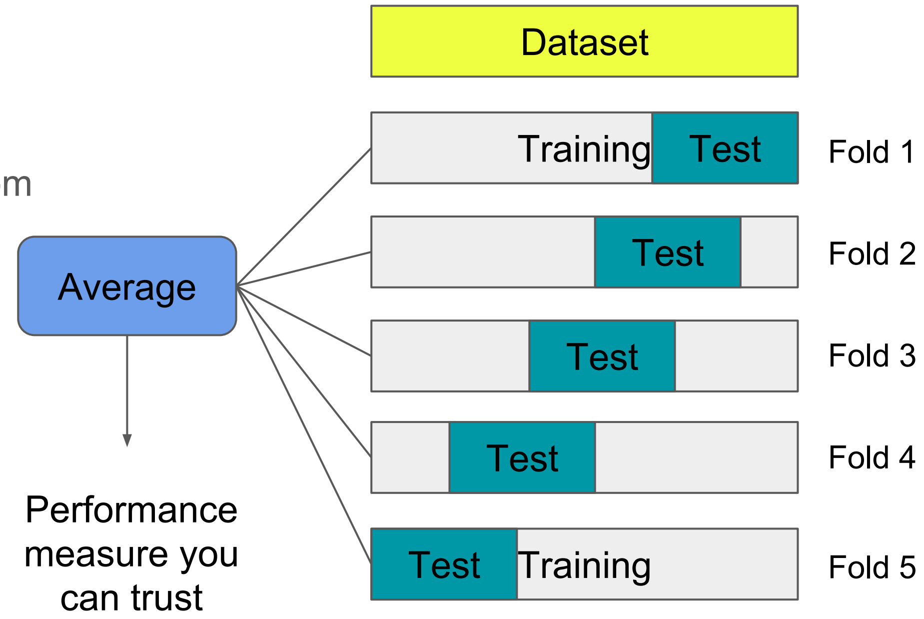
# Backup

# Cross validation

1. Partition dataset into multiple train : test folds
2. Train and evaluate model with all folds
3. Average scores

Averaged performance measure is independent from train : test distribution

Solution to overfitting



# HCAL: semi-supervised results

Reconstruction of good, dead and hot